An Automatic Finite-Sample Robustness Metric: Can Dropping a Little Data Change Conclusions?

Tamara Broderick

Associate Professor,
MIT

With Ryan Giordano, Rachael Meager

When can I trust my data analysis?

[https://stanfordmlgroup.github.io/projects/chexnet/, https://www.ibmbigdatahub.com/blog/cyber-security-powered-ai-and-machine-learning, https://www.fairwinds.org/inside-fairwinds/fraud-protection-center/report-fraud.html]

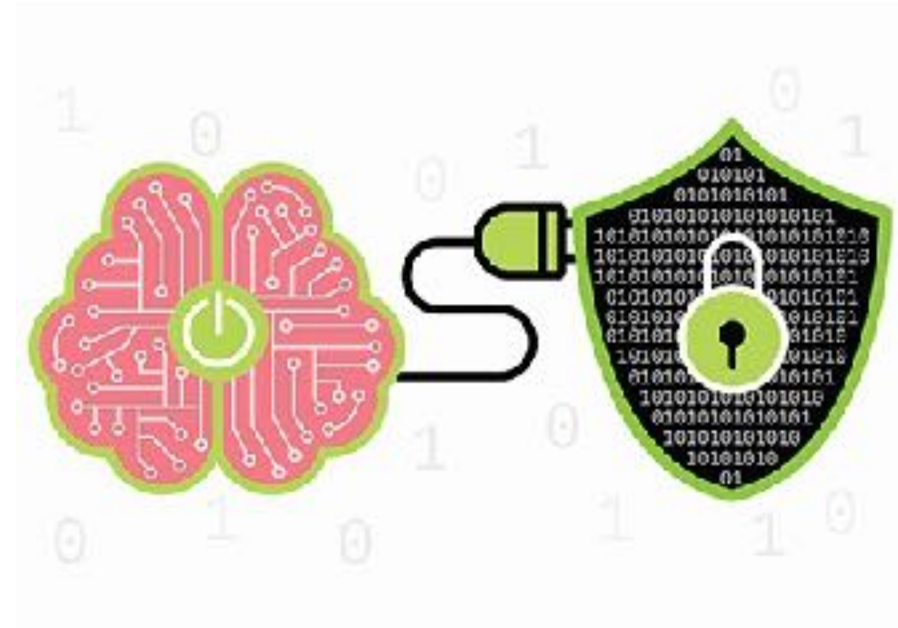
When can I trust my data analysis?

- More data & better computation ➔ data analyses increasingly drive life-changing decisions

[https://stanfordmlgroup.github.io/projects/chexnet/, https://www.ibmbigdatahub.com/blog/cyber-security-powered-ai-and-machine-learning, https://www.fairwinds.org/inside-fairwinds/fraud-protection-center/report-fraud.html]

When can I trust my data analysis?

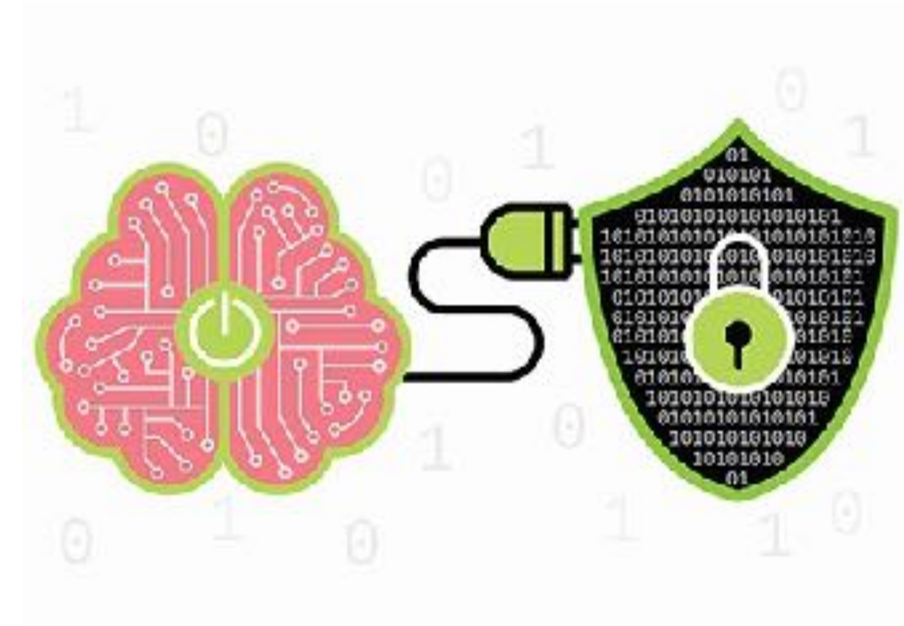
- More data & better computation ➔ data analyses increasingly drive life-changing decisions



[https://stanfordmlgroup.github.io/projects/chexnet/, https://www.ibmbigdatahub.com/blog/cyber-security-powered-ai-and-machine-learning, https://www.fairwinds.org/inside-fairwinds/fraud-protection-center/report-fraud.html]

When can I trust my data analysis?

- More data & better computation ➔ data analyses increasingly drive life-changing decisions

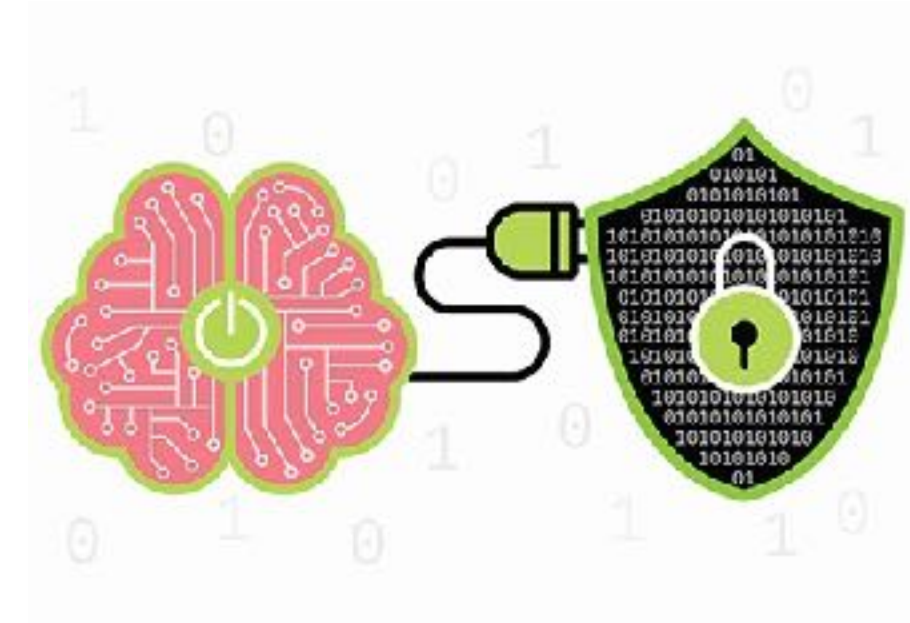


- One question: Would you be concerned if dropping a small fraction of data changed substantive conclusions?

[https://stanfordmlgroup.github.io/projects/chexnet/, https://www.ibmbigdatahub.com/blog/cyber-security-powered-ai-and-machine-learning, https://www.fairwinds.org/inside-fairwinds/fraud-protection-center/report-fraud.html]

When can I trust my data analysis?

- More data & better computation ➔ data analyses increasingly drive life-changing decisions

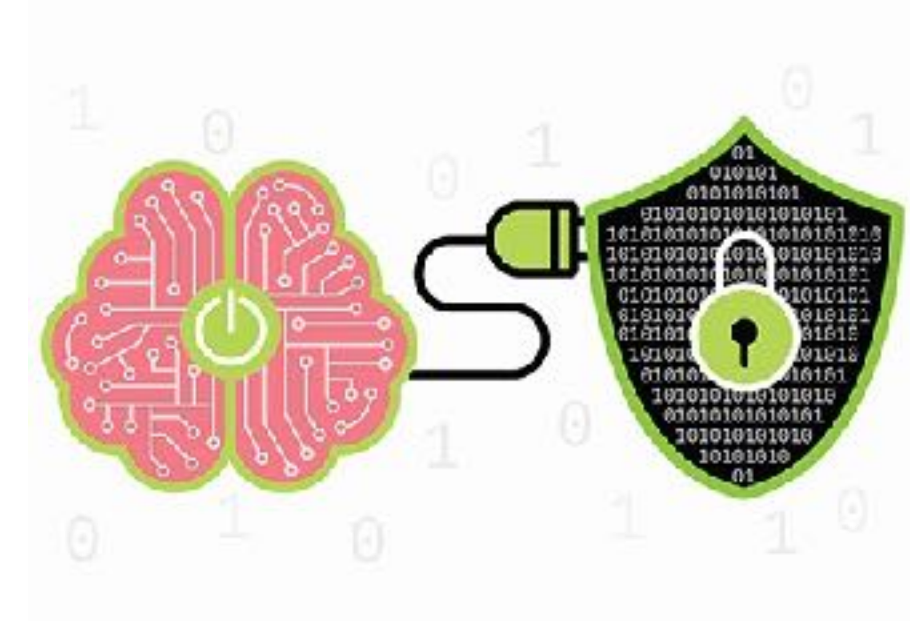


- One question: Would you be concerned if dropping a small fraction of data changed substantive conclusions?
- **Challenge**: Too expensive to check every data subset

[https://stanfordmlgroup.github.io/projects/chexnet/, https://www.ibmbigdatahub.com/blog/cyber-security-powered-ai-and-machine-learning, https://www.fairwinds.org/inside-fairwinds/fraud-protection-center/report-fraud.html]

When can I trust my data analysis?

- More data & better computation ➔ data analyses increasingly drive life-changing decisions

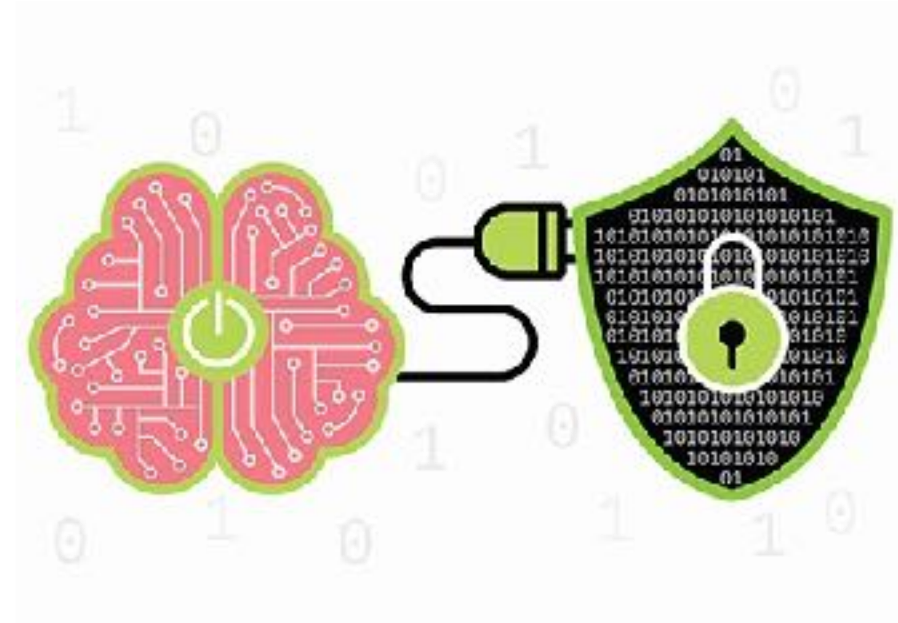


- One question: Would you be concerned if dropping a small fraction of data changed substantive conclusions?
- **Challenge**: Too expensive to check every data subset
- **Our Solution**: a fast, automated, accurate *approximation*

[https://stanfordmlgroup.github.io/projects/chexnet/, https://www.ibmbigdatahub.com/blog/cyber-security-powered-ai-and-machine-learning, https://www.fairwinds.org/inside-fairwinds/fraud-protection-center/report-fraud.html]

When can I trust my data analysis?

- More data & better computation ➔ data analyses increasingly drive life-changing decisions



- One question: Would you be concerned if dropping a small fraction of data changed substantive conclusions?
- **Challenge**: Too expensive to check every data subset
- **Our Solution**: a fast, automated, accurate *approximation*
- E.g. in a study of microcredit with ~16,500 data points, we find a single data point that drives the sign of the effect

[https://stanfordmlgroup.github.io/projects/chexnet/, https://www.ibmbigdatahub.com/blog/cyber-security-powered-ai-and-machine-learning, https://www.fairwinds.org/inside-fairwinds/fraud-protection-center/report-fraud.html]

Roadmap

Roadmap

- When do we care about dropping data subsets?

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation
- Many analyses are robust but some aren't

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation
- Many analyses are robust but some aren't
  - Here non-robustness isn't just a product of gross outliers, large p-values, heavy tails, etc.

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation
- Many analyses are robust but some aren’t
    - Here non-robustness isn’t just a product of gross outliers, large p-values, heavy tails, etc.
    - It’s a product of low signal-to-noise

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation
- Many analyses are robust but some aren’t
  - Here non-robustness isn’t just a product of gross outliers, large p-values, heavy tails, etc.
  - It’s a product of low signal-to-noise

Why care about dropping data subsets?

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem
- Let’s look at economics (e.g. because of the important applications and wonderful reproducibility)

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem
- Let’s look at economics (e.g. because of the important applications and wonderful reproducibility)
  - Report a convenient proxy (e.g. mean)

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem
- Let’s look at economics (e.g. because of the important applications and wonderful reproducibility)
  - Report a convenient proxy (e.g. mean)
  - Small fractions of data often missing not-at-random

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem
- Let's look at economics (e.g. because of the important applications and wonderful reproducibility)
    - Report a convenient proxy (e.g. mean)
    - Small fractions of data often missing not-at-random
    - Policy population different from analyzed population

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem
- Let's look at economics (e.g. because of the important applications and wonderful reproducibility)
  - Report a convenient proxy (e.g. mean)
  - Small fractions of data often missing not-at-random
  - Policy population different from analyzed population
  - Models are necessarily misspecified

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem
- Let's look at economics (e.g. because of the important applications and wonderful reproducibility)
    - Report a convenient proxy (e.g. mean)
    - Small fractions of data often missing not-at-random
    - Policy population different from analyzed population
    - Models are necessarily misspecified
- In all these cases, we'd be concerned if dropping a small fraction of data changed our conclusions

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem
- Let's look at economics (e.g. because of the important applications and wonderful reproducibility)
  - Report a convenient proxy (e.g. mean)
  - Small fractions of data often missing not-at-random
  - Policy population different from analyzed population
  - Models are necessarily misspecified
- In all these cases, we'd be concerned if dropping a small fraction of data changed our conclusions
- Concerns not specific to economics

Why care about dropping data subsets?

- Any useful data analysis should be sensitive to *some* change in the data
- What types of sensitivity concern us? Varies by problem
- Let's look at economics (e.g. because of the important applications and wonderful reproducibility)
    - Report a convenient proxy (e.g. mean)
    - Small fractions of data often missing not-at-random
    - Policy population different from analyzed population
    - Models are necessarily misspecified
- In all these cases, we'd be concerned if dropping a small fraction of data changed our conclusions
- Concerns not specific to economics
- Even if doesn't bother you, should be up front about it

Dropping data & computational cost

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most $100\alpha\%$ of the data

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0,1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
  - **Most Influential Set**: data dropped to achieve MIP

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
  - **Most Influential Set**: data dropped to achieve MIP
  - **Perturbation-Inducing Proportion**: Min data proportion to achieve a certain change (or `NA` if none)

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0,1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
  - **Most Influential Set**: data dropped to achieve MIP
  - **Perturbation-Inducing Proportion**: Min data proportion to achieve a certain change (or `NA` if none)
- How to find Maximum Influence Perturbation: re-run data analysis with every appropriate subset dropped

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0,1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
  - **Most Influential Set**: data dropped to achieve MIP
  - **Perturbation-Inducing Proportion**: Min data proportion to achieve a certain change (or `NA` if none)
- How to find Maximum Influence Perturbation: re-run data analysis with every appropriate subset dropped
- Example: 400 data points, $\alpha = 0.01$

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
  - **Most Influential Set**: data dropped to achieve MIP
  - **Perturbation-Inducing Proportion**: Min data proportion to achieve a certain change (or `NA` if none)
- How to find Maximum Influence Perturbation: re-run data analysis with every appropriate subset dropped
- Example: 400 data points, $\alpha = 0.01$ → >1 billion re-runs

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
  - **Most Influential Set**: data dropped to achieve MIP
  - **Perturbation-Inducing Proportion**: Min data proportion to achieve a certain change (or `NA` if none)
- How to find Maximum Influence Perturbation: re-run data analysis with every appropriate subset dropped
- Example: 400 data points, $\alpha = 0.01$ → >1 billion re-runs
  - If analysis takes 1 second, check takes >31 years

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most $100\alpha\%$ of the data
  - **Most Influential Set**: data dropped to achieve MIP
  - **Perturbation-Inducing Proportion**: Min data proportion to achieve a certain change (or `NA` if none)
- How to find Maximum Influence Perturbation: re-run data analysis with every appropriate subset dropped
- Example: 16,000 data points, $\alpha = 0.001$

Dropping data & computational cost

- Might worry if removing small fraction $\alpha \in (0, 1)$ of data:
  - Changed sign of effect
  - Changed significance of effect
  - Changed both sign and significance, etc.
- Define
  - **Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
  - **Most Influential Set**: data dropped to achieve MIP
  - **Perturbation-Inducing Proportion**: Min data proportion to achieve a certain change (or `NA` if none)
- How to find Maximum Influence Perturbation: re-run data analysis with every appropriate subset dropped
- Example: 16,000 data points, $\alpha = 0.001$ → >10^{53} re-runs
  - If analysis takes 1 second, check takes >10^{46} years

A Motivating Example: Microcredit

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*
  - Original model:

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*
- Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
    - *Fantastic reproducibility and data sharing!*

microcredit indicator

- Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

profit microcredit indicator

- Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

profit parameters microcredit indicator

- Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

profit parameters microcredit indicator

  - Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
  - Result: $\hat{\theta}_1 =$ -4.55 USD PPP/2 weeks, std error 5.88

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

  profit parameters microcredit indicator

  - Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
  - Result: $\hat{\theta}_1 =$ -4.55 USD PPP/2 weeks, std error 5.88
- Our approximation:

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

    profit parameters microcredit indicator

  - Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
  - Result: $\hat{\theta}_1 =$ -4.55 USD PPP/2 weeks, std error 5.88
- Our approximation:
  - Takes 2 seconds to run (not 10^{46} years)

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

    profit parameters microcredit indicator

  - Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
  - Result: $\hat{\theta}_1 =$ -4.55 USD PPP/2 weeks, std error 5.88
- Our approximation:
  - Takes 2 seconds to run (not 10^{46} years)
  - Can remove 1 household & change sign: neg → pos

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
    - *Fantastic reproducibility and data sharing!*

profit parameters microcredit indicator

    - Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$
    - Result: $\hat{\theta}_1 =$ -4.55 USD PPP/2 weeks, std error 5.88
- Our approximation:
    - Takes 2 seconds to run (not 10^{46} years)
    - Can remove 1 household & change sign: neg → pos
    - Can remove 15 points to get $\hat{\theta}_1 =$ 7.03, std err 2.55

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

profit parameters microcredit indicator

  - Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
  - Result: $\hat{\theta}_1 =$ -4.55 USD PPP/2 weeks, std error 5.88
- Our approximation:
  - Takes 2 seconds to run (not 10^{46} years)
  - Can remove 1 household & change sign: neg → pos
  - Can remove 15 points to get $\hat{\theta}_1 =$ 7.03, std err 2.55
  - Can re-run regression to check directly

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

profit parameters microcredit indicator

  - Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
  - Result: $\hat{\theta}_1 =$ -4.55 USD PPP/2 weeks, std error 5.88
- Our approximation:
  - Takes 2 seconds to run (not 10^{46} years)
  - Can remove 1 household & change sign: neg → pos
  - Can remove 15 points to get $\hat{\theta}_1 =$ 7.03, std err 2.55
  - Can re-run regression to check directly
- It's not just non-significance, gross outliers, heavy tails, reporting means, or not using Bayes

A Motivating Example: Microcredit

- Angelucci et al 2015: Largest (16,561 households) of 7 randomized controlled trials examining effect of microcredit
  - *Fantastic reproducibility and data sharing!*

    profit parameters microcredit indicator
  - Original model: $y_n = \theta_0 + \theta_1 x_n + \epsilon_n, \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
  - Result: $\hat{\theta}_1 =$ -4.55 USD PPP/2 weeks, std error 5.88
- Our approximation:
  - Takes 2 seconds to run (not 10^{46} years)
  - Can remove 1 household & change sign: neg → pos
  - Can remove 15 points to get $\hat{\theta}_1 =$ 7.03, std err 2.55
  - Can re-run regression to check directly
- It's not just non-significance, gross outliers, heavy tails, reporting means, or not using Bayes; issue is signal to noise

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation
- Many analyses are robust but some aren’t
    - Here non-robustness isn’t just a product of gross outliers, large p-values, heavy tails, etc.
    - It’s a product of low signal-to-noise

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation
- Many analyses are robust but some aren't
  - Here non-robustness isn't just a product of gross outliers, large p-values, heavy tails, etc.
  - It's a product of low signal-to-noise

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation
- Many analyses are robust but some aren't
  - Here non-robustness isn't just a product of gross outliers, large p-values, heavy tails, etc.
  - It's a product of low signal-to-noise

Roadmap

- When do we care about dropping data subsets?
- How should we drop data subsets?
- Why is dropping data subsets computationally expensive?
- We provide a fast & automatic approximation
- Many analyses are robust but some aren't
  - Here non-robustness isn't just a product of gross outliers, large p-values, heavy tails, etc.
  - It's a product of low signal-to-noise

Setup for dropping data

Setup for dropping data

- A data analysis:

Setup for dropping data

- A data analysis:

d_n

data point

Setup for dropping data

- A data analysis:

d_n

data point;
e.g. (x_n, y_n)

Setup for dropping data

- A data analysis:

parameters

θ, d_n

data point;
e.g. (x_n, y_n)

Setup for dropping data

- A data analysis: $f(\theta, d_n)$

loss

parameters

data point;
e.g. (x_n, y_n)

Setup for dropping data

- A data analysis: $\sum_{n=1}^{N} f(\theta, d_n)$

loss

parameters

data point;
e.g. (x_n, y_n)

Setup for dropping data

loss parameters

- A data analysis: $\operatorname*{argmin}_{\theta} \sum_{n=1}^{N} f(\theta, d_n)$

data point;
e.g. (x_n, y_n)

Setup for dropping data

- A data analysis: $\hat{\theta} := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} f(\theta, d_n)$

loss

parameters

estimator

data point;
e.g. (x_n, y_n)

Setup for dropping data

- A data analysis: $\hat{\theta} := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} f(\theta, d_n)$

loss

parameters

estimator

data point;
e.g. (x_n, y_n)

  - E.g. max likelihood, min loss

Setup for dropping data

- A data analysis: $\hat{\theta} := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} f(\theta, d_n)$
  - E.g. max likelihood, min loss

loss

parameters

estimator

data point;
e.g. (x_n, y_n)

(Our approach actually handles even more general analyses)

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} f(\theta, d_n)$

estimator

data point;
e.g. (x_n, y_n)

  - E.g. max likelihood, min loss
- A quantity of interest ϕ

(Our approach actually handles even more general analyses)

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} f(\theta, d_n)$

estimator

data point; e.g. (x_n, y_n)

  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$

(Our approach actually handles even more general analyses)

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} f(\theta, d_n)$

estimator

data point;
e.g. (x_n, y_n)

  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} f(\theta, d_n)$

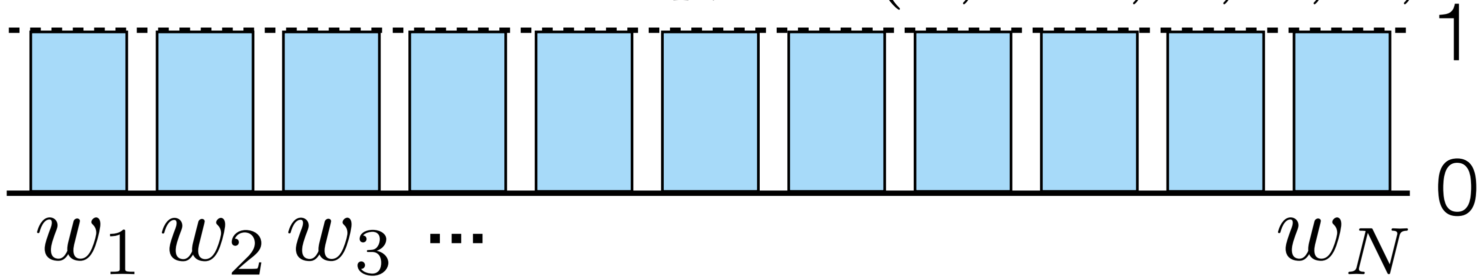
estimator

data point; e.g. (x_n, y_n)

  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} f(\theta, d_n)$

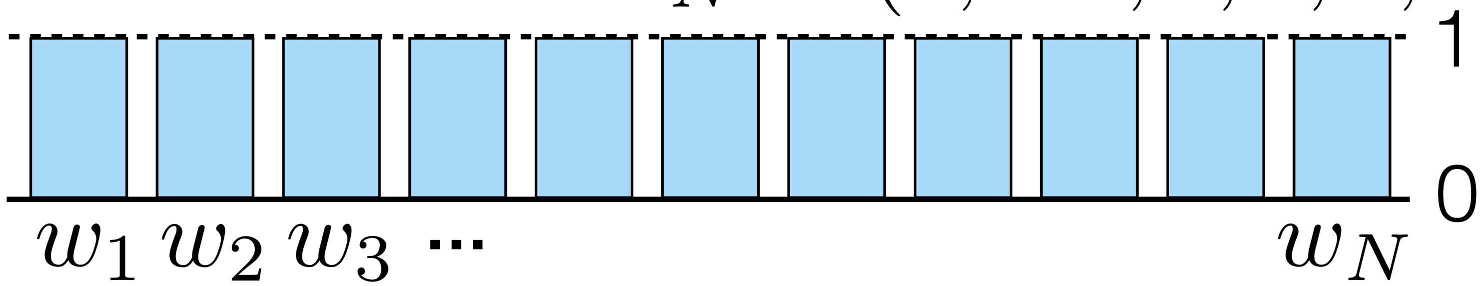
estimator

data point; e.g. (x_n, y_n)

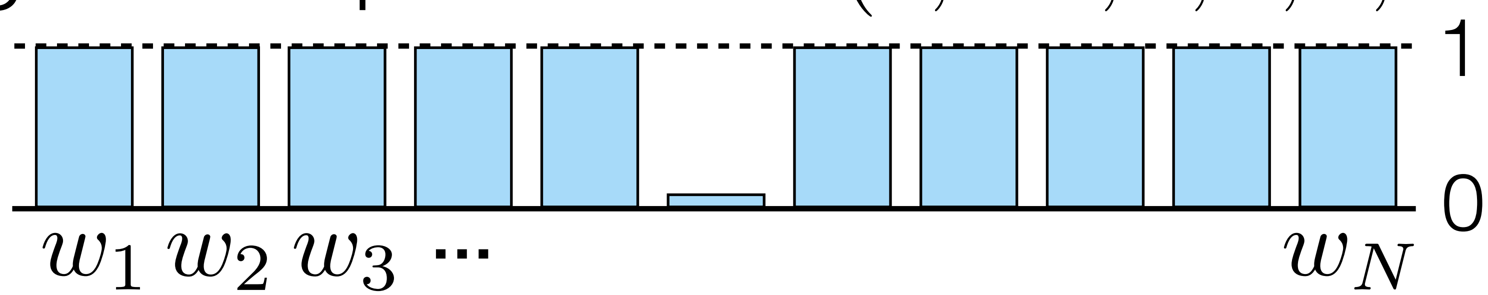
  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} f(\theta, d_n)$

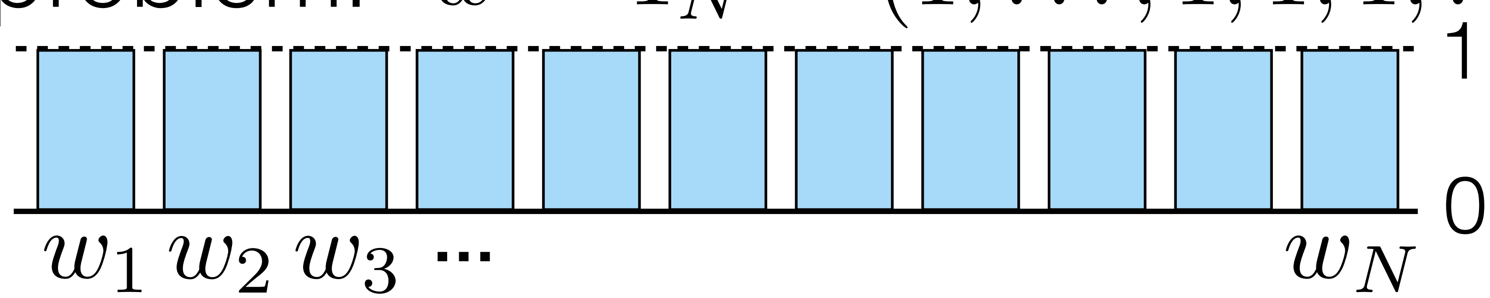
estimator

data point; e.g. (x_n, y_n)

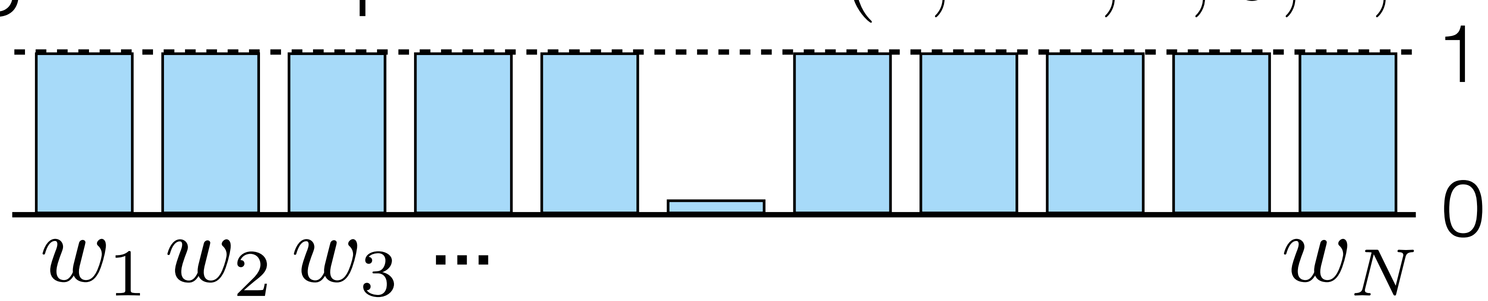
  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \underset{\theta}{\operatorname{argmin}} \boxed{\sum_{n=1}^{N} f(\theta, d_n)}$

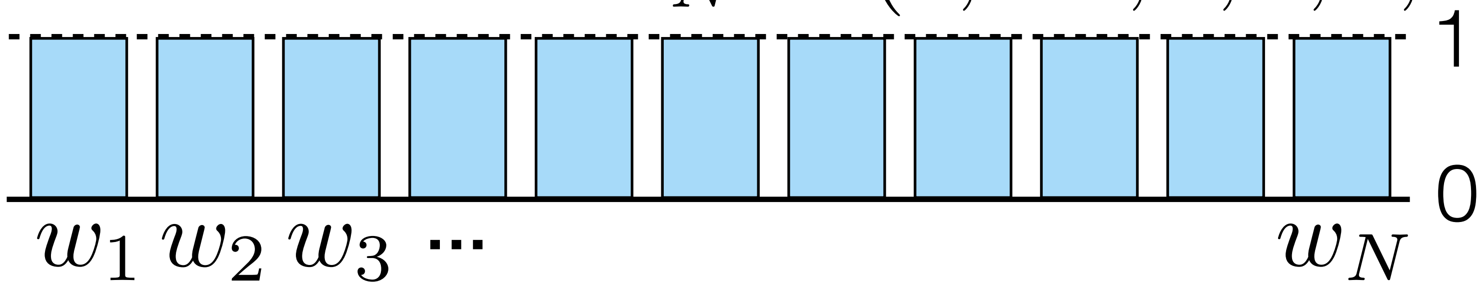
estimator

data point; e.g. (x_n, y_n)

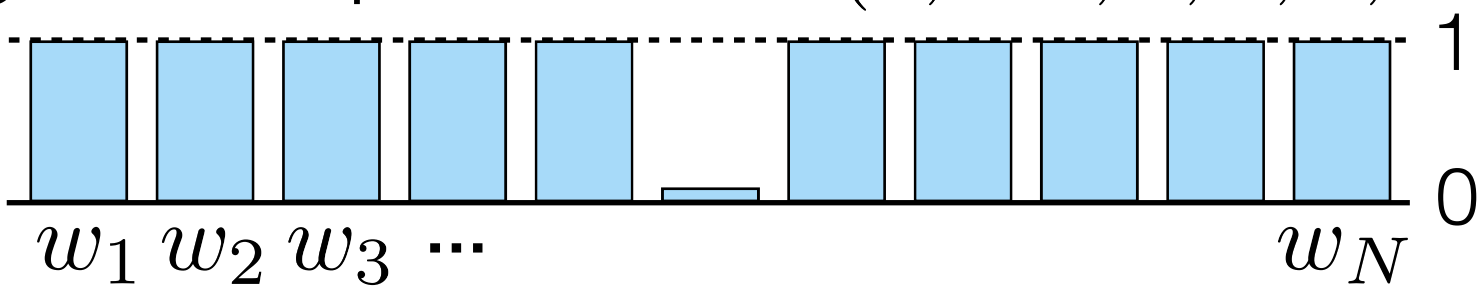
  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} f(\theta, d_n)$

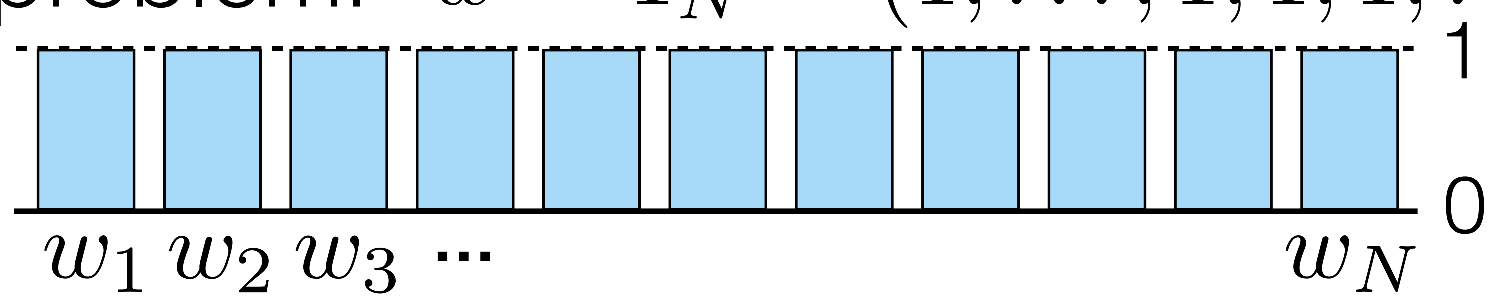
estimator

data point; e.g. (x_n, y_n)

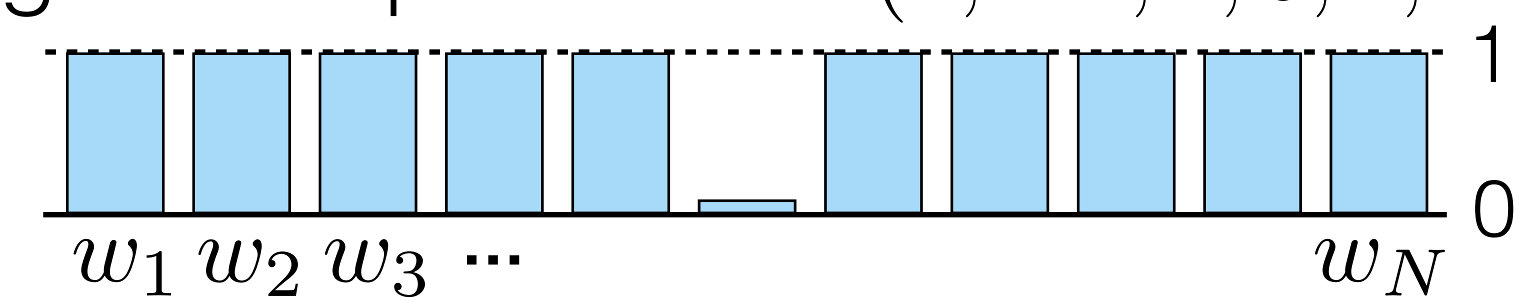
  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$

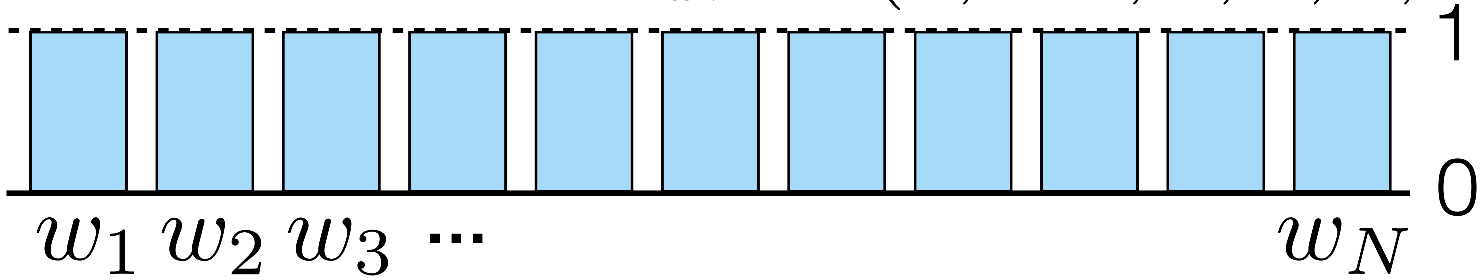
estimator

data point; e.g. (x_n, y_n)

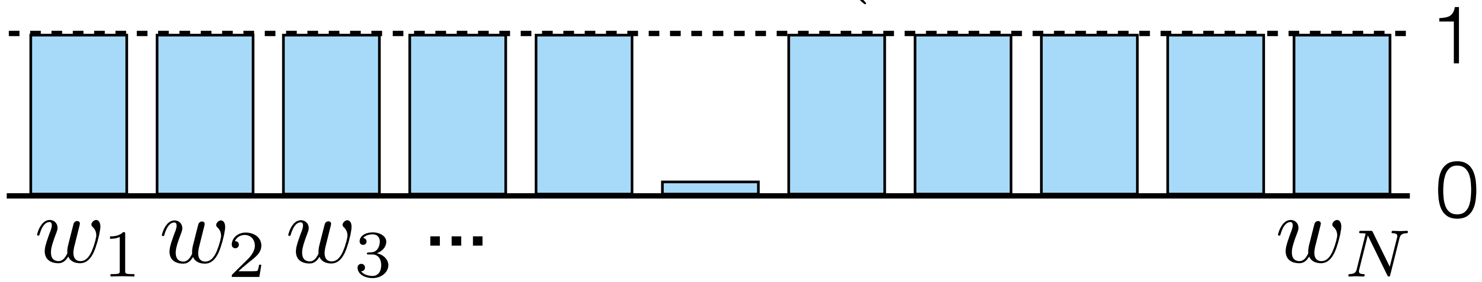
  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta} := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$

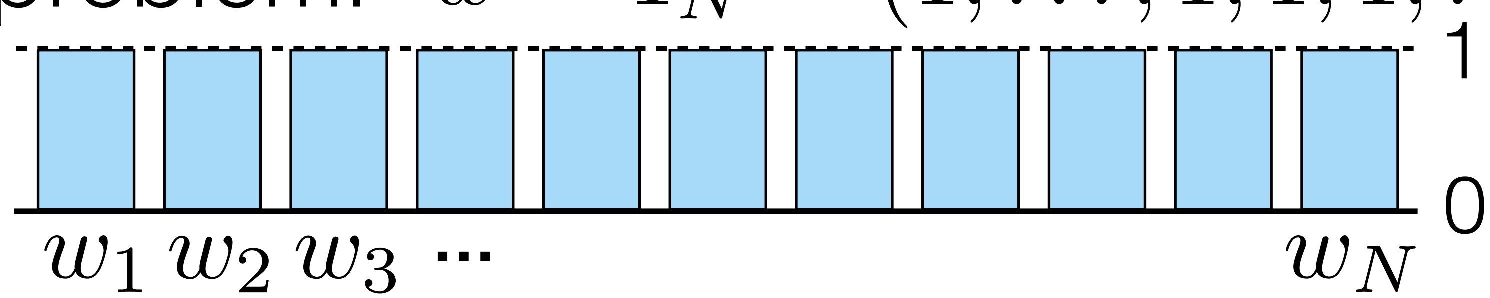
estimator

data point; e.g. (x_n, y_n)

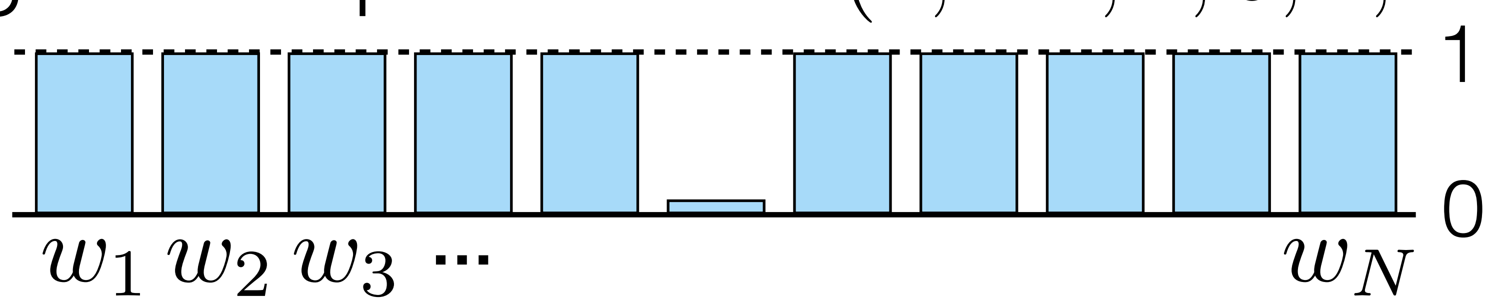
  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$

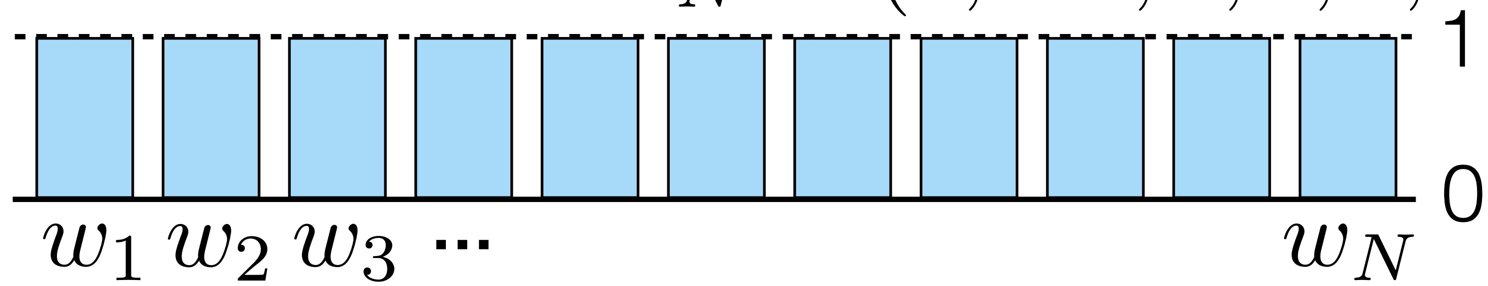
estimator

data point;
e.g. (x_n, y_n)

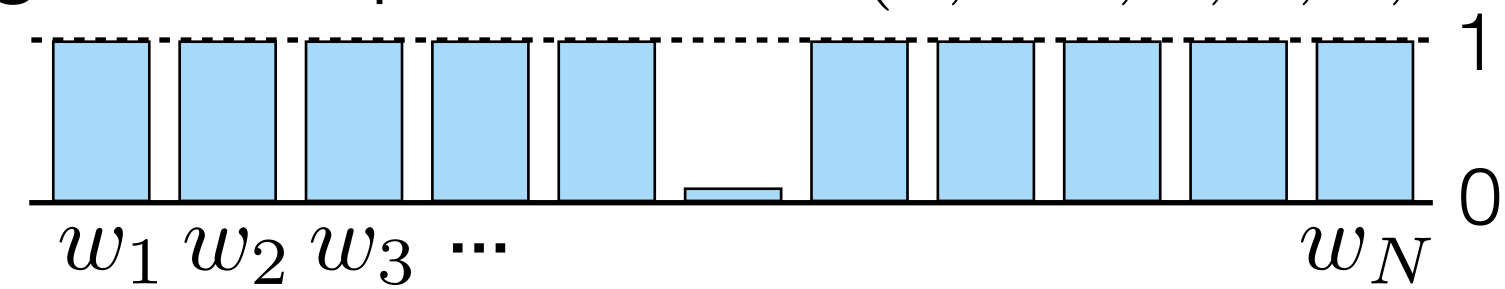
  - E.g. max likelihood, min loss
- A quantity of interest ϕ
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$

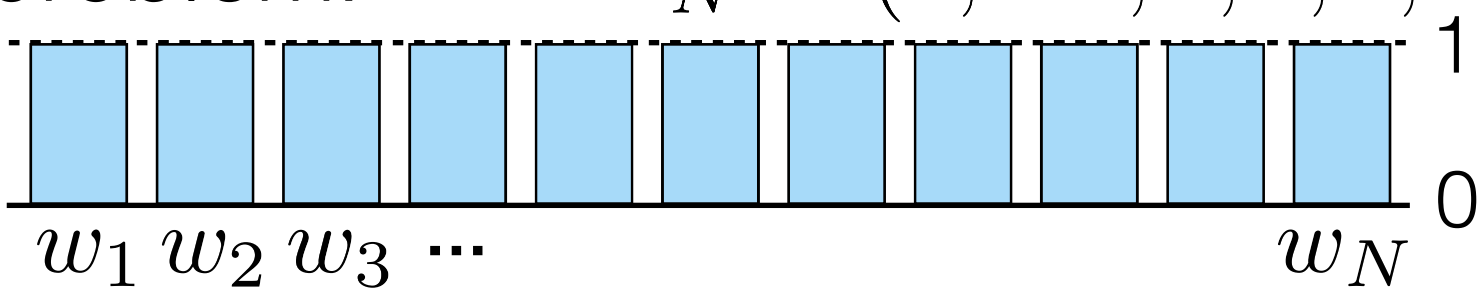
estimator

data point; e.g. (x_n, y_n)

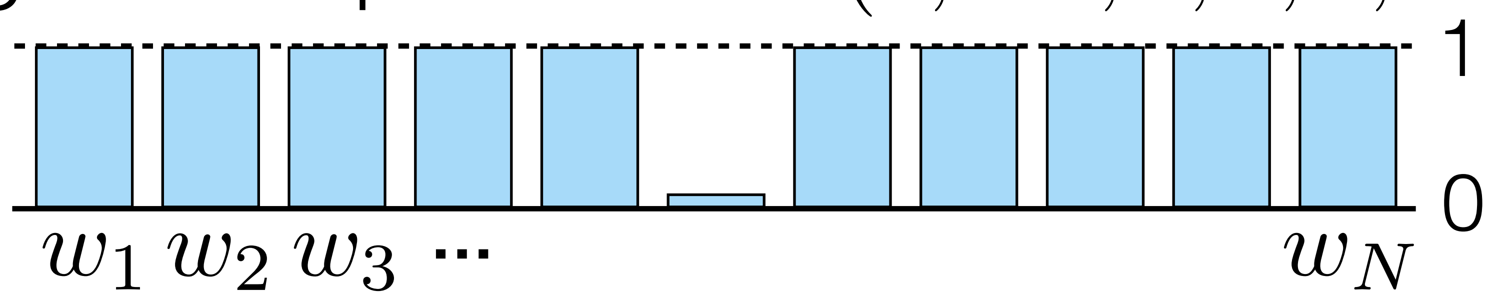
  - E.g. max likelihood, min loss
- A quantity of interest $\boxed{\phi}$
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$

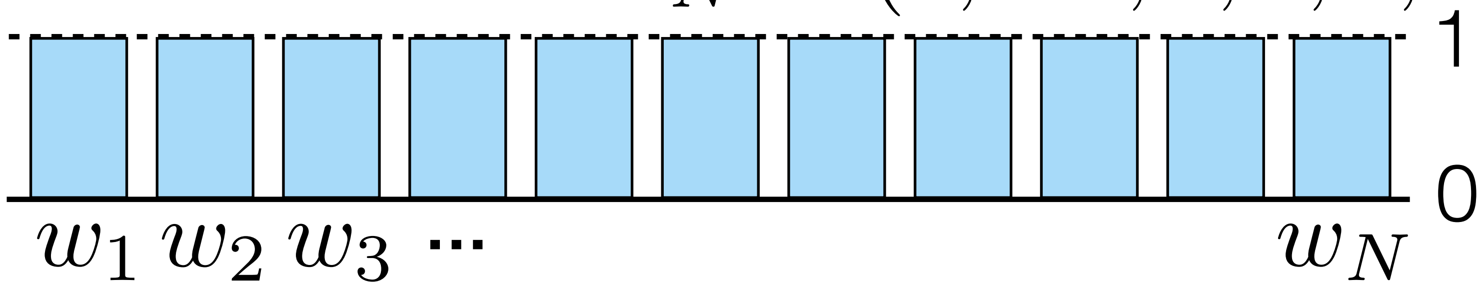
estimator

data point;
e.g. (x_n, y_n)

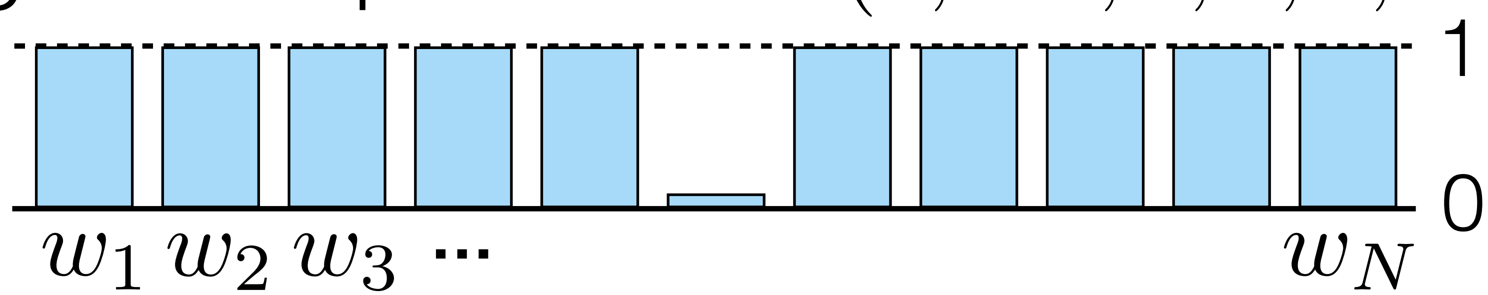
  - E.g. max likelihood, min loss
- A quantity of interest $\phi(w)$
  - E.g. $\phi = \hat{\theta}_p$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$

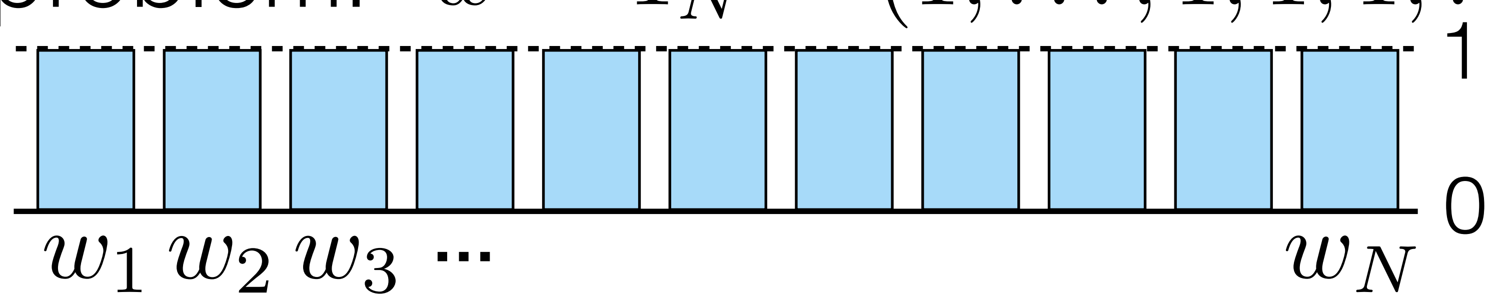
estimator

data point; e.g. (x_n, y_n)

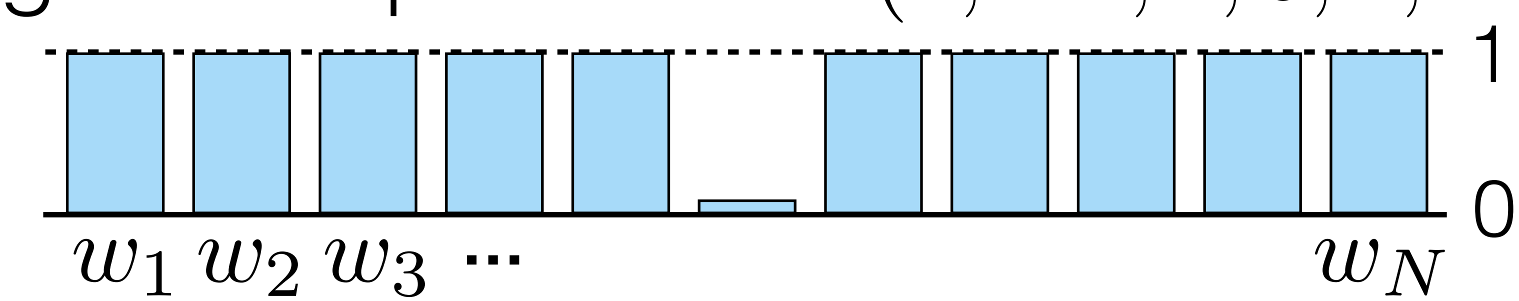
  - E.g. max likelihood, min loss
- A quantity of interest $\phi(w)$
  - E.g. $\boxed{\phi = \hat{\theta}_p}$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$

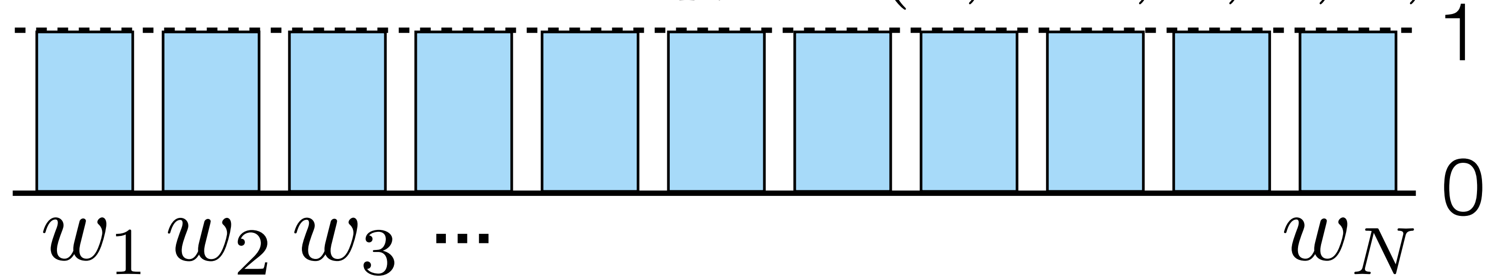
estimator

data point; e.g. (x_n, y_n)

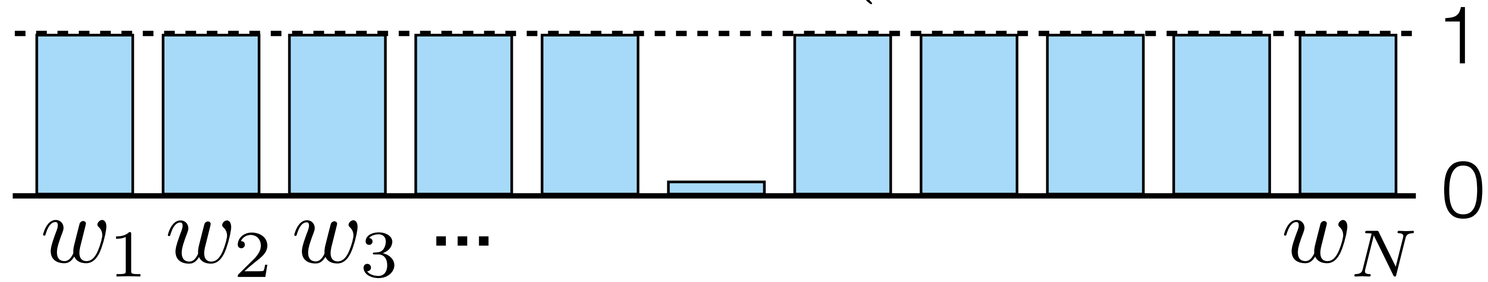
  - E.g. max likelihood, min loss
- A quantity of interest $\phi(w)$
  - E.g. $\phi(w) = \hat{\theta}_p(w)$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$

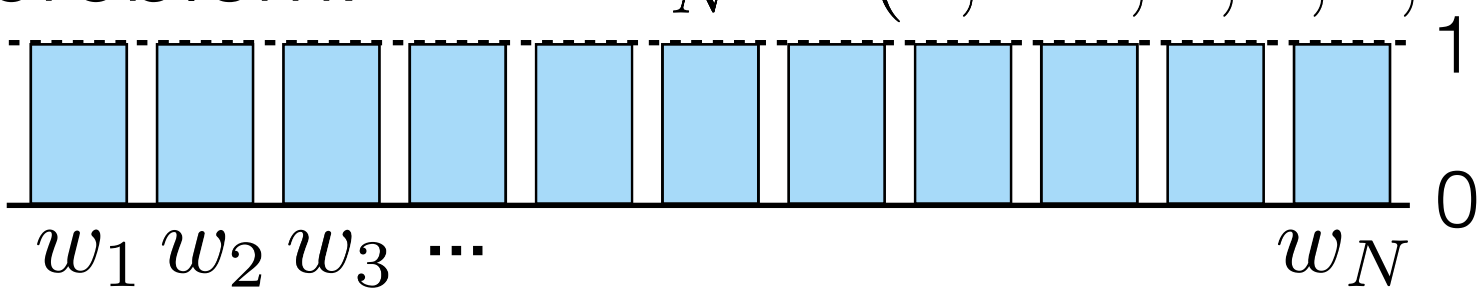
estimator

data point; e.g. (x_n, y_n)

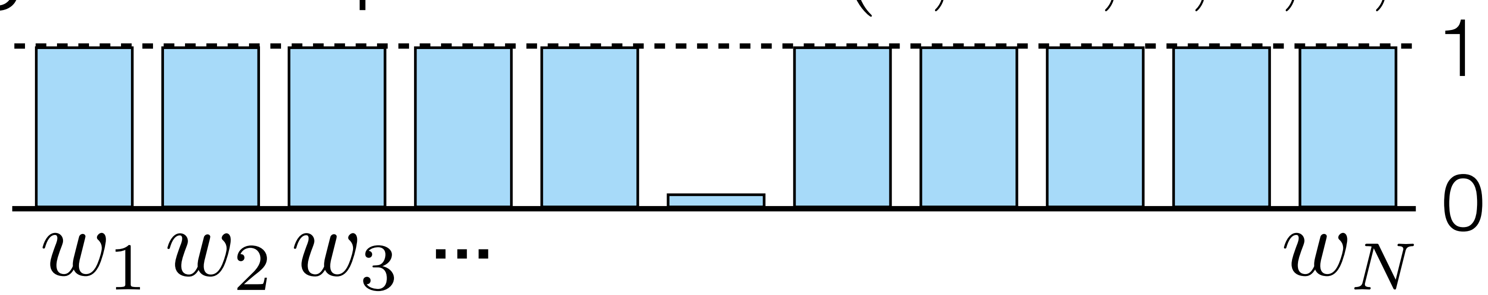
  - E.g. max likelihood, min loss
- A quantity of interest $\phi(w)$
  - E.g. $\phi(w) = \hat{\theta}_p(w)$
  - E.g. $\phi = \hat{\theta}_p - 1.96\sigma_p$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

Setup for dropping data

loss parameters

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$

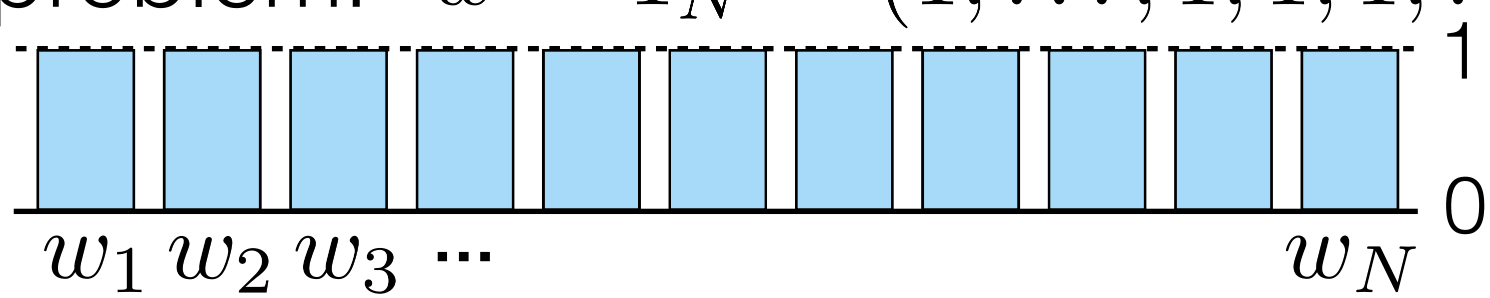
estimator

data point; e.g. (x_n, y_n)

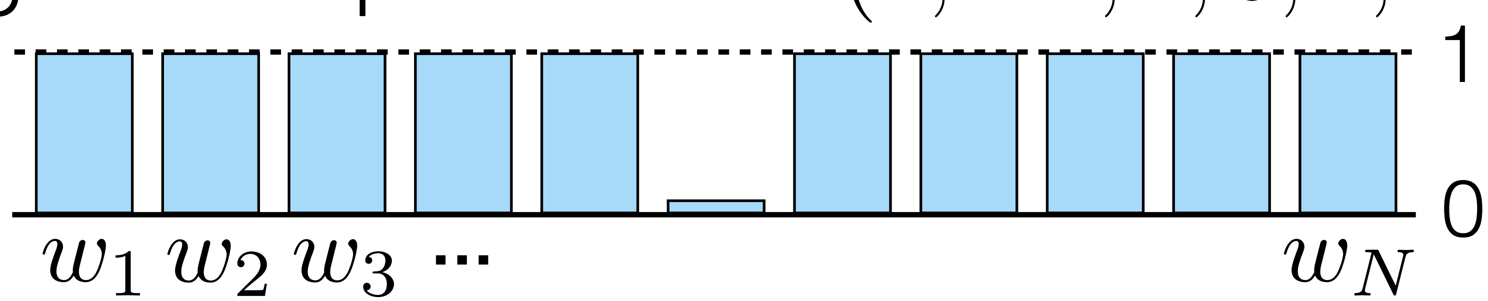
  - E.g. max likelihood, min loss
- A quantity of interest $\phi(w)$
  - E.g. $\phi(w) = \hat{\theta}_p(w)$
  - E.g. $\phi(w) = \hat{\theta}_p(w) - 1.96\sigma_p(\hat{\theta}(w), w)$

(Our approach actually handles even more general analyses)

- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Dropping a data point: $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- Each dropped data subset corresponds to a different w

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\boxed{\phi(w)} \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \boxed{\phi^{\text{lin}}(w)} := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

our approximation

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

our approximation

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \boxed{\sum_{n=1}^{N} (w_n - 1)\psi_n}, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

our approximation

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} \boxed{(w_n - 1)} \psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

our approximation

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

our approximation

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

our approximation

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

our approximation

influence score of the *n*th data point

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

our approximation

influence score of the *n*th data point

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

our approximation *influence score* of the *n*th data point

- We can write formula for the influence score with the implicit function theorem and chain rule

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N}(w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

our approximation — *influence score* of the *n*th data point

- We can write formula for the influence score with the implicit function theorem and chain rule
  - We compute this formula with automatic differentiation (note: very different from numerical or symbolic diff)

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

our approximation   *influence score* of the *n*th data point

- We can write formula for the influence score with the implicit function theorem and chain rule
    - We compute this formula with automatic differentiation (note: very different from numerical or symbolic diff)

[Baydin et al 2018: “Automatic Differentiation in Machine Learning: a Survey”]

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

our approximation

influence score of the *n*th data point

- We can write formula for the influence score with the implicit function theorem and chain rule
  - We compute this formula with automatic differentiation (note: very different from numerical or symbolic diff)
- Need to run **only 1** data analysis

[Baydin et al 2018: “Automatic Differentiation in Machine Learning: a Survey”]

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

our approximation

influence score of the *n*th data point

- We can write formula for the influence score with the implicit function theorem and chain rule
  - We compute this formula with automatic differentiation (note: very different from numerical or symbolic diff)
- Need to run **only 1** data analysis
- Can handle more general cases (e.g. multistage & priors)

[Baydin et al 2018: “Automatic Differentiation in Machine Learning: a Survey”]

How to approximate dropping data

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

exact

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

our approximation *influence score* of the *n*th data point

- We can write formula for the influence score with the implicit function theorem and chain rule
  - We compute this formula with automatic differentiation (note: very different from numerical or symbolic diff)
- Need to run **only 1** data analysis
- Can handle more general cases (e.g. multistage & priors)

[Baydin et al 2018: “Automatic Differentiation in Machine Learning: a Survey”]
[What about higher orders? Some very early work: Giordano, Jordan, Broderick 2019]

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \underbrace{\psi_n}_{\text{influence score}} := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\text{lin}}(w) - \phi(1_N)$

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

influence score

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\text{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n$

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\text{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\mathrm{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n : w_n = 0} -\psi_n$

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\mathrm{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\mathrm{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$
- *Algorithm*:

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

influence score

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\text{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$
- *Algorithm*: Compute influence scores

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \overset{\text{influence score}}{\psi_n} := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\mathrm{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$
- *Algorithm*: Compute influence scores; sort

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\text{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$
- *Algorithm*: Compute influence scores; sort; remove largest

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\text{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$
- *Algorithm*: Compute influence scores; sort; remove largest
- **Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\mathrm{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left. \frac{\partial \phi(w)}{\partial w_n} \right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\mathrm{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$
- *Algorithm*: Compute influence scores; sort; remove largest
- **Approximate Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^{N} (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\text{lin}}(w) - \phi(1_N) = \sum_{n=1}^{N} (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$
- *Algorithm*: Compute influence scores; sort; remove largest
- **Approximate Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
- **Approx Most Influential Set**: data dropped to get AMIP

Why does the approximation help?

- A data analysis: $\hat{\theta}(w) := \operatorname*{argmin}_\theta \sum_{n=1}^N w_n f(\theta, d_n)$
- Quantity of interest: $\phi(w)$, e.g. $\phi(w) = \hat{\theta}_p(w)$
- Idea: (first-order) Taylor expansion around $w = 1_N$:

influence score

$$\phi(w) \approx \phi^{\text{lin}}(w) := \phi(1_N) + \sum_{n=1}^N (w_n - 1)\psi_n, \quad \psi_n := \left.\frac{\partial \phi(w)}{\partial w_n}\right|_{w=1_N}$$

- Exact change: $\phi(w) - \phi(1_N)$
- Approx: $\phi^{\text{lin}}(w) - \phi(1_N) = \sum_{n=1}^N (w_n - 1)\psi_n = \sum_{n: w_n = 0} -\psi_n$
- *Algorithm*: Compute influence scores; sort; remove largest
- **Approximate Maximum Influence Perturbation**: largest possible change by dropping at most 100α% of the data
- **Approx Most Influential Set**: data dropped to get AMIP
- **Approximate Perturbation-Inducing Proportion**: Min data proportion to achieve a certain change (`NA` if none)

What makes an analysis non-robust?

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)$$

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)$$

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \qquad x_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_X^2)$$

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \qquad x_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \qquad x_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \qquad x_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

- Can we flip sign of $\hat{\theta}$ by dropping some of 10,000 points?

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \qquad x_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

- Can we flip sign of $\hat{\theta}$ by dropping some of 10,000 points?
- Signal = size of change of interest: $\Delta = \hat{\theta}$

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \qquad x_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

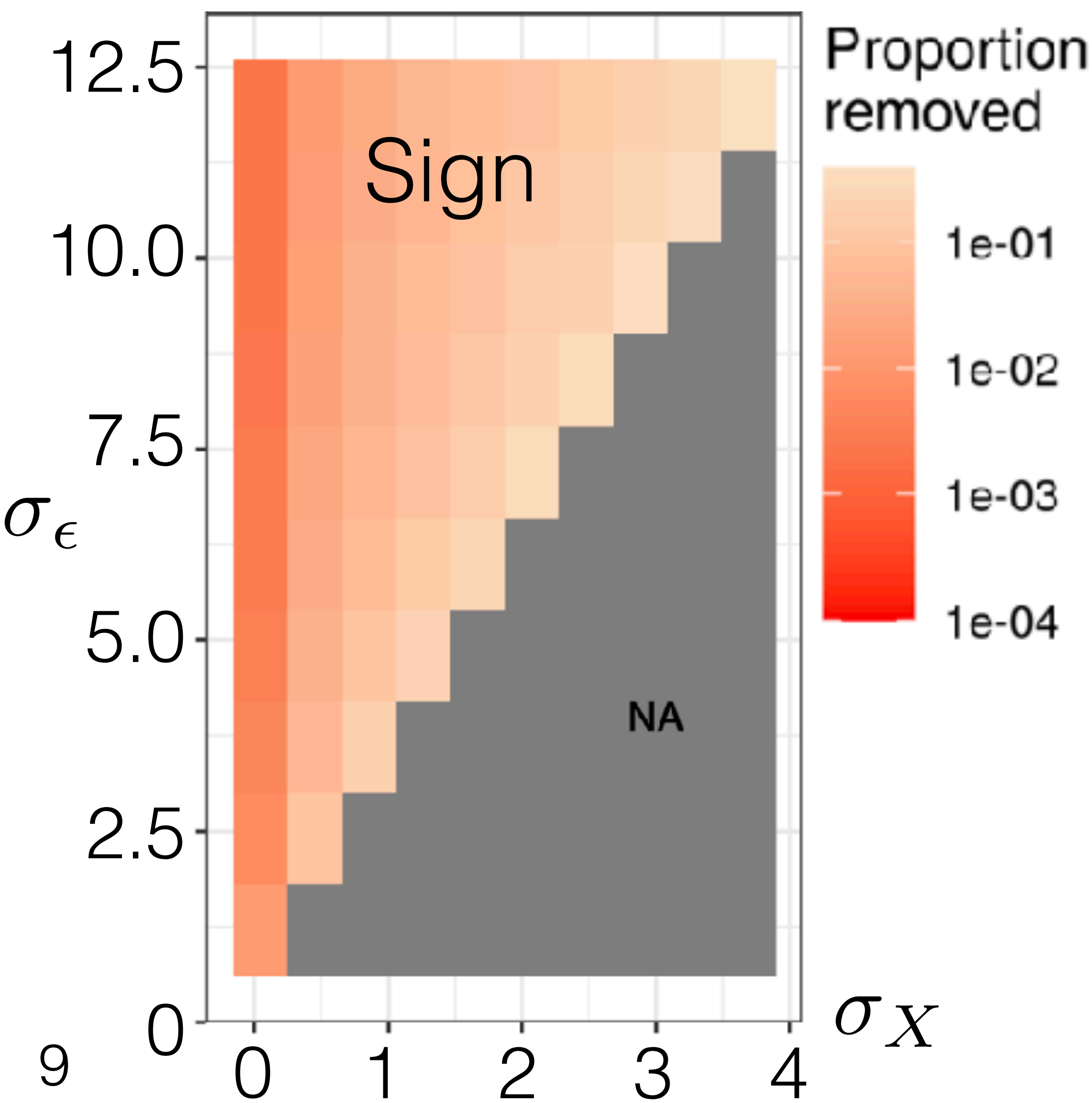
- Can we flip sign of $\hat{\theta}$ by dropping some of 10,000 points?
- Signal = size of change of interest: $\Delta = \hat{\theta}$
- Noise = estimate of the (scaled) asymptotic std dev: $\approx \frac{\sigma_\epsilon}{\sigma_X}$

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \quad x_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

- Can we flip sign of $\hat{\theta}$ by dropping some of 10,000 points?
- Signal = size of change of interest: $\Delta = \hat{\theta}$
- Noise = estimate of the (scaled) asymptotic std dev: $\approx \frac{\sigma_\epsilon}{\sigma_X}$

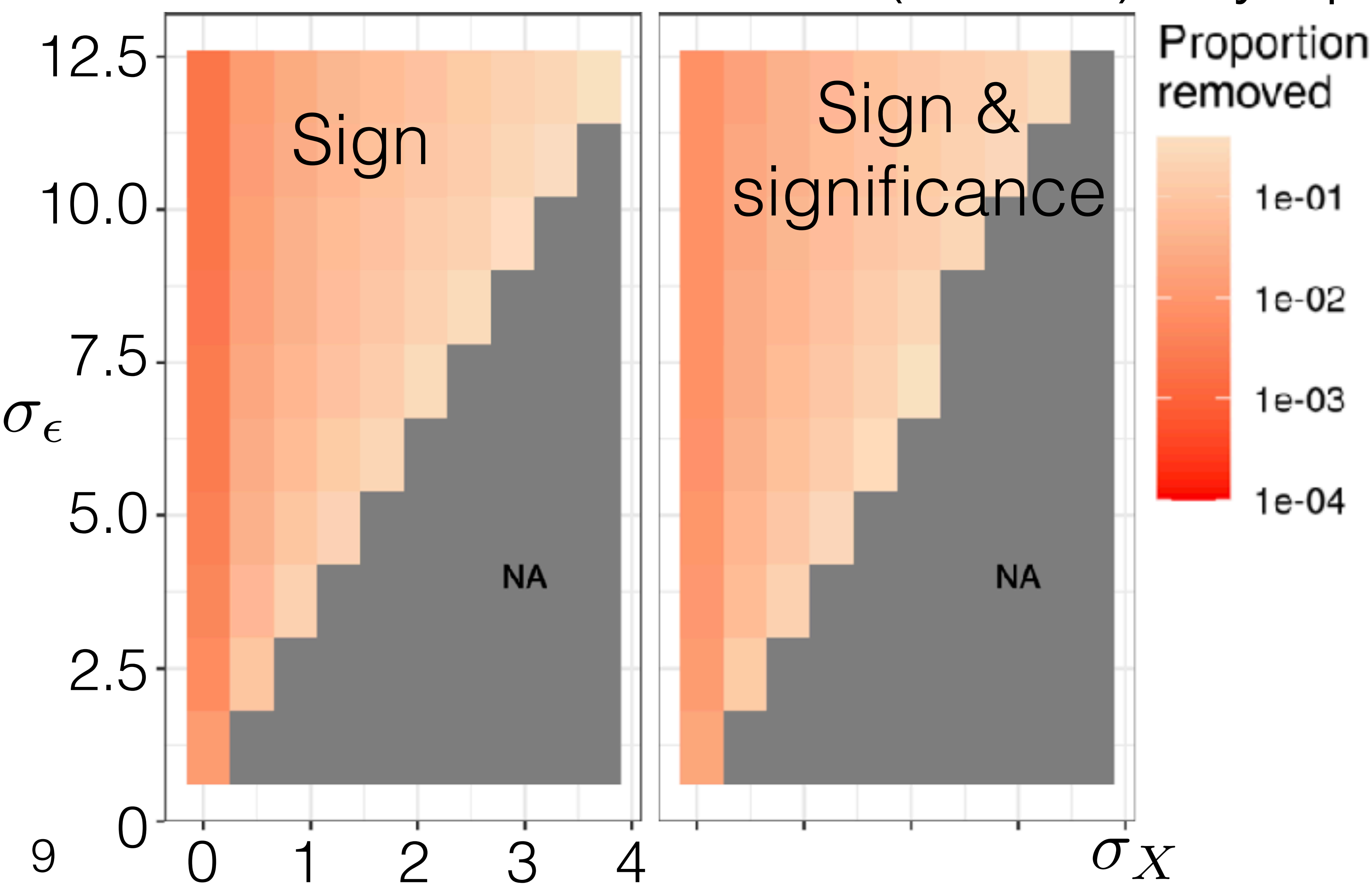


What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \quad x_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

- Can we flip sign of $\hat{\theta}$ by dropping some of 10,000 points?
- Signal = size of change of interest: $\Delta = \hat{\theta}$
- Noise = estimate of the (scaled) asymptotic std dev: $\approx \frac{\sigma_\epsilon}{\sigma_X}$

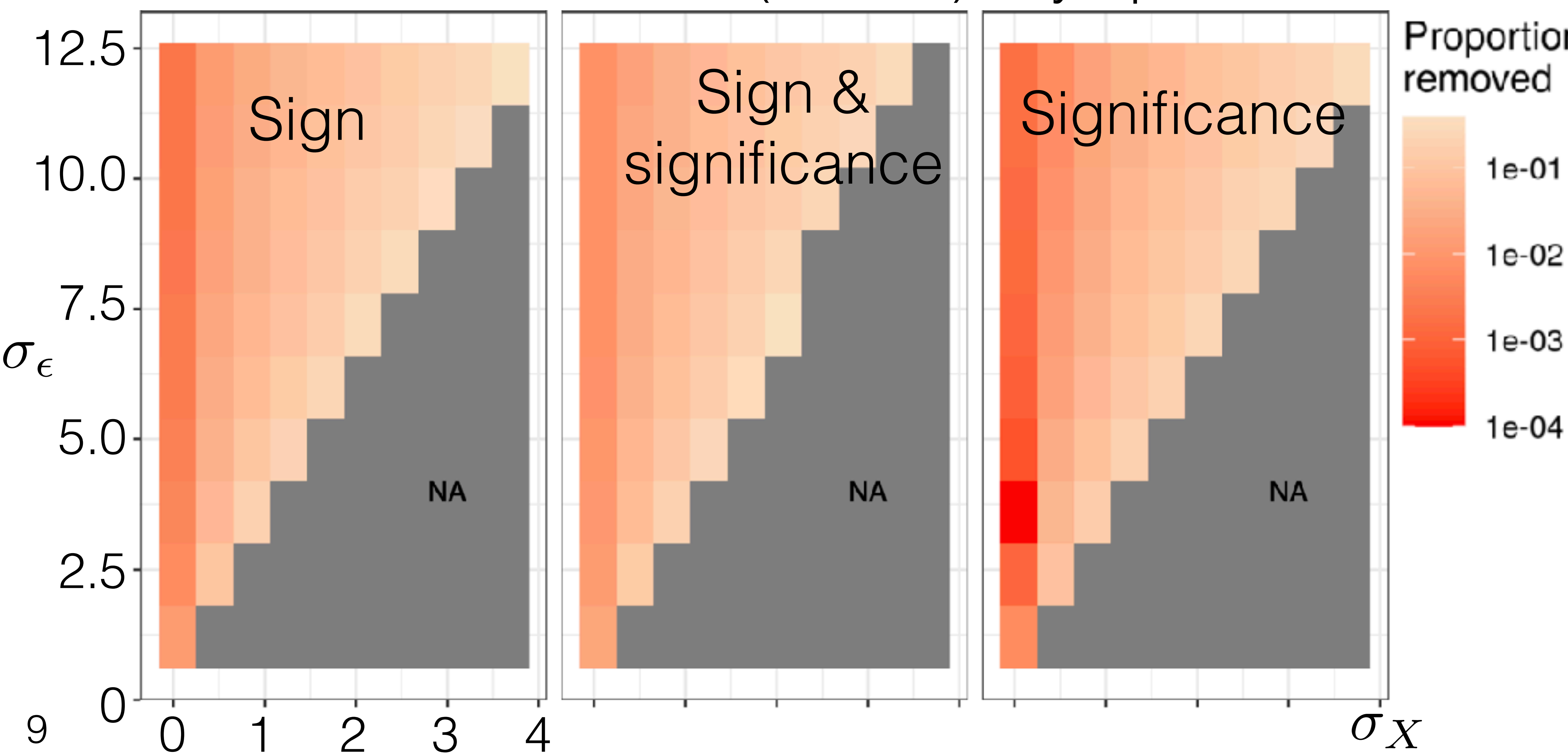


What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \qquad x_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

- Can we flip sign of $\hat{\theta}$ by dropping some of 10,000 points?
- Signal = size of change of interest: $\Delta = \hat{\theta}$
- Noise = estimate of the (scaled) asymptotic std dev: $\approx \dfrac{\sigma_\epsilon}{\sigma_X}$

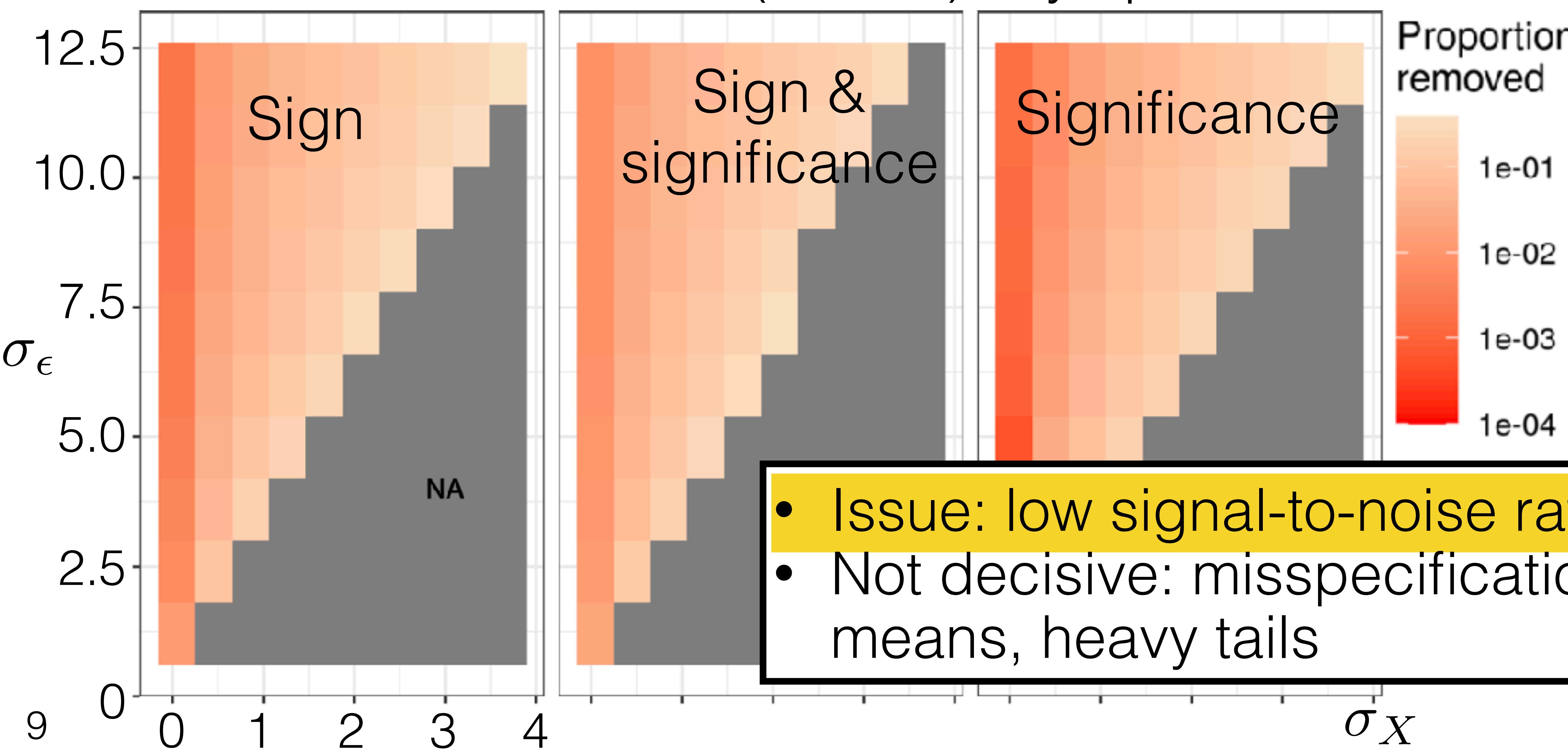


What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \quad x_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

- Can we flip sign of $\hat{\theta}$ by dropping some of 10,000 points?
- Signal = size of change of interest: $\Delta = \hat{\theta}$
- Noise = estimate of the (scaled) asymptotic std dev: $\approx \frac{\sigma_\epsilon}{\sigma_X}$



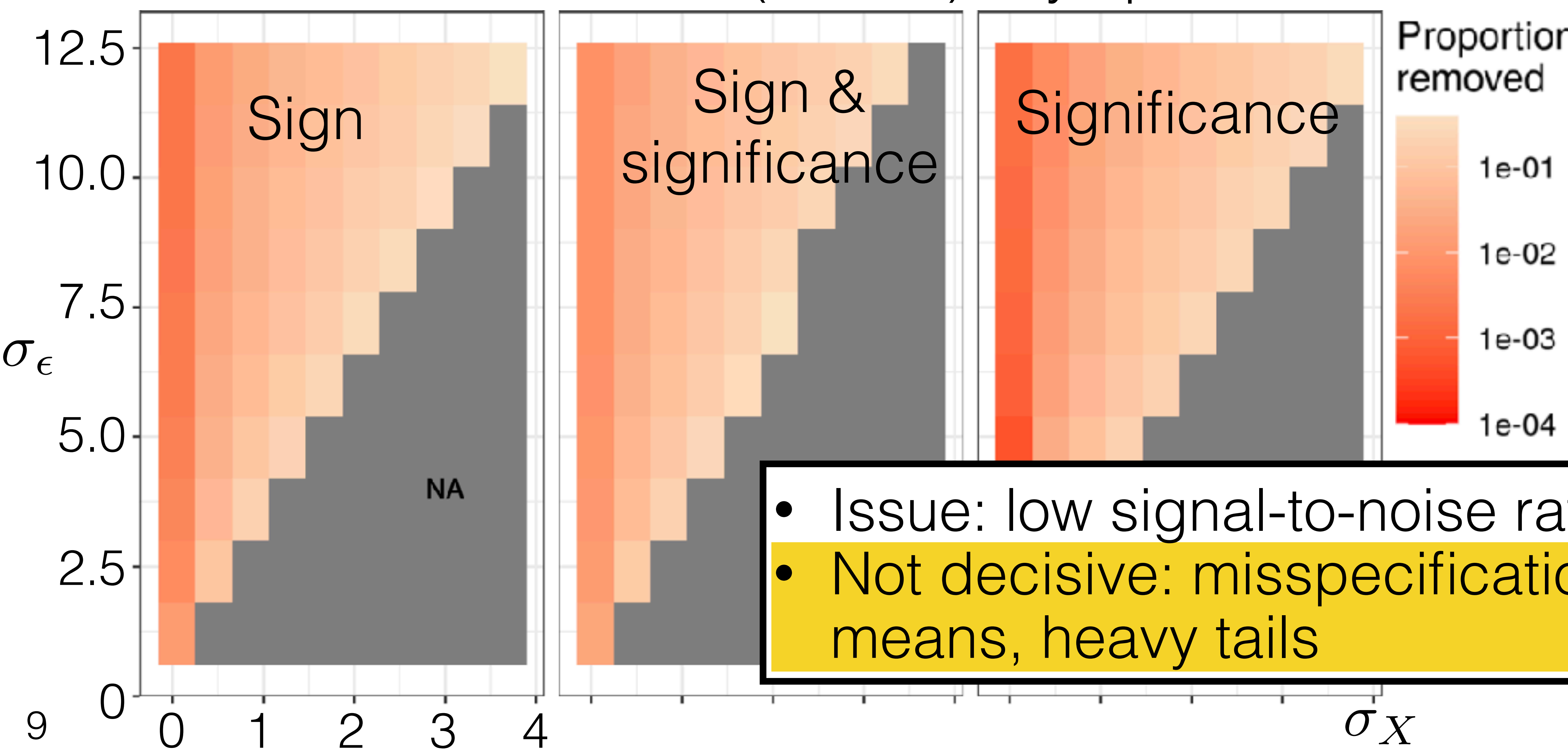
- Issue: low signal-to-noise ratio
- Not decisive: misspecification, means, heavy tails

What makes an analysis non-robust?

- Simulations from linear model with Gaussian noise

$$y_n = \theta x_n + \epsilon_n, \quad \epsilon_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \quad x_n \overset{iid}{\sim} \mathcal{N}(0, \sigma_X^2), \quad \theta = -1$$

- Can we flip sign of $\hat{\theta}$ by dropping some of 10,000 points?
- Signal = size of change of interest: $\Delta = \hat{\theta}$
- Noise = estimate of the (scaled) asymptotic std dev: $\approx \frac{\sigma_\epsilon}{\sigma_X}$



- Issue: low signal-to-noise ratio
- Not decisive: misspecification, means, heavy tails

Oregon Medicaid Study

Oregon Medicaid Study

- Small p-value is not decisive

Oregon Medicaid Study

- Small p-value is not decisive
- Finkelstein et al 2012: *again, fantastic reproducibility!*

Oregon Medicaid Study

- Small p-value is not decisive
- Finkelstein et al 2012: *again, fantastic reproducibility!*
    - Lottery in Oregon; winners could sign up for Medicaid

Oregon Medicaid Study

- Small p-value is not decisive
- Finkelstein et al 2012: *again, fantastic reproducibility!*
  - Lottery in Oregon; winners could sign up for Medicaid
  - Effect of lottery on health, e.g. impaired activity 30 days

Oregon Medicaid Study

- Small p-value is not decisive
- Finkelstein et al 2012: *again, fantastic reproducibility!*
  - Lottery in Oregon; winners could sign up for Medicaid
  - Effect of lottery on health, e.g. impaired activity 30 days
  - >21,000 data points (survey responders)

Oregon Medicaid Study

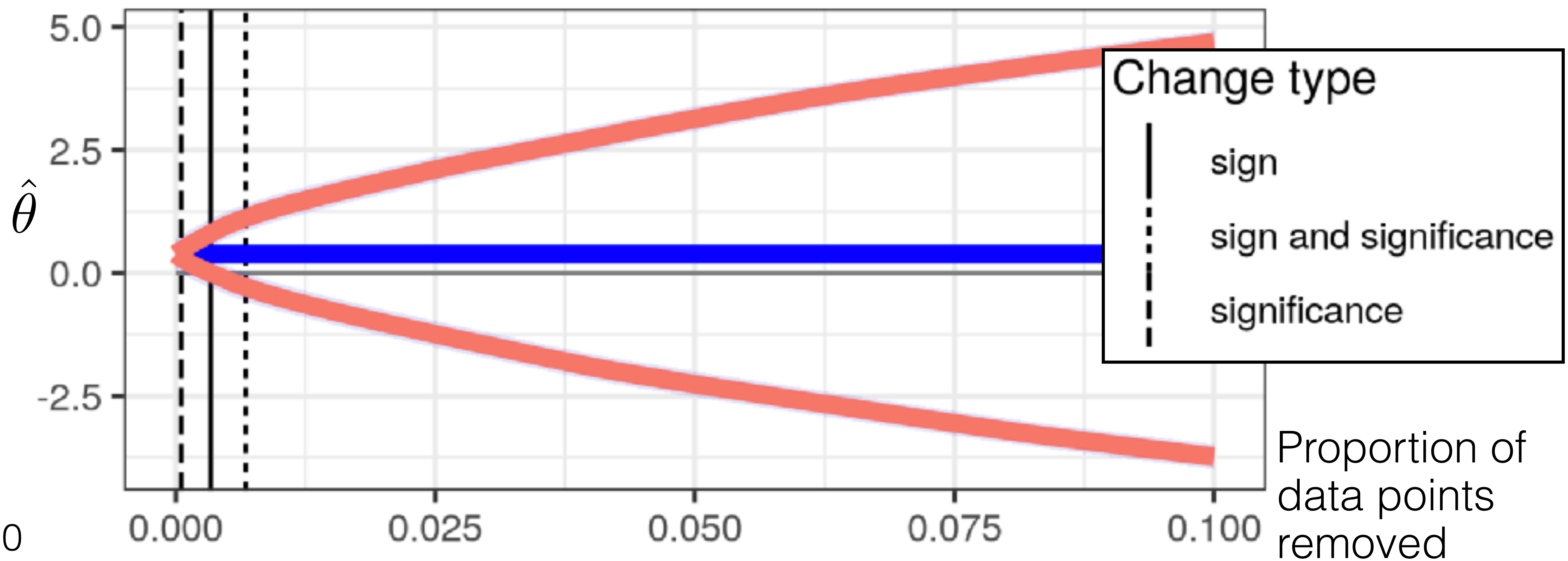
- Small p-value is not decisive
- Finkelstein et al 2012: *again, fantastic reproducibility!*
  - Lottery in Oregon; winners could sign up for Medicaid
  - Effect of lottery on health, e.g. impaired activity 30 days
  - >21,000 data points (survey responders)
  - $p < 0.01$ for a positive effect

Oregon Medicaid Study

- Small p-value is not decisive
- Finkelstein et al 2012: *again, fantastic reproducibility!*
  - Lottery in Oregon; winners could sign up for Medicaid
  - Effect of lottery on health, e.g. impaired activity 30 days
  - >21,000 data points (survey responders)
  - $p < 0.01$ for a positive effect
    - But dropping 11 points (0.05%) changes significance

Oregon Medicaid Study

- Small p-value is not decisive
- Finkelstein et al 2012: *again, fantastic reproducibility!*
  - Lottery in Oregon; winners could sign up for Medicaid
  - Effect of lottery on health, e.g. impaired activity 30 days
  - >21,000 data points (survey responders)
  - $p < 0.01$ for a positive effect
    - But dropping 11 points (0.05%) changes significance



Proportion of data points removed

Cash transfers

Cash transfers

- Can be robust! Removing outliers isn't a panacea

Cash transfers

- Can be robust! Removing outliers isn't a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*

Cash transfers

- Can be robust! Removing outliers isn't a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*
    - Direct effect of cash transfers for poor households on household consumption

Cash transfers

- Can be robust! Removing outliers isn't a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*
  - Direct effect of cash transfers for poor households on household consumption
  - "Spillover" effect: non-poor households in same village

Cash transfers

- Can be robust! Removing outliers isn’t a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*
  - Direct effect of cash transfers for poor households on household consumption
  - “Spillover” effect: non-poor households in same village
- Poor households, >10,000 data points

Cash transfers

- Can be robust! Removing outliers isn't a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*
  - Direct effect of cash transfers for poor households on household consumption
  - "Spillover" effect: non-poor households in same village
- Poor households, >10,000 data points
  - Must drop 4–10% data to change sign/significance/both

Cash transfers

- Can be robust! Removing outliers isn’t a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*
  - Direct effect of cash transfers for poor households on household consumption
  - “Spillover” effect: non-poor households in same village
- Poor households, >10,000 data points
  - Must drop 4–10% data to change sign/significance/both
- Spillover, >4,000 data points

Cash transfers

- Can be robust! Removing outliers isn't a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*
  - Direct effect of cash transfers for poor households on household consumption
  - "Spillover" effect: non-poor households in same village
- Poor households, >10,000 data points
  - Must drop 4–10% data to change sign/significance/both
- Spillover, >4,000 data points
  - Original analysis deleted households with consumption greater than 10,000 units (i.e. largest response)

Cash transfers

- Can be robust! Removing outliers isn't a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*
  - Direct effect of cash transfers for poor households on household consumption
  - "Spillover" effect: non-poor households in same village
- Poor households, >10,000 data points
  - Must drop 4–10% data to change sign/significance/both
- Spillover, >4,000 data points
  - Original analysis deleted households with consumption greater than 10,000 units (i.e. largest response)
  - Still sensitive: can drop 3 points to change significance

Cash transfers

- Can be robust! Removing outliers isn’t a panacea
- Angelucci and De Giorgi (2009): *awesomely reproducible!*
  - Direct effect of cash transfers for poor households on household consumption
  - “Spillover” effect: non-poor households in same village
- Poor households, >10,000 data points
  - Must drop 4–10% data to change sign/significance/both
- Spillover, >4,000 data points
  - Original analysis deleted households with consumption greater than 10,000 units (i.e. largest response)
  - Still sensitive: can drop 3 points to change significance
  - We show: in linear regression, influence score = residual times leverage

Connections

Connections

- Again: these concerns are *not* specific to econometrics

Connections

- Again: these concerns are *not* specific to econometrics
- Our approximation is local; complementary to global checks [Leamer 1984, 1985; Sobol 2001; Saltelli 2004; He et al. 1990; Masten and Poirier 2020]

Connections

- Again: these concerns are *not* specific to econometrics
- Our approximation is local; complementary to global checks [Leamer 1984, 1985; Sobol 2001; Saltelli 2004; He et al. 1990; Masten and Poirier 2020]
- Complementary to other types of robustness (including Huber or tailored checks)

Connections

- Again: these concerns are *not* specific to econometrics
- Our approximation is local; complementary to global checks [Leamer 1984, 1985; Sobol 2001; Saltelli 2004; He et al. 1990; Masten and Poirier 2020]
- Complementary to other types of robustness (including Huber or tailored checks)
    - & to robustification procedures

Connections

- Again: these concerns are *not* specific to econometrics
- Our approximation is local; complementary to global checks [Leamer 1984, 1985; Sobol 2001; Saltelli 2004; He et al. 1990; Masten and Poirier 2020]
- Complementary to other types of robustness (including Huber or tailored checks)
    - & to robustification procedures
- A growing literature on approximate cross-validation and use of influence functions for practical checks [Obuchi and Kabashima, 2016, 2018; Beirami, Razaviyayn, Shahrampour, Tarokh 2017; Rad and Maleki 2020; Wang, Zhou, Lu, Maleki, Mirrokni 2018; Koh and Liang 2017; Koh, Ang, Teo, and Liang 2019]

Connections

- Again: these concerns are *not* specific to econometrics
- Our approximation is local; complementary to global checks [Leamer 1984, 1985; Sobol 2001; Saltelli 2004; He et al. 1990; Masten and Poirier 2020]
- Complementary to other types of robustness (including Huber or tailored checks)
    - & to robustification procedures
- A growing literature on approximate cross-validation and use of influence functions for practical checks [Obuchi and Kabashima, 2016, 2018; Beirami, Razaviyayn, Shahrampour, Tarokh 2017; Rad and Maleki 2020; Wang, Zhou, Lu, Maleki, Mirrokni 2018; Koh and Liang 2017; Koh, Ang, Teo, and Liang 2019; Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Connections

- Again: these concerns are *not* specific to econometrics
- Our approximation is local; complementary to global checks [Leamer 1984, 1985; Sobol 2001; Saltelli 2004; He et al. 1990; Masten and Poirier 2020]
- Complementary to other types of robustness (including Huber or tailored checks)
    - & to robustification procedures
- A growing literature on approximate cross-validation and use of influence functions for practical checks [Obuchi and Kabashima, 2016, 2018; Beirami, Razaviyayn, Shahrampour, Tarokh 2017; Rad and Maleki 2020; Wang, Zhou, Lu, Maleki, Mirrokni 2018; Koh and Liang 2017; Koh, Ang, Teo, and Liang 2019; Giordano, Stephenson, Liu, Jordan, Broderick 2019; Stephenson, Broderick 2020; Stephenson, Udell, Broderick 2020]

Connections

- Again: these concerns are *not* specific to econometrics
- Our approximation is local; complementary to global checks [Leamer 1984, 1985; Sobol 2001; Saltelli 2004; He et al. 1990; Masten and Poirier 2020]
- Complementary to other types of robustness (including Huber or tailored checks)
    - & to robustification procedures
- A growing literature on approximate cross-validation and use of influence functions for practical checks [Obuchi and Kabashima, 2016, 2018; Beirami, Razaviyayn, Shahrampour, Tarokh 2017; Rad and Maleki 2020; Wang, Zhou, Lu, Maleki, Mirrokni 2018; Koh and Liang 2017; Koh, Ang, Teo, and Liang 2019; Giordano, Stephenson, Liu, Jordan, Broderick 2019; Stephenson, Broderick 2020; Stephenson, Udell, Broderick 2020; Ghosh*, Stephenson*, Nguyen, Desphande, Broderick 2020]

Connections

- Again: these concerns are *not* specific to econometrics
- Our approximation is local; complementary to global checks [Leamer 1984, 1985; Sobol 2001; Saltelli 2004; He et al. 1990; Masten and Poirier 2020]
- Complementary to other types of robustness (including Huber or tailored checks)
    - & to robustification procedures
- A growing literature on approximate cross-validation and use of influence functions for practical checks [Obuchi and Kabashima, 2016, 2018; Beirami, Razaviyayn, Shahrampour, Tarokh 2017; Rad and Maleki 2020; Wang, Zhou, Lu, Maleki, Mirrokni 2018; Koh and Liang 2017; Koh, Ang, Teo, and Liang 2019; Giordano, Stephenson, Liu, Jordan, Broderick 2019; Stephenson, Broderick 2020; Stephenson, Udell, Broderick 2020; Ghosh*, Stephenson*, Nguyen, Desphande, Broderick 2020]
    - Cf. the classical "infinitesimal jackknife" [Jaeckel 1972; Clarke 1983]

Try it out!

- We present a metric to check if there is a small fraction of data you can drop to change conclusions
- **Paper**: T Broderick, R Giordano, R Meager “An Automatic Finite-Sample Robustness Metric: Can Dropping a Little Data Change Conclusions?” 2020
  `https://arxiv.org/abs/2011.14999`
- **Code, readme, and examples**:
  `https://github.com/rgiordan/zaminfluence`
- Try it out on your data analysis and email us!
  `tbroderick@mit.edu, rgiordan@mit.edu, r.meager@lse.ac.uk`
- See also: “Transparency and Reproducibility in Artificial Intelligence,” *Nature Matters Arising*, 2020.
- Introduction to ML: `tamarabroderick.com/ml.html`

Try it out!

- We present a metric to check if there is a small fraction of data you can drop to change conclusions
- **Paper**: T Broderick, R Giordano, R Meager "An Automatic Finite-Sample Robustness Metric: Can Dropping a Little Data Change Conclusions?" 2020
  `https://arxiv.org/abs/2011.14999`
- **Code, readme, and examples**:
  `https://github.com/rgiordan/zaminfluence`

See also:

- R Giordano, W Stephenson, R Liu, MI Jordan, T Broderick. A Swiss Army infinitesimal jackknife. *AISTATS* 2019.
- W Stephenson, T Broderick. Approximate Cross-Validation in High Dimensions with Guarantees. *AISTATS* 2020.
- WT Stephenson, M Udell, T Broderick. Approximate Cross-Validation with Low-Rank Data in High Dimensions. *NeurIPS* 2020.
- S Ghosh*, WT Stephenson*, TD Nguyen, SK Deshpande, T Broderick. Approximate Cross-Validation for Structured Models. *NeurIPS* 2020. (*equal)