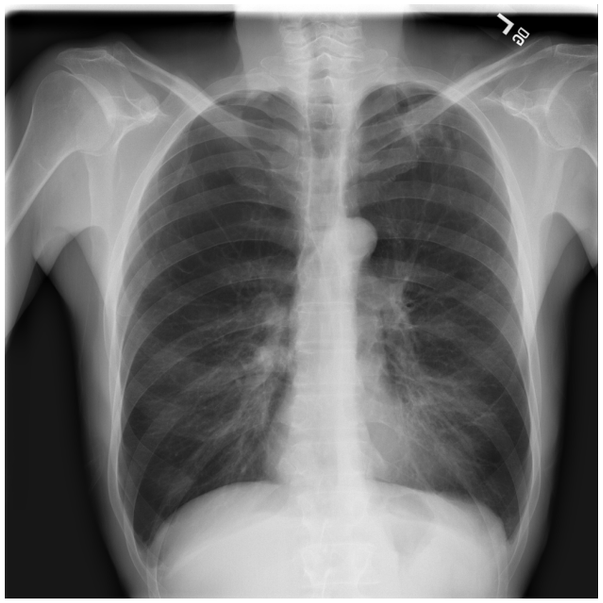
Approximate Cross Validation for Large Data and High Dimensions

Tamara Broderick

Associate Professor,
MIT

Ryan Giordano, William T. Stephenson

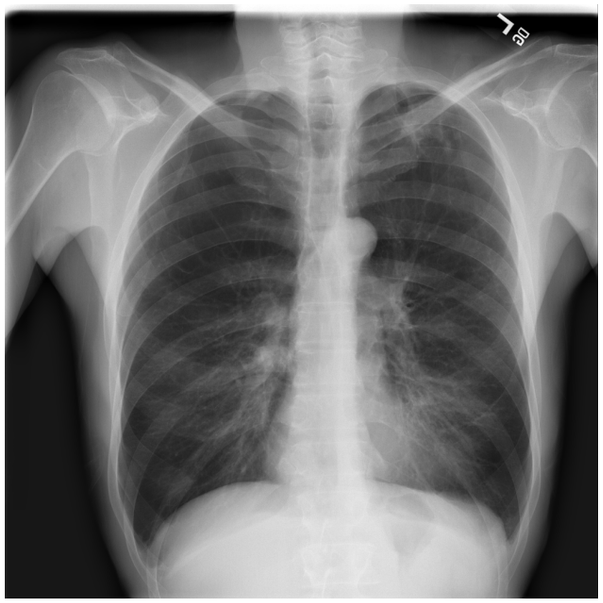
- Better computation ➔ increasingly complex and expensive data analysis in life-changing application areas



[1] [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]

Evaluation

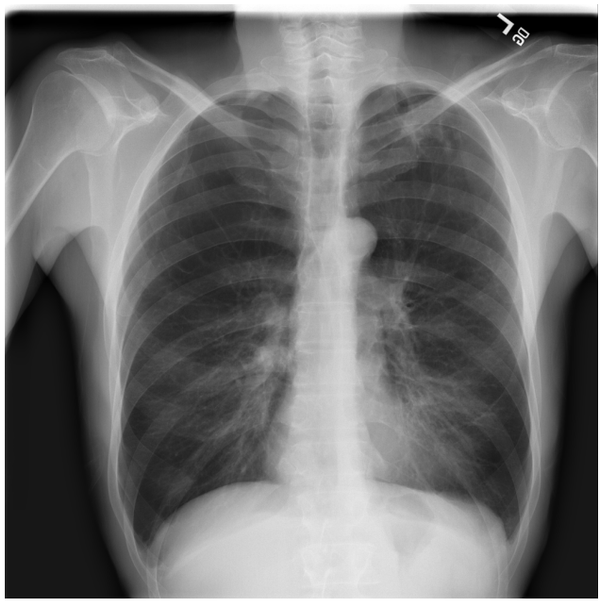
- Better computation ➔ increasingly complex and expensive data analysis in life-changing application areas



[1] [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]

Evaluation

- Better computation ➔ increasingly complex and expensive data analysis in life-changing application areas

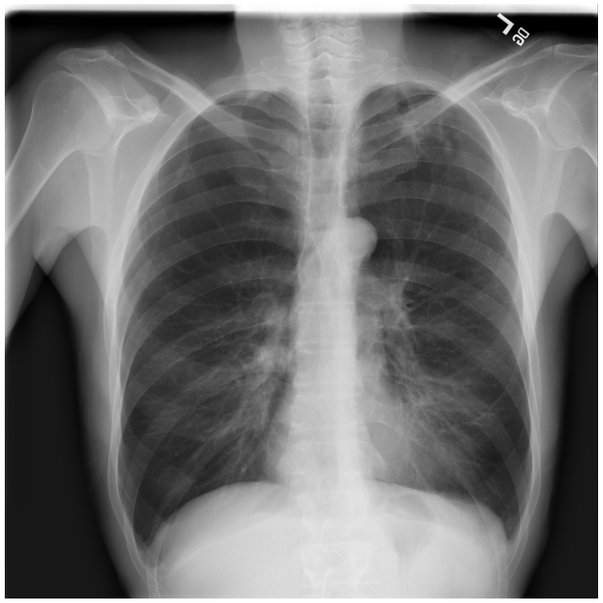


- How well does the data analysis work?

[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]

Evaluation

- Better computation ➔ increasingly complex and expensive data analysis in life-changing application areas

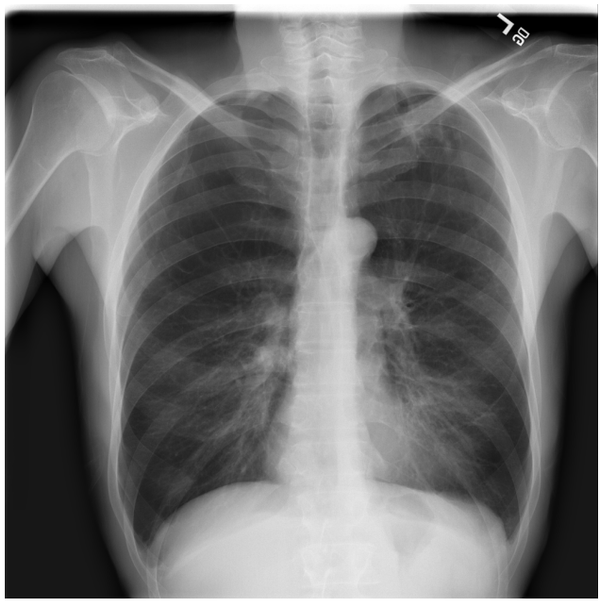


- How well does the data analysis work? Evaluation concerns: accuracy, compute time, & user time

[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]

Evaluation

- Better computation ➔ increasingly complex and expensive data analysis in life-changing application areas

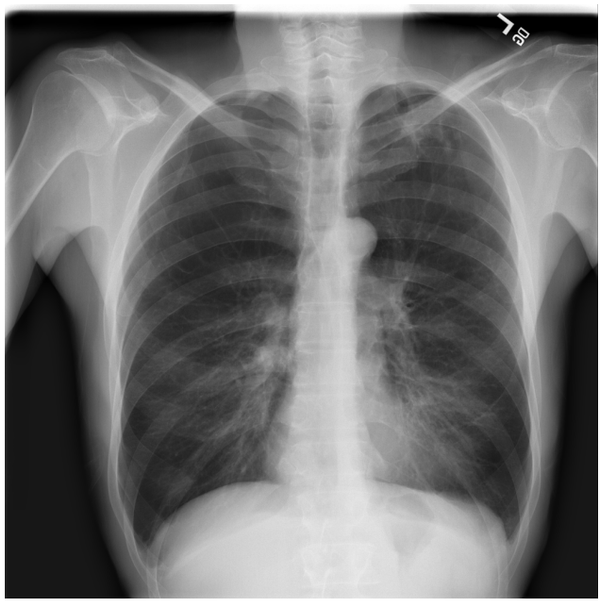


- How well does the data analysis work? Evaluation concerns: accuracy, compute time, & user time
- Cross validation seems to work well, is model agnostic, but can be expensive (multiple analysis runs)

[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]

Evaluation

- Better computation ➔ increasingly complex and expensive data analysis in life-changing application areas

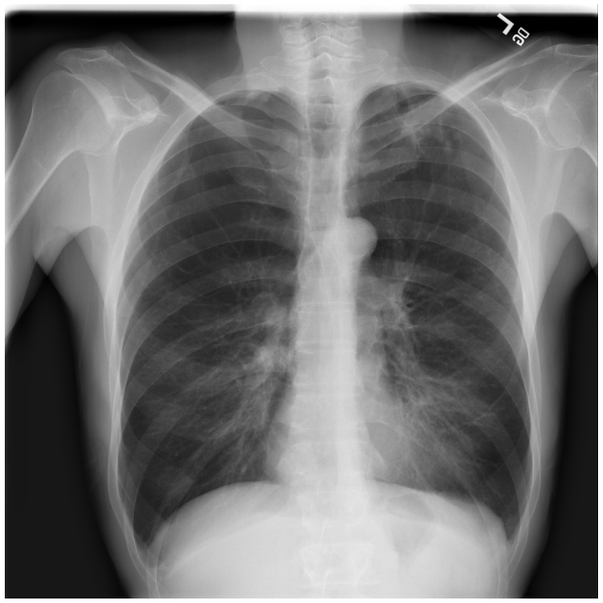


- How well does the data analysis work? Evaluation concerns: accuracy, compute time, & user time
- Cross validation seems to work well, is model agnostic, but can be expensive (multiple analysis runs)
- Can *approximate* CV: Taylor expand + reduce dimension
  - We show: is **fast**, **automatic**, and provably **high-quality**

[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]

Evaluation

- Better computation ➔ increasingly complex and expensive data analysis in life-changing application areas

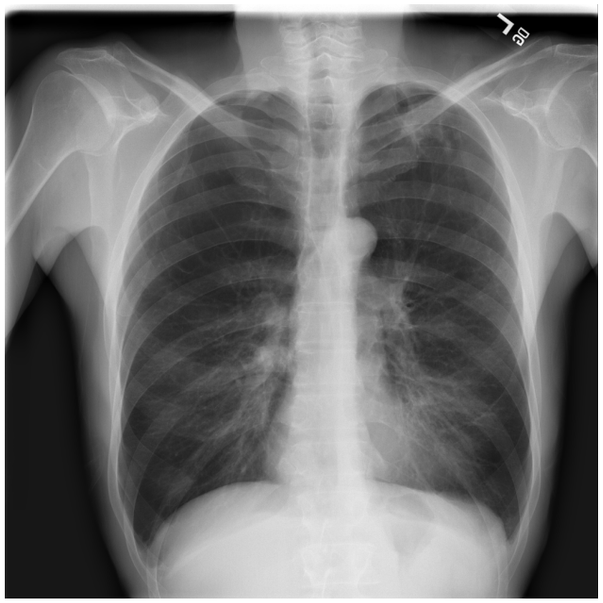


- How well does the data analysis work? Evaluation concerns: accuracy, compute time, & user time
- Cross validation seems to work well, is model agnostic, but can be expensive (multiple analysis runs)
- Can *approximate* CV
  - We show: is **fast**, **automatic**, and provably **high-quality**

[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]

Evaluation

- Better computation ➔ increasingly complex and expensive data analysis in life-changing application areas



- How well does the data analysis work? Evaluation concerns: accuracy, compute time, & user time
- Cross validation seems to work well, is model agnostic, but can be expensive (multiple analysis runs)
- Can *approximate* general resampling/reweighting
  - We show: is **fast**, **automatic**, and provably **high-quality**

[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

- Cross validation (CV) setup

Roadmap

- Cross validation (CV) setup
- CV Challenge #1: an expensive data analysis

Roadmap

- Cross validation (CV) setup
- CV Challenge #1: an expensive data analysis
- Solution #1: a Taylor series approximation

Roadmap

- Cross validation (CV) setup
- CV Challenge #1: an expensive data analysis
- Solution #1: a Taylor series approximation ("Swiss Army infinitesimal jackknife")

Roadmap

- Cross validation (CV) setup
- CV Challenge #1: an expensive data analysis
- Solution #1: a Taylor series approximation ("Swiss Army infinitesimal jackknife")
- CV Challenge #2: high dimensions

Roadmap

- Cross validation (CV) setup
- CV Challenge #1: an expensive data analysis
- Solution #1: a Taylor series approximation ("Swiss Army infinitesimal jackknife")
- CV Challenge #2: high dimensions
- Solution #2: using sparsity (carefully)

Roadmap

- Cross validation (CV) setup
- CV Challenge #1: an expensive data analysis
- Solution #1: a Taylor series approximation ("Swiss Army infinitesimal jackknife")
- CV Challenge #2: high dimensions
- Solution #2: using sparsity (carefully)

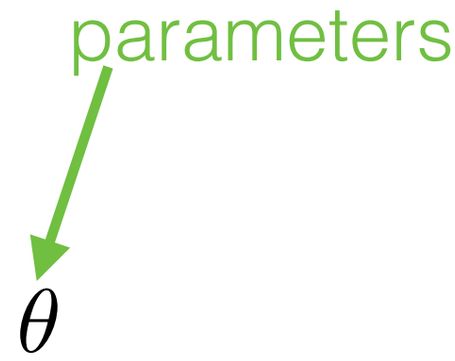
Cross validation setup

Cross validation setup

- A data analysis:

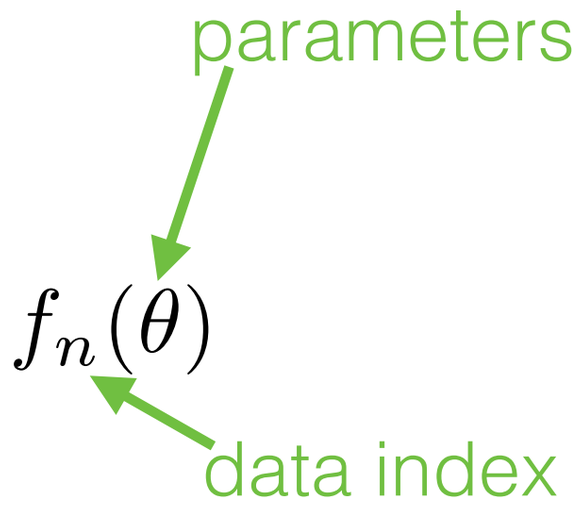
Cross validation setup

- A data analysis:



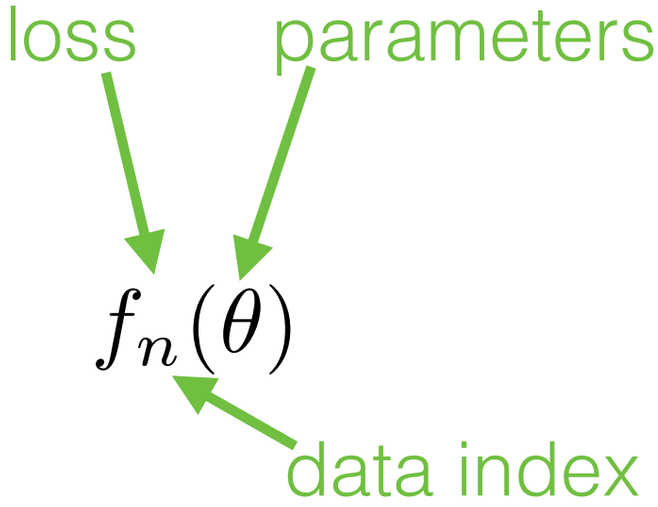
Cross validation setup

- A data analysis:



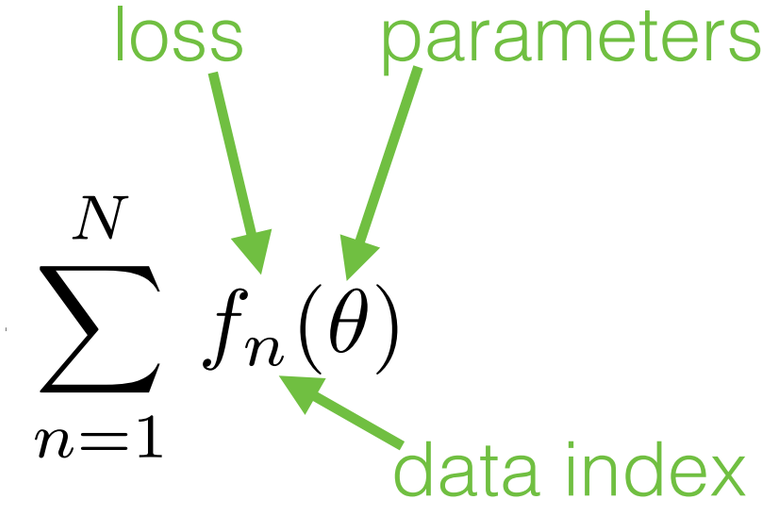
Cross validation setup

- A data analysis:



Cross validation setup

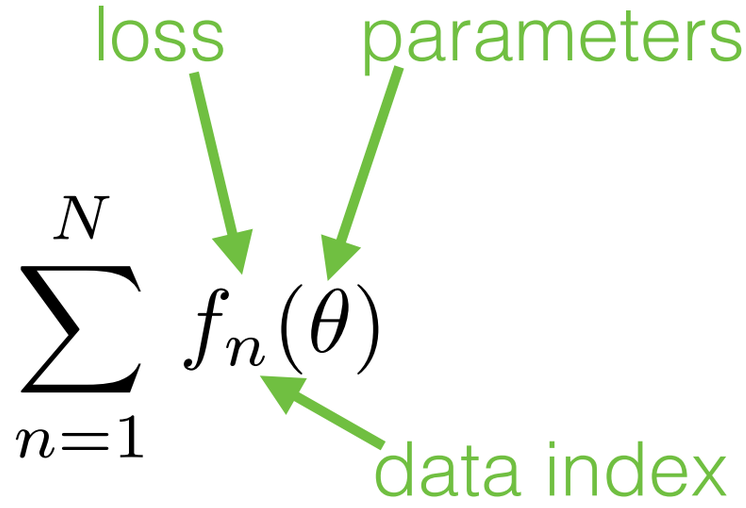
- A data analysis:



Cross validation setup

- A data analysis:

$$\underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{n=1}^{N} f_n(\theta)$$



Cross validation setup

- A data analysis:

$$\underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} f_n(\theta)$$

loss

parameters

data index

Cross validation setup

- A data analysis:

$$\hat{\theta} := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} f_n(\theta)$$

estimator

loss

parameters

data index

Cross validation setup

- A data analysis:

$$\underbrace{\hat{\theta}}_{\text{estimator}} := \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} \overbrace{f_n}^{\text{loss}}(\overbrace{\theta}^{\text{parameters}})$$

data index

  - E.g. max likelihood, min loss, M-estimation

Cross validation setup

- A data analysis:

$$\hat{\theta} := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} f_n(\theta) + \lambda R(\theta)$$

estimator

loss

parameters

penalty

data index

  - E.g. max likelihood, min loss, M-estimation

Cross validation setup

- A data analysis:

$$\hat{\theta} := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$$

estimator

loss

parameters

penalty

data index

  - E.g. max likelihood, min loss, M-estimation

Cross validation setup

- A data analysis:

$$\underbrace{\hat{\theta}}_{\text{estimator}} := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n \underbrace{f_n(\theta)}_{\text{loss}} + \lambda \underbrace{R(\theta)}_{\text{penalty}}$$

loss parameters penalty data index

  - E.g. max likelihood, min loss, M-estimation

Cross validation setup

- A data analysis: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

  estimator

  loss parameters penalty data index

  - E.g. max likelihood, min loss, M-estimation

Cross validation setup

- A data analysis: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

  estimator

  loss parameters penalty data index

  - E.g. max likelihood, min loss, M-estimation

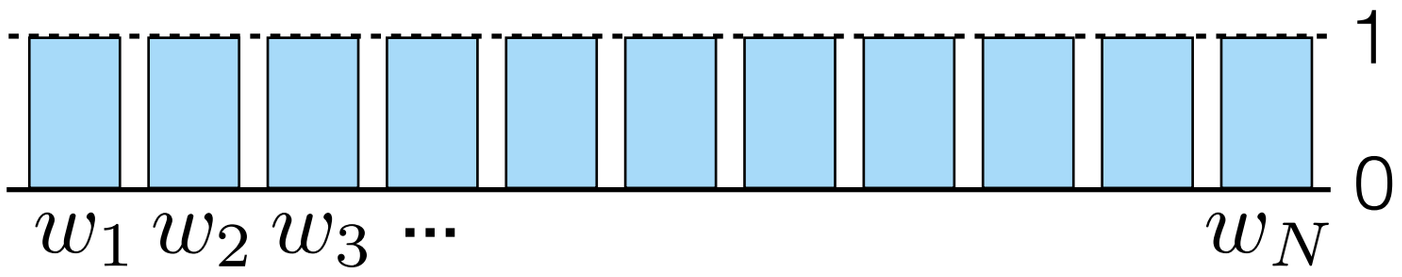
Cross validation setup

- A data analysis: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

  estimator; loss; parameters; penalty; data index

  - E.g. max likelihood, min loss, M-estimation

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



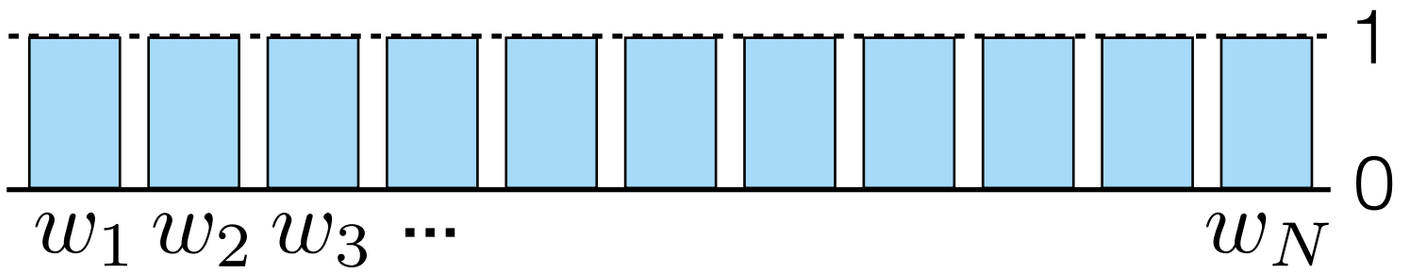
Cross validation setup

- A data analysis: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

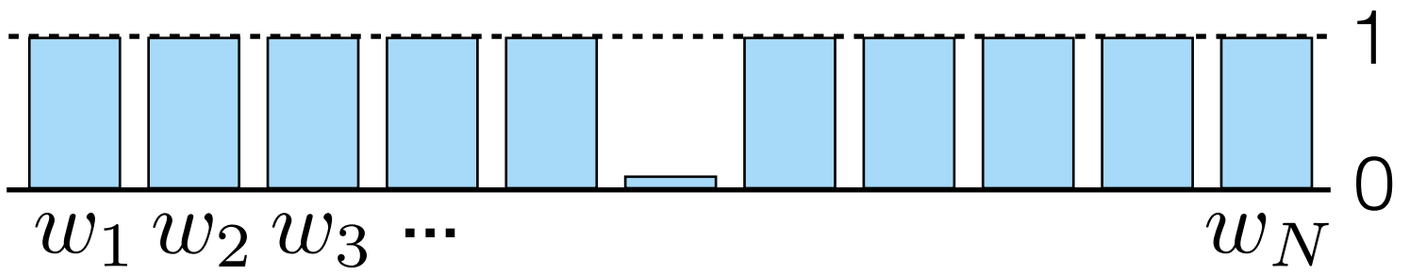
  estimator; loss; parameters; penalty; data index

  - E.g. max likelihood, min loss, M-estimation

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



- Cross validation (CV): $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



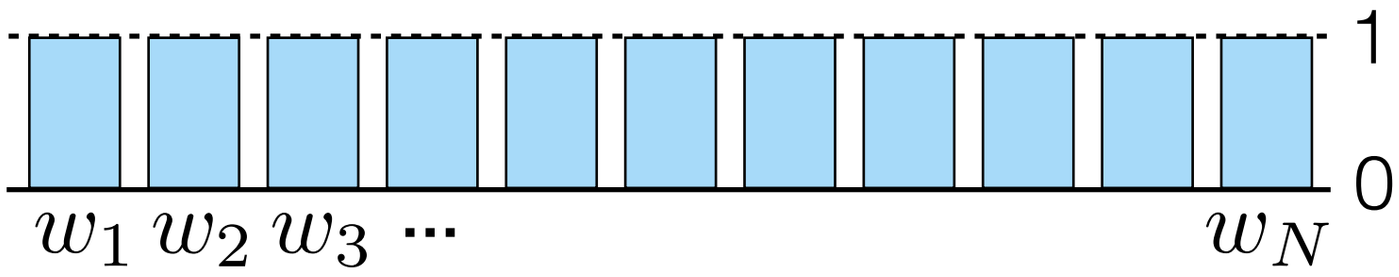
Cross validation setup

- A data analysis: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

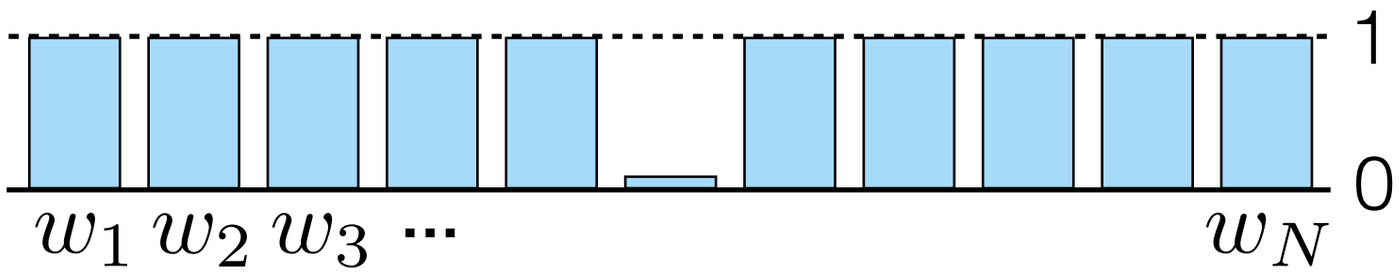
  estimator; loss; parameters; penalty; data index

  - E.g. max likelihood, min loss, M-estimation

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



- Cross validation (CV): $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



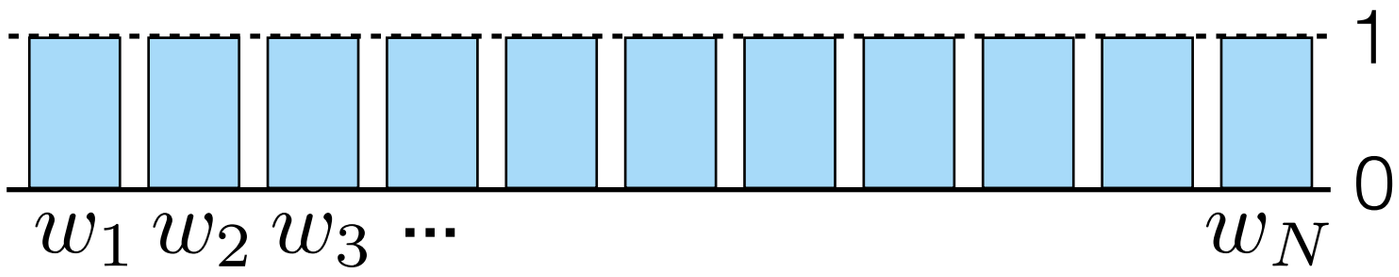
- CV user time cost: easy to use

Cross validation setup

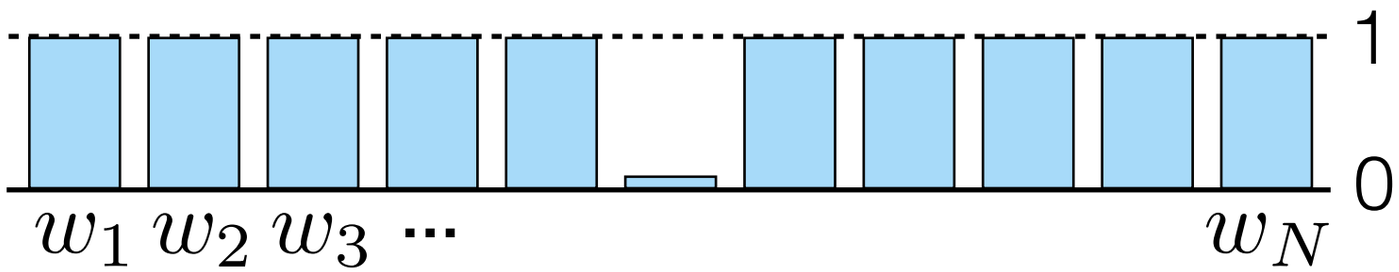
- A data analysis: $\underset{\text{estimator}}{\hat{\theta}(w)} := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

  loss, parameters, penalty, data index

  - E.g. max likelihood, min loss, M-estimation
- Original problem: $w = 1_N = (1, \ldots, 1, 1, 1, \ldots 1)$



- Cross validation (CV): $w = (1, \ldots, 1, 0, 1, \ldots, 1)$



- CV user time cost: easy to use
- CV compute time cost: many x (cost of 1 data analysis)

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

$$H_{d_1 d_2} = \frac{\partial^2}{\partial \theta_{d_1} \partial \theta_{d_2}} \text{objective}$$

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

$$H_{d_1 d_2} = \frac{\partial^2}{\partial \theta_{d_1} \partial \theta_{d_2}} \text{objective} \qquad G_{dn} = \frac{\partial^2}{\partial \theta_d \partial w_n} \text{objective}$$

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

$$H_{d_1 d_2} = \frac{\partial^2}{\partial \theta_{d_1} \partial \theta_{d_2}} \text{objective} \qquad G_{dn} = \frac{\partial^2}{\partial \theta_d \partial w_n} \text{objective}$$

$$G = \frac{1}{N} \begin{pmatrix} | & & | \\ \nabla_\theta f_1 & \cdots & \nabla_\theta f_N \\ | & & | \end{pmatrix}$$

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

- Just one data analysis

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

- Just one data analysis; requires 2x differentiability

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

- Just one data analysis; requires 2x differentiability
- Local sensitivity analysis / influence scores

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Giordano, Broderick, Jordan 2018]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

- Just one data analysis; requires 2x differentiability
- Local sensitivity analysis / influence scores
- Other linear approximations

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Giordano, Broderick, Jordan 2018; Obuchi, Kabashima 2016, 2018; Koh & Liang 2017; Beirami et al 2017; Rad & Maleki 2018; Wang et al 2018]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

- Just one data analysis; requires 2x differentiability
- Local sensitivity analysis / influence scores
- Other linear approximations
- What about higher orders? See our latest on arXiv

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Giordano, Broderick, Jordan 2018; Giordano, Jordan, Broderick 2019; Obuchi, Kabashima 2016, 2018; Koh & Liang 2017; Beirami et al 2017; Rad & Maleki 2018; Wang et al 2018]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

IJ: infinitesimal jackknife

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

- Just one data analysis; requires 2x differentiability
- Local sensitivity analysis / influence scores
- Other linear approximations
- What about higher orders? See our latest on arXiv

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Giordano, Broderick, Jordan 2018; Giordano, Jordan, Broderick 2019; Jaeckel 1972; Clarke 1983; Obuchi, Kabashima 2016, 2018; Koh & Liang 2017; Beirami et al 2017; Rad & Maleki 2018; Wang et al 2018]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

IJ: infinitesimal jackknife

H: Hessian of the objective

G: matrix of gradients

Our contributions:

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left.\frac{d\hat{\theta}(w)}{dw^T}\right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

IJ: infinitesimal jackknife

H: Hessian of the objective

G: matrix of gradients

Our contributions:

- Practical automatic differentiation wrappers

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Baydin et al 2018]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

IJ: infinitesimal jackknife

H: Hessian of the objective

G: matrix of gradients

Our contributions:

- Practical automatic differentiation wrappers
- Practical theory assumptions (vs. e.g. term-wise bounded gradients)

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Baydin et al 2018]

Approximation for large data

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left.\frac{d\hat{\theta}(w)}{dw^T}\right|_{w=1_N} (w - 1_N)$$

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

IJ: infinitesimal jackknife

H: Hessian of the objective

G: matrix of gradients

Our contributions:

- Practical automatic differentiation wrappers
- Practical theory assumptions (vs. e.g. term-wise bounded gradients)
- Finite-sample theoretical bounds on quality

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Baydin et al 2018]

A "Swiss Army infinitesimal jackknife"

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\text{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Idea: (first-order) Taylor expansion around $w = 1_N$

$$\hat{\theta}_{IJ}(w) := \hat{\theta}(1_N) + \left. \frac{d\hat{\theta}(w)}{dw^T} \right|_{w=1_N} (w - 1_N)$$

IJ: infinitesimal jackknife

$$= \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

H: Hessian of the objective

G: matrix of gradients

Our contributions:

- Practical automatic differentiation wrappers
- Practical theory assumptions (vs. e.g. term-wise bounded gradients)
- Finite-sample theoretical bounds on quality

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Baydin et al 2018]

Swiss Army IJ: Easy to use

- In Python:
  - **`autograd`** for autodiff https://github.com/HIPS/autograd
  - **`vittles`** for sensitivity https://github.com/rgiordan/vittles

Swiss Army IJ: Easy to use

- In Python:
  - **`autograd`** for autodiff https://github.com/HIPS/autograd
  - **`vittles`** for sensitivity https://github.com/rgiordan/vittles

```
import autograd.numpy as np
import vittles

def objective_fun(theta, w):
   # Your objective here

# optimize…
theta_opt = optimize(lambda theta: objective_fun(theta,
w_ones))

predictor =
vittles.HyperparameterSensitivityLinearApproximation(
objective_fun, theta_opt, w_ones)

theta_IJ = predictor.predict_opt_par_from_hyper_par(w_new)
```

Theory assumptions (no *term-wise* bounded gradients)

Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

Theory assumptions (no *term-wise* bounded gradients)

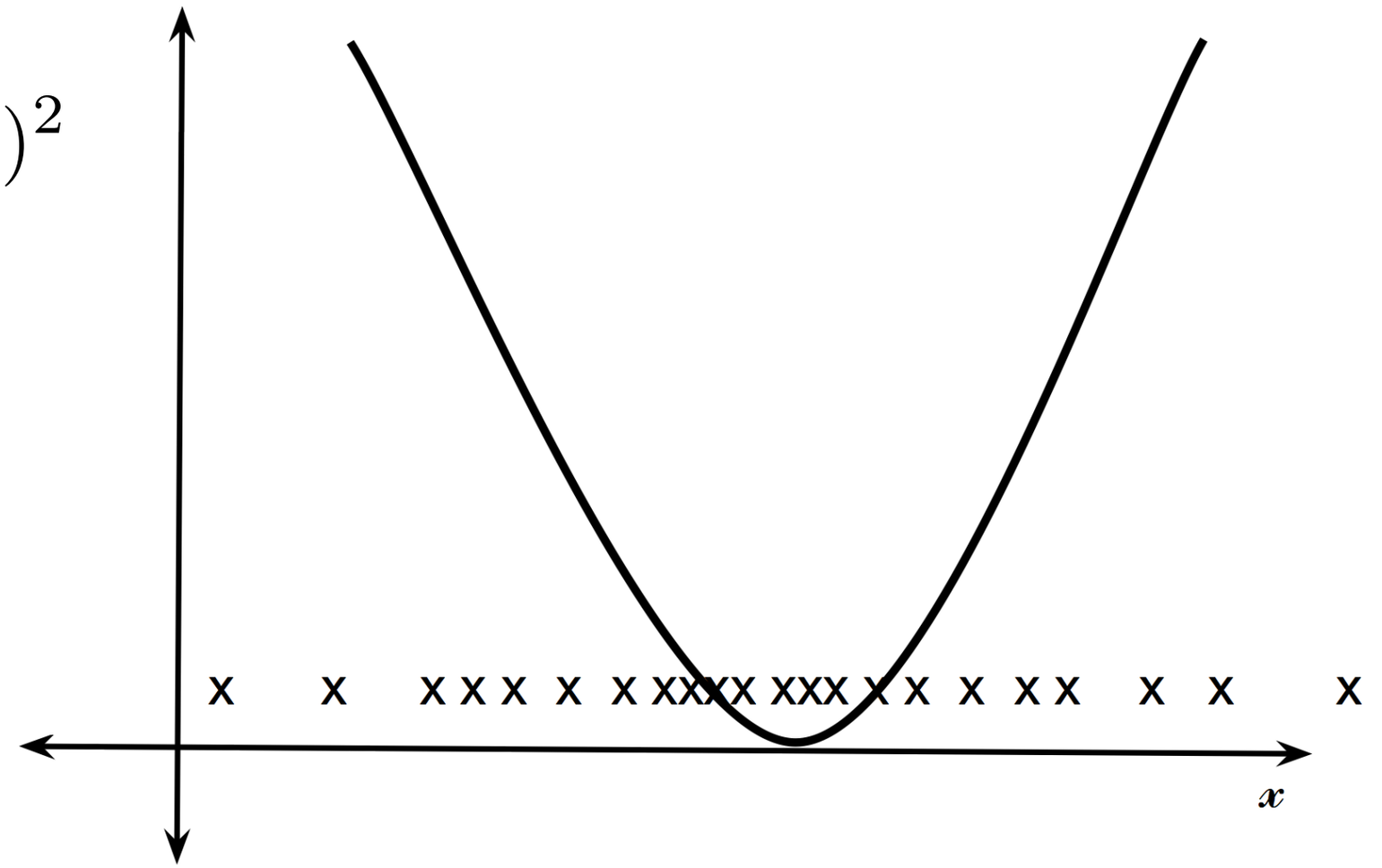
- Exact CV round: $\hat{\theta}(w) := \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

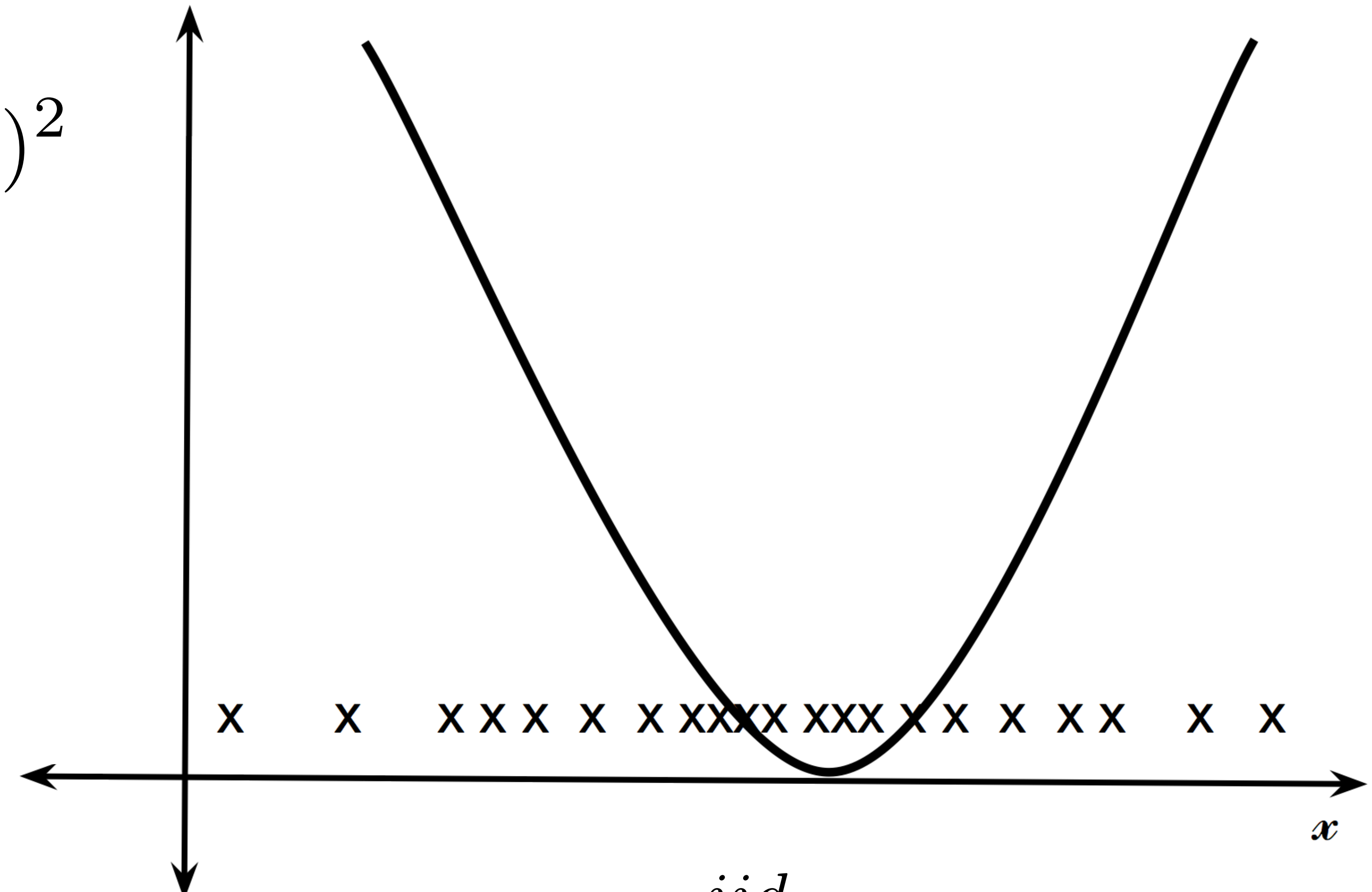
$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$



Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$



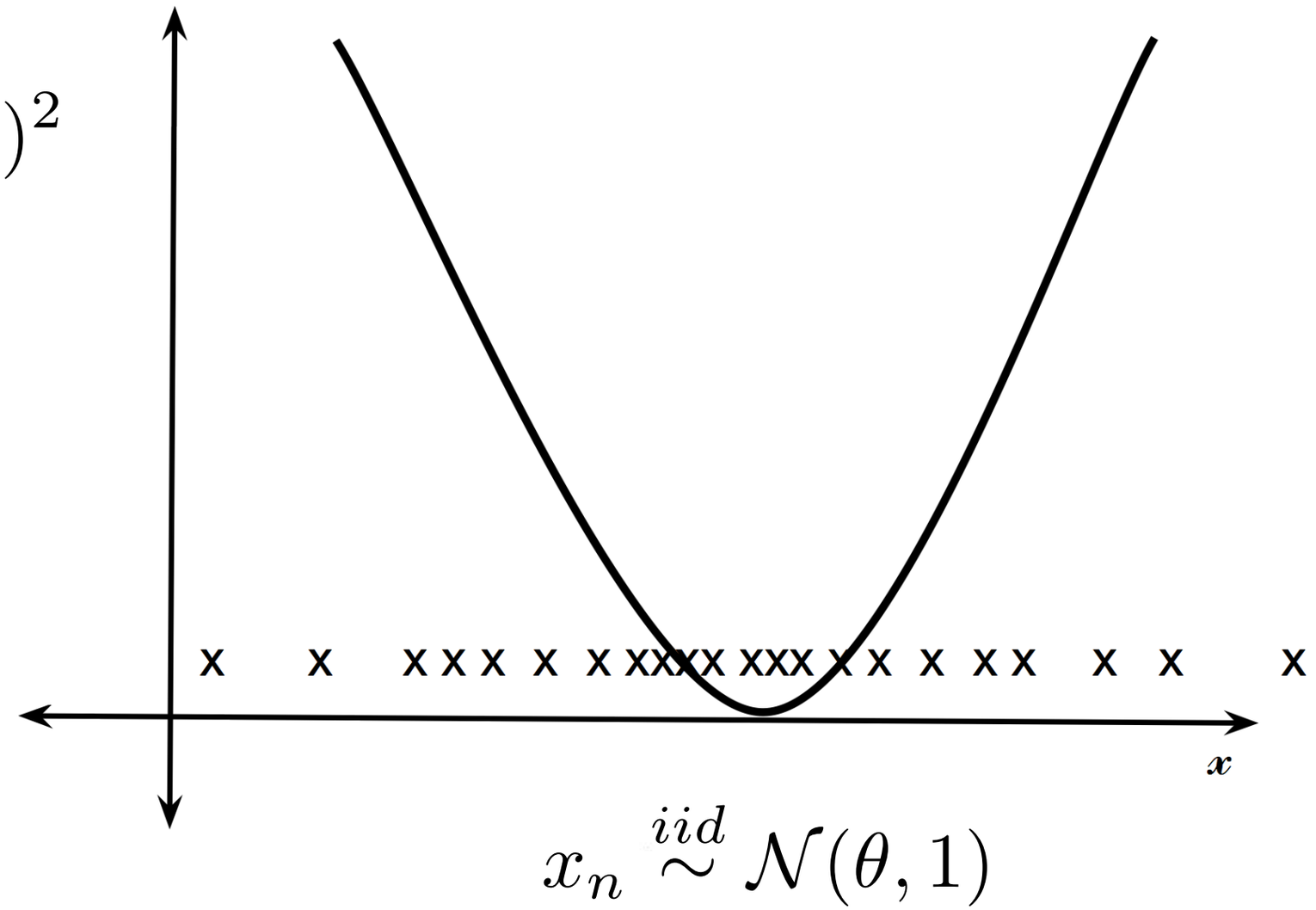
$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

$$\nabla_\theta f_n(\theta) = x_n - \theta$$



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

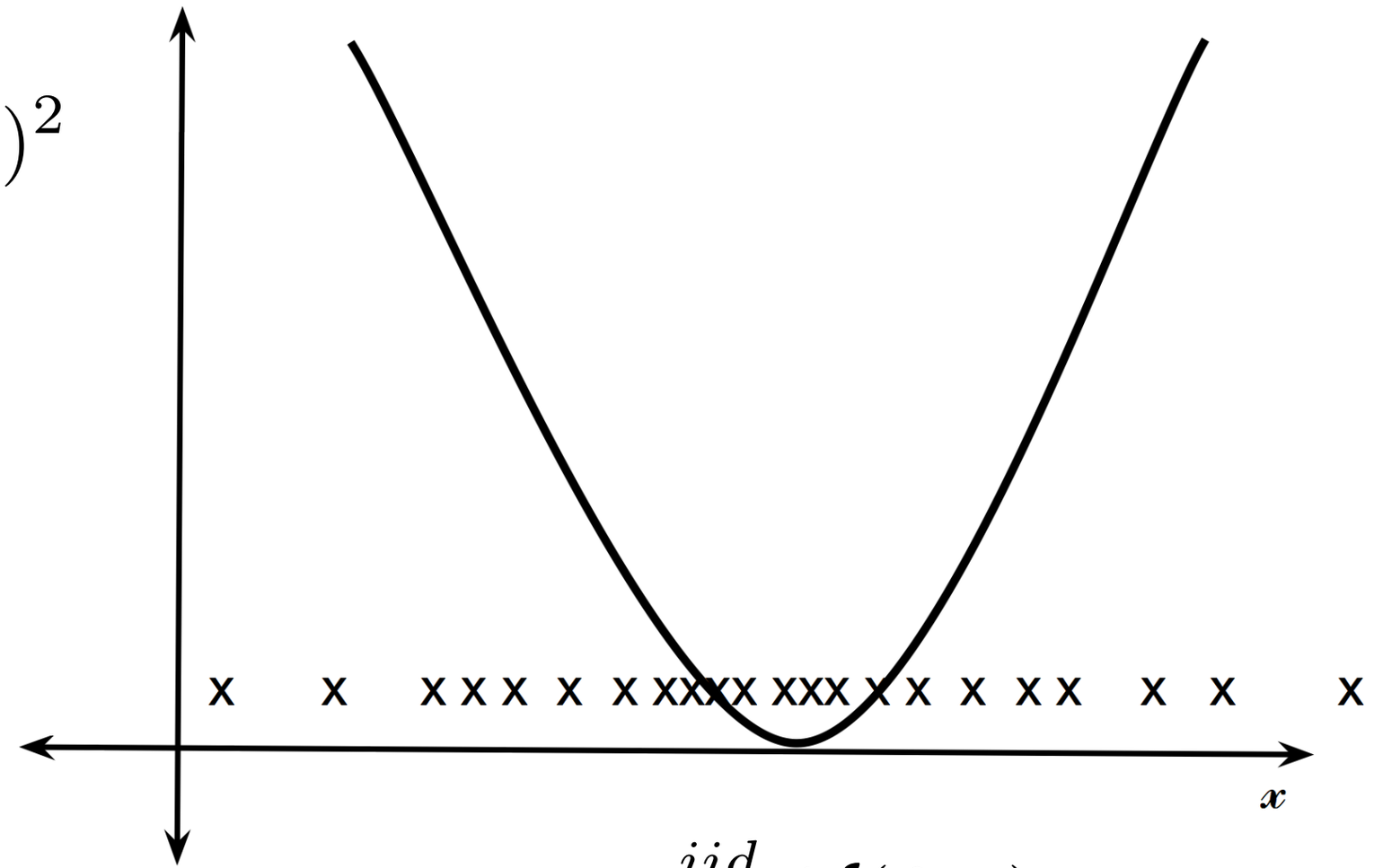
Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

$$\nabla_\theta f_n(\theta) = x_n - \theta$$

$$\max_{n \in 1:N} |x_n - \theta|$$



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

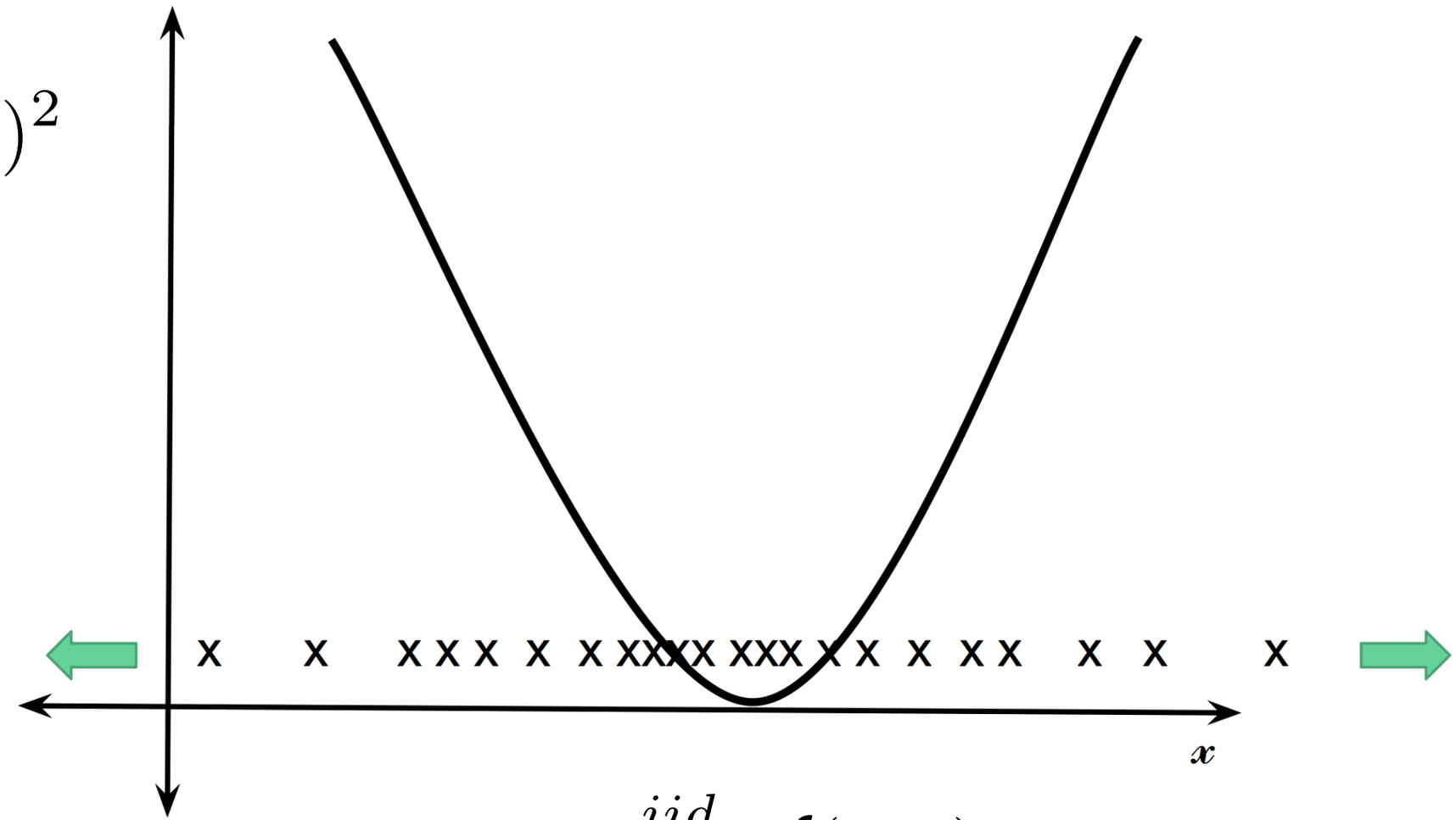
Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

$$\nabla_\theta f_n(\theta) = x_n - \theta$$

$$\max_{n \in 1:N} |x_n - \theta|$$



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

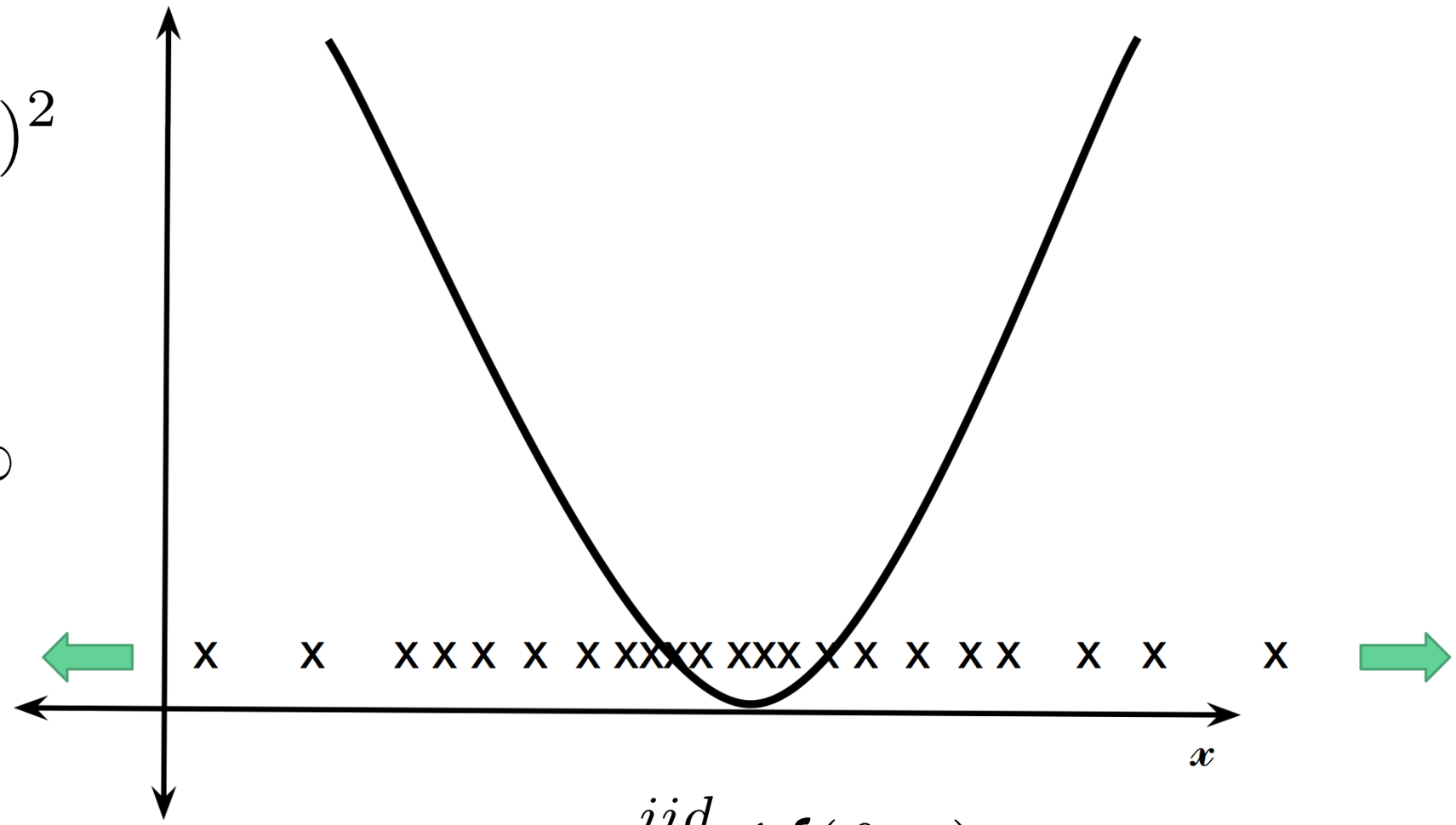
Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

$$\nabla_\theta f_n(\theta) = x_n - \theta$$

$$\max_{n \in 1:N} |x_n - \theta| \to \infty$$



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

Theory assumptions (no *term-wise* bounded gradients)

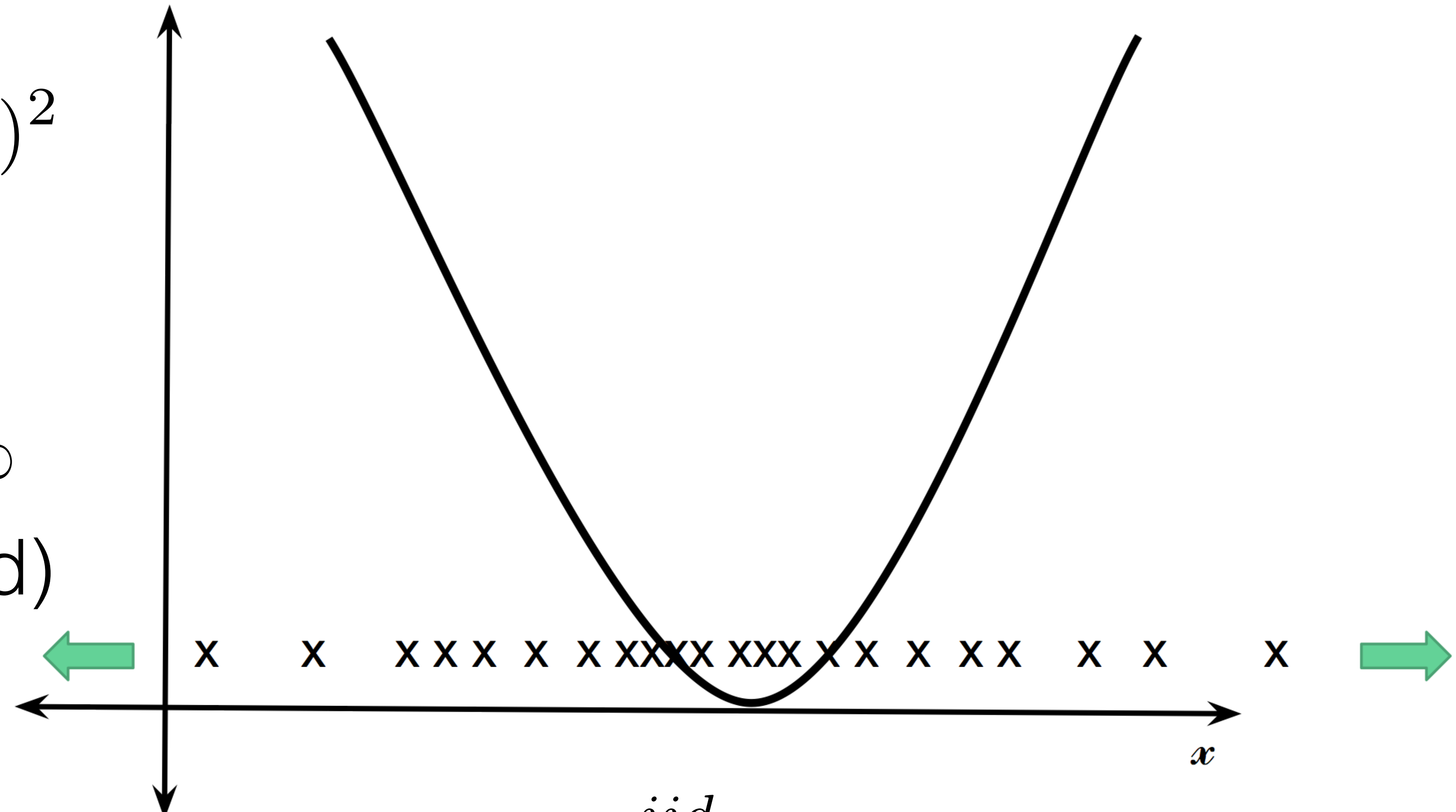
- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

$$\nabla_\theta f_n(\theta) = x_n - \theta$$

$$\max_{n \in 1:N} |x_n - \theta| \to \infty$$

(not bounded)



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\text{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

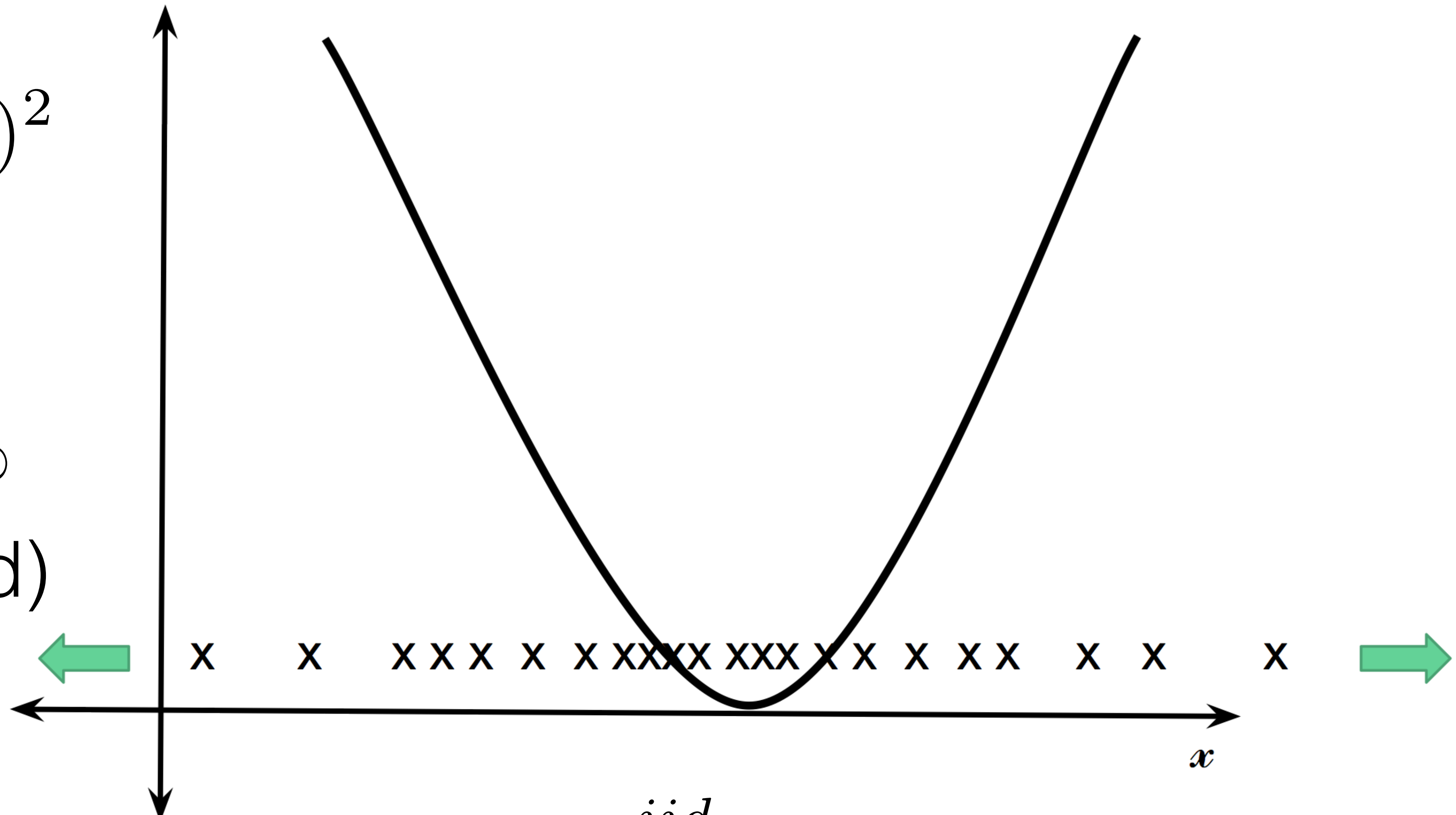
$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

$$\nabla_\theta f_n(\theta) = x_n - \theta$$

$$\max_{n \in 1:N} |x_n - \theta| \to \infty$$

(not bounded)

$$\frac{1}{N} \sum_{n=1}^{N} (x_n - \theta)^2$$



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

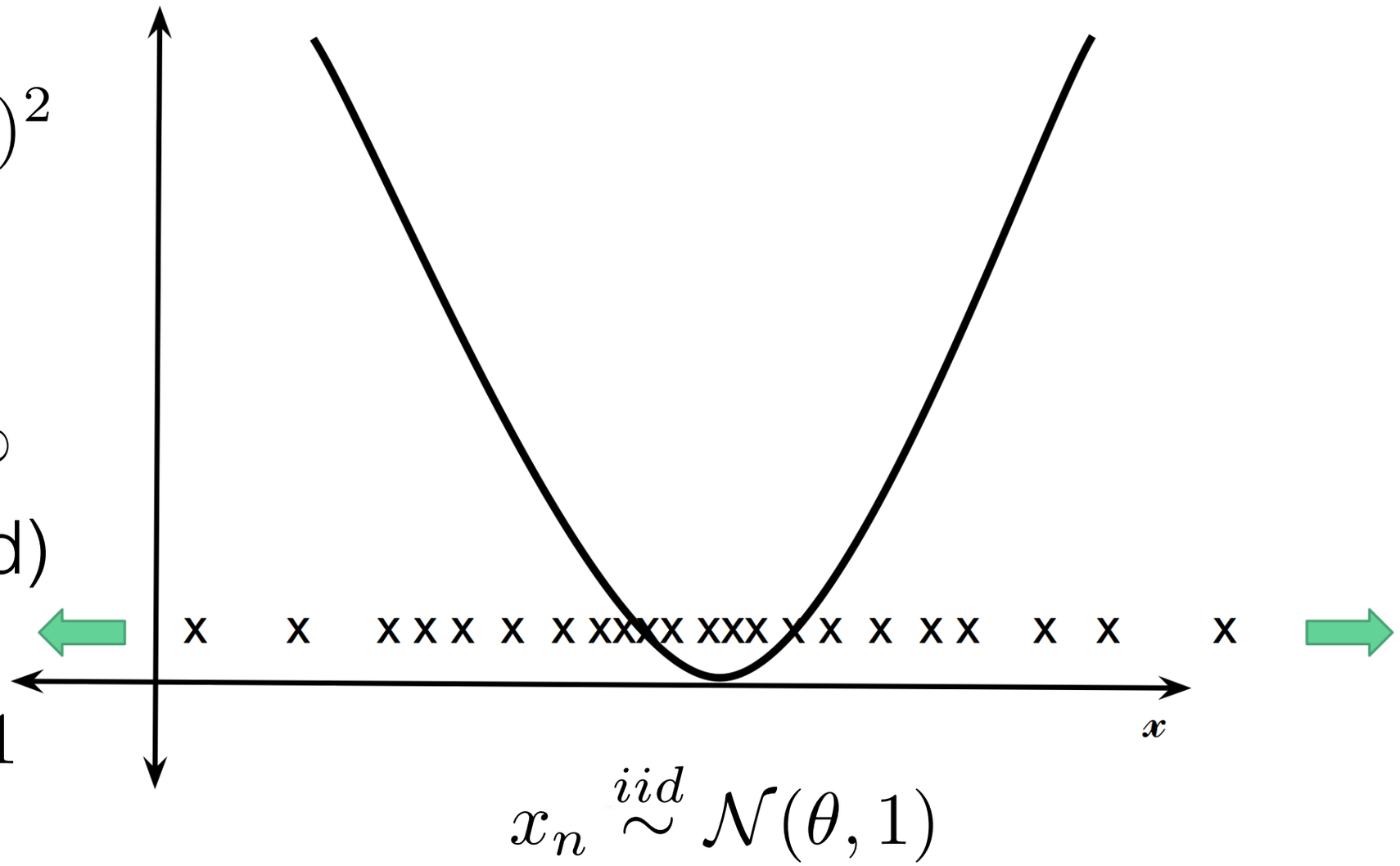
$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

$$\nabla_\theta f_n(\theta) = x_n - \theta$$

$$\max_{n \in 1:N} |x_n - \theta| \to \infty$$

(not bounded)

$$\frac{1}{N} \sum_{n=1}^{N} (x_n - \theta)^2 \to 1$$



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

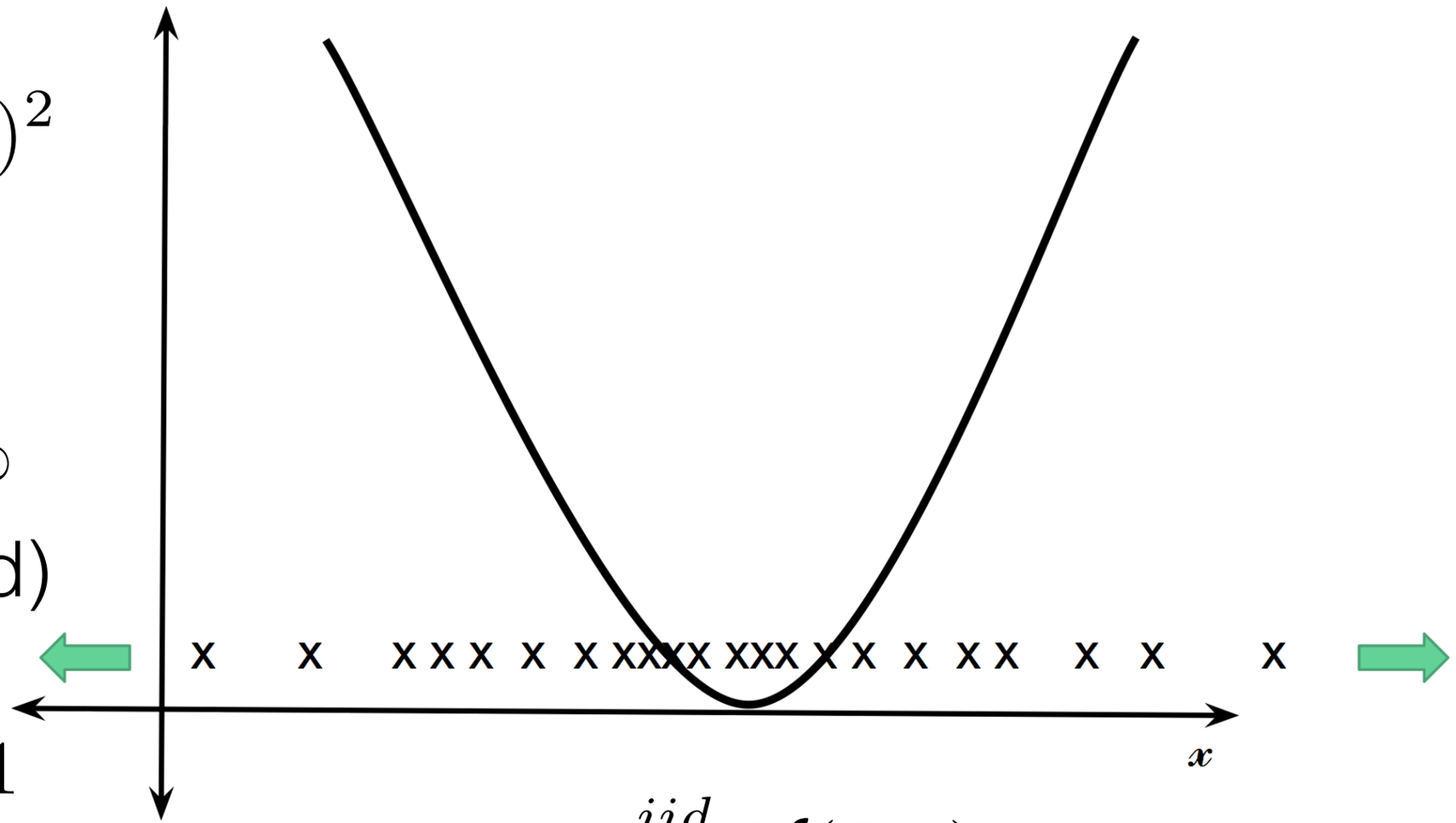
$$\nabla_\theta f_n(\theta) = x_n - \theta$$

$$\max_{n \in 1:N} |x_n - \theta| \to \infty$$

(not bounded)

$$\frac{1}{N} \sum_{n=1}^{N} (x_n - \theta)^2 \to 1$$

(is bounded)



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

Theory assumptions (no *term-wise* bounded gradients)

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$
- Consider:

$$f_n(\theta) = \frac{1}{2}(x_n - \theta)^2$$

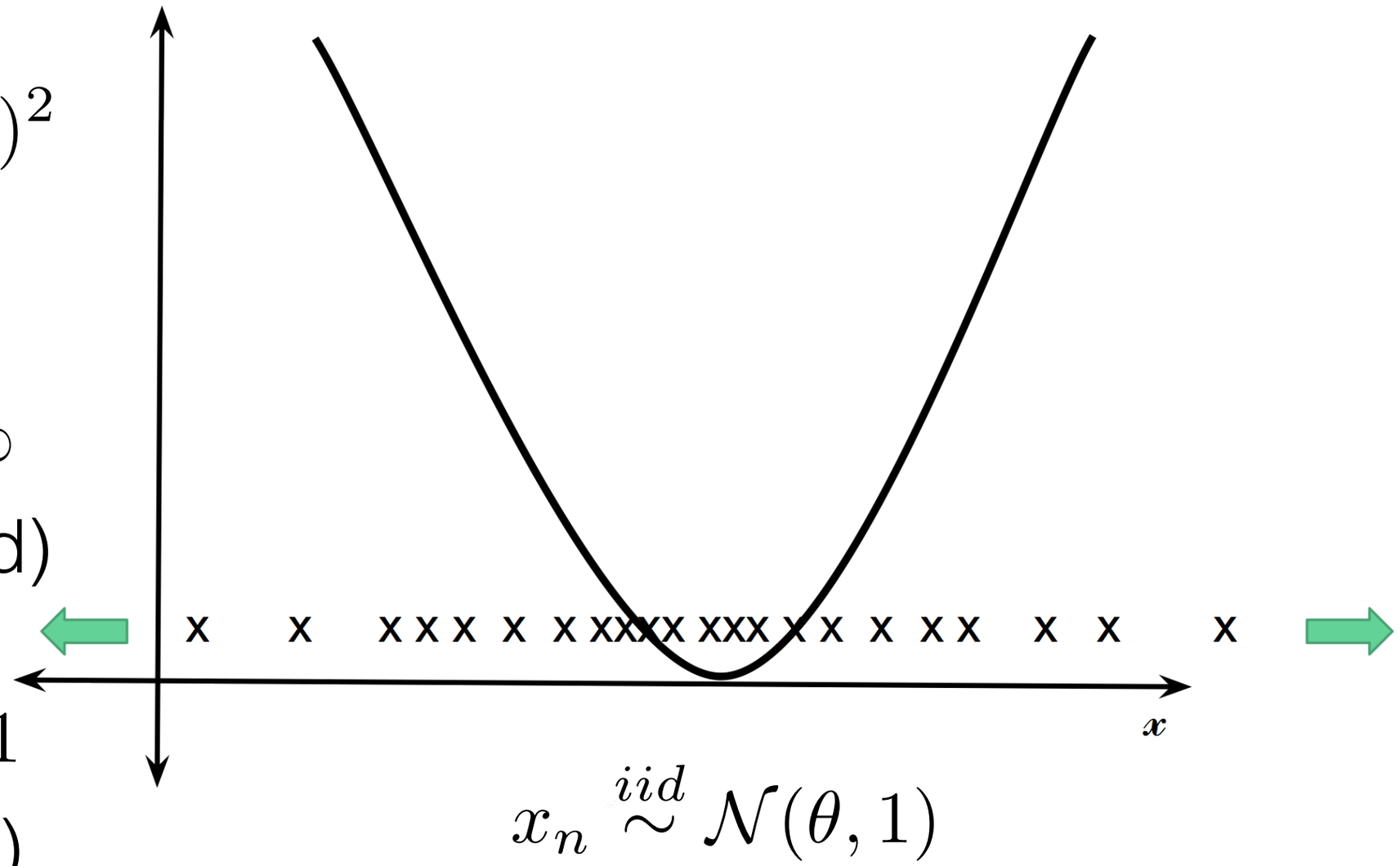
$$\nabla_\theta f_n(\theta) = x_n - \theta$$

$$\max_{n \in 1:N} |x_n - \theta| \to \infty$$

(not bounded)

$$\frac{1}{N} \sum_{n=1}^{N} (x_n - \theta)^2 \to 1$$

(is bounded)



$$x_n \overset{iid}{\sim} \mathcal{N}(\theta, 1)$$

- Our assumption: bounded *sample variances* of gradients and Hessians

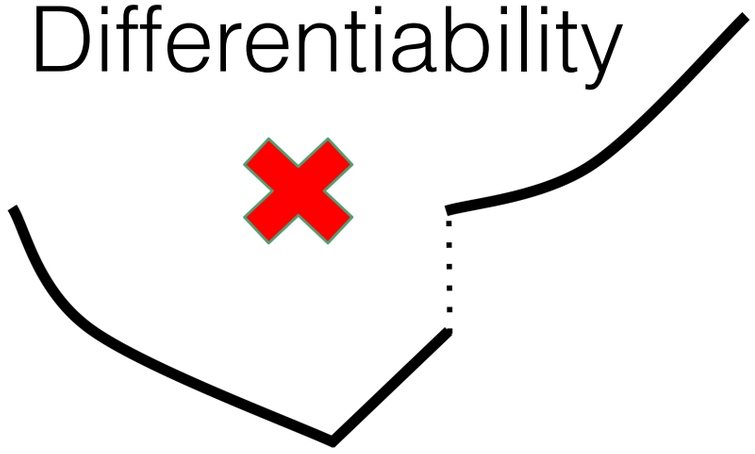
More theory assumptions

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

More theory assumptions

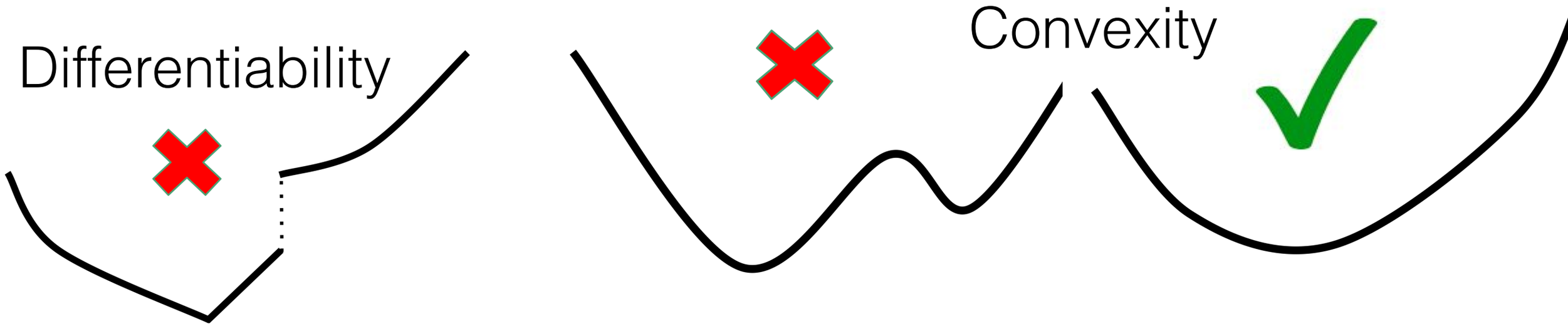
- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

Differentiability



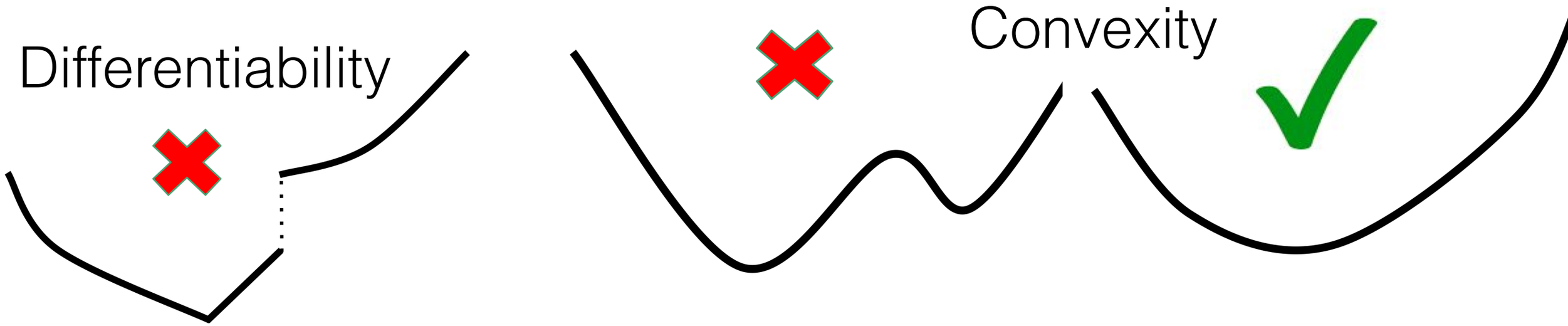
More theory assumptions

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$



More theory assumptions

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

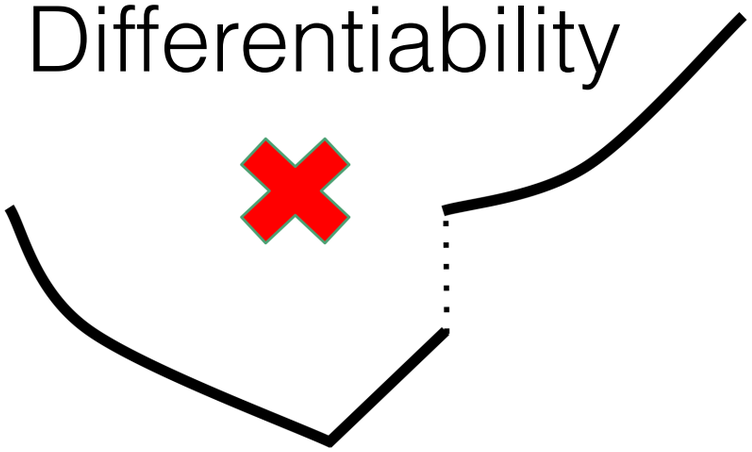


Weights not too extreme

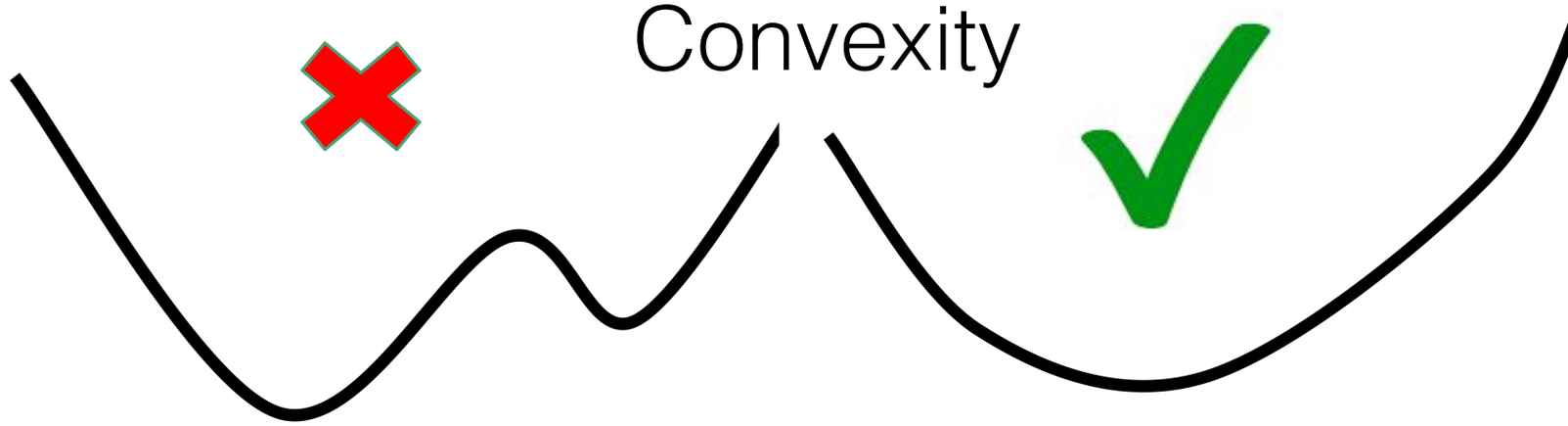
More theory assumptions

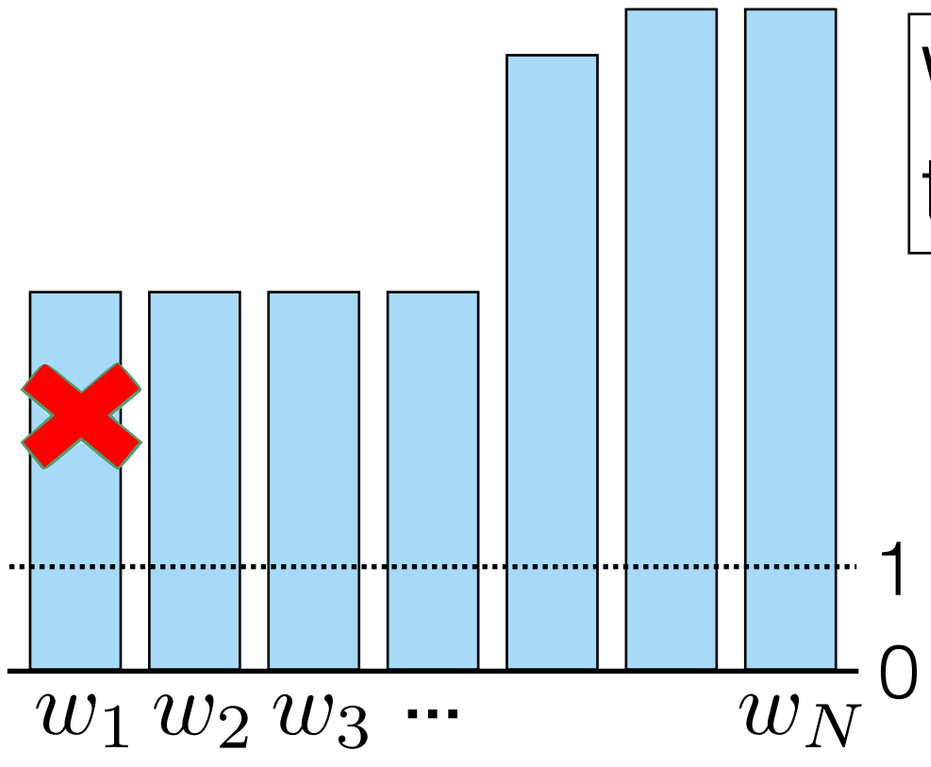
- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

Differentiability



Convexity



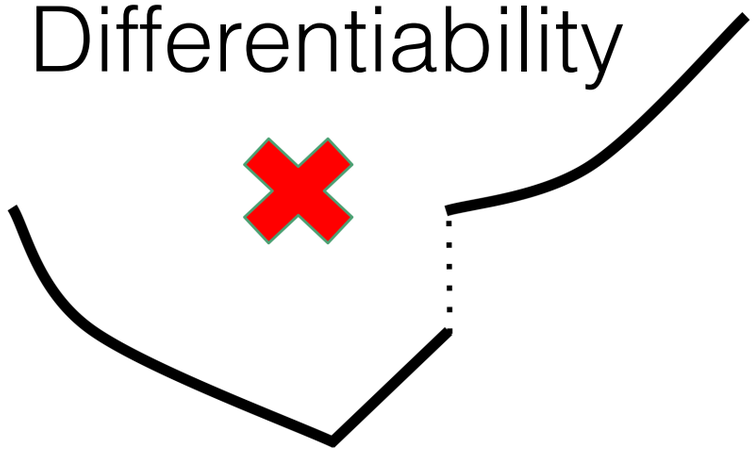


Weights not too extreme

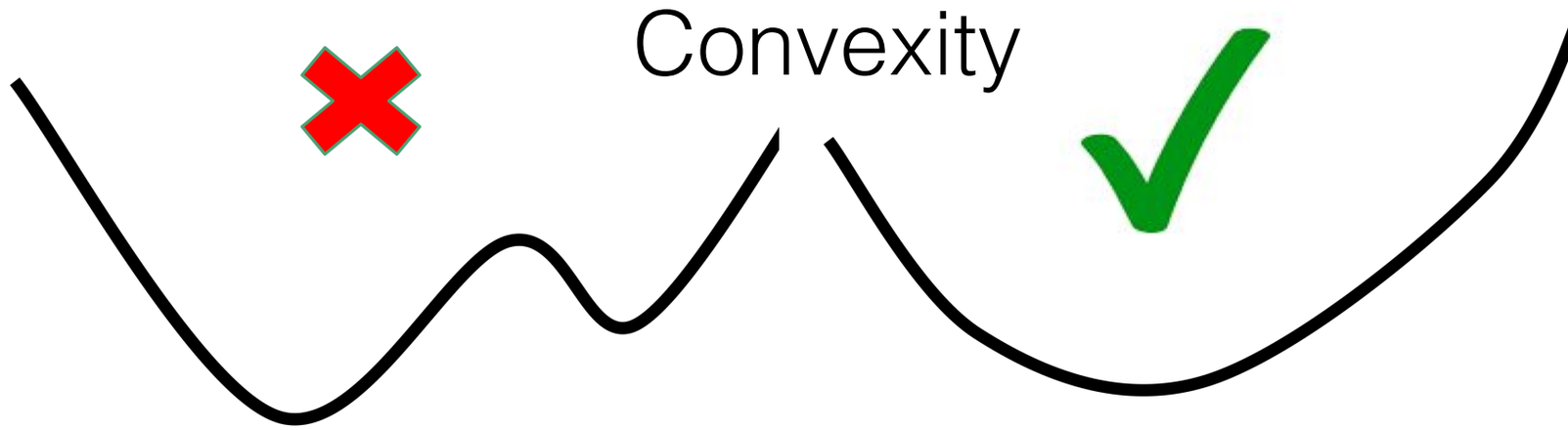
More theory assumptions

- Exact CV round: $\hat{\theta}(w) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n f_n(\theta) + \lambda R(\theta)$

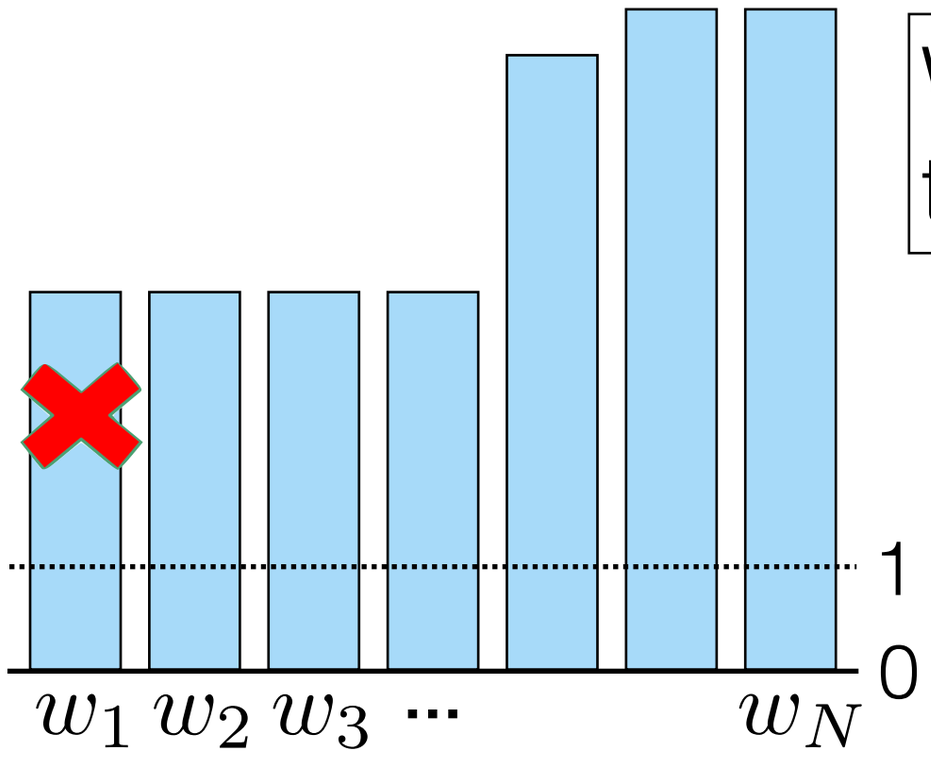
Differentiability

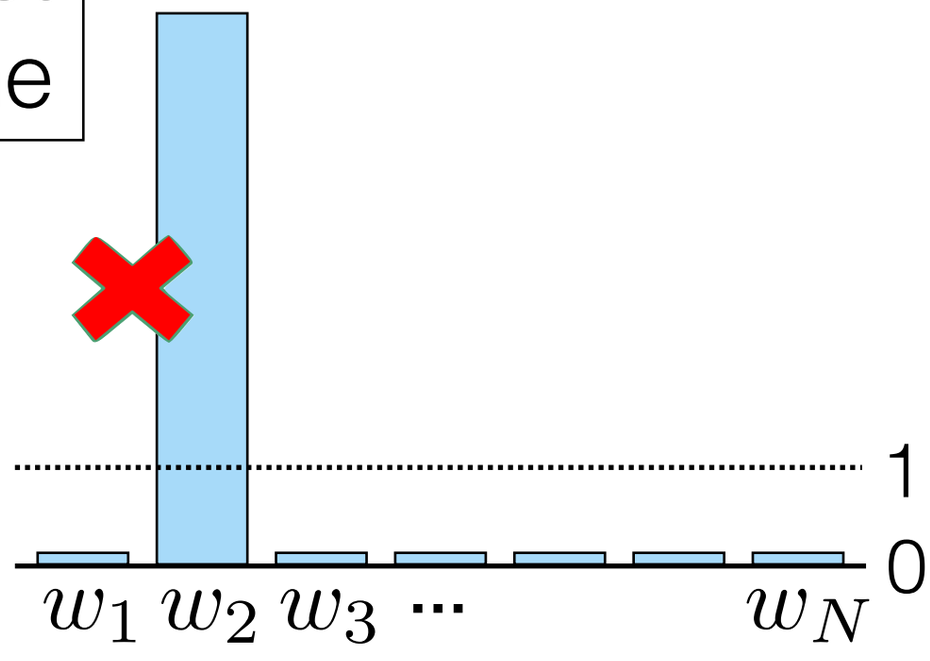


Convexity



Weights not too extreme





Theory condition

Theory condition

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$

Theory condition

- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$

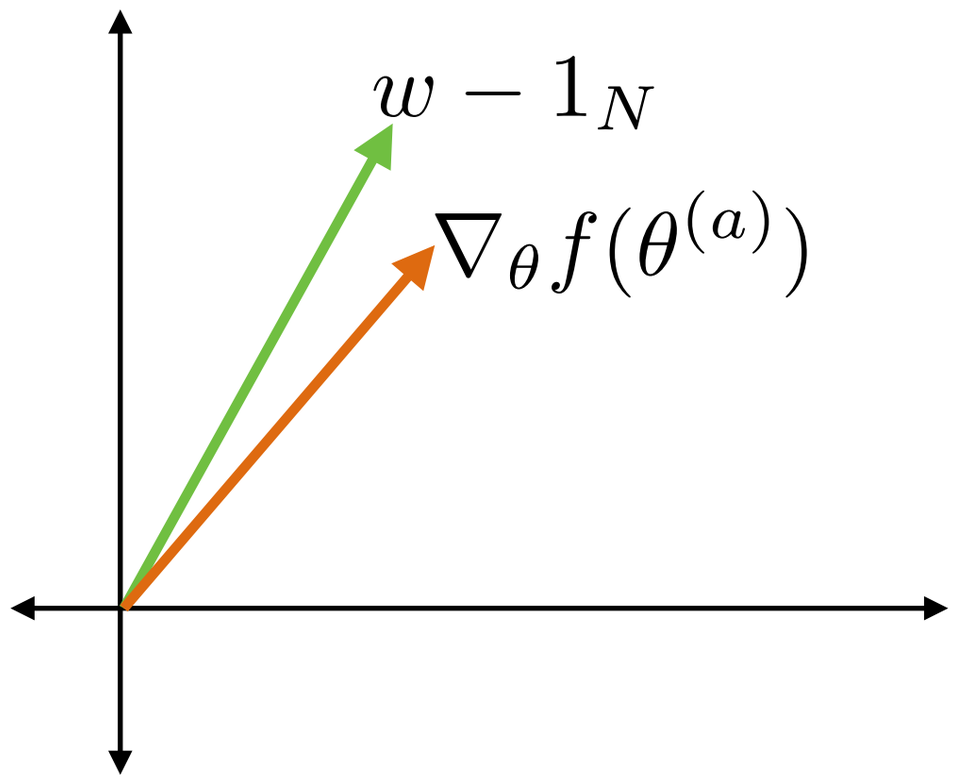
Theory condition

- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$



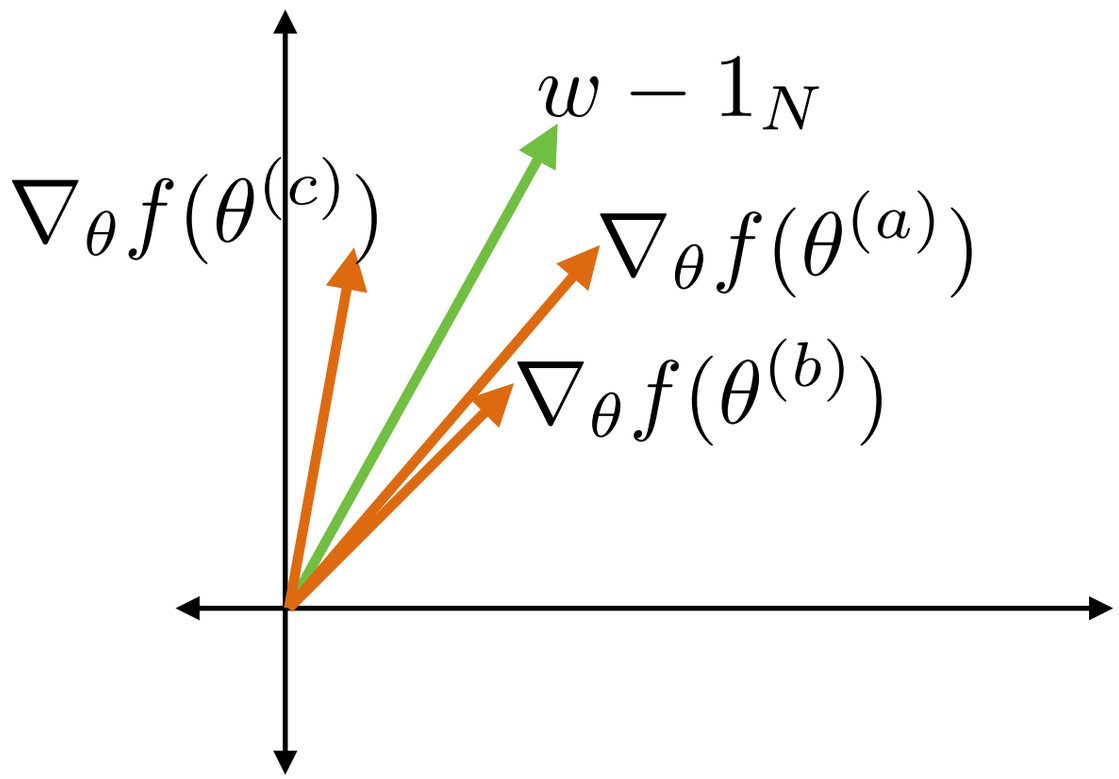
Theory condition

- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$



Theory condition

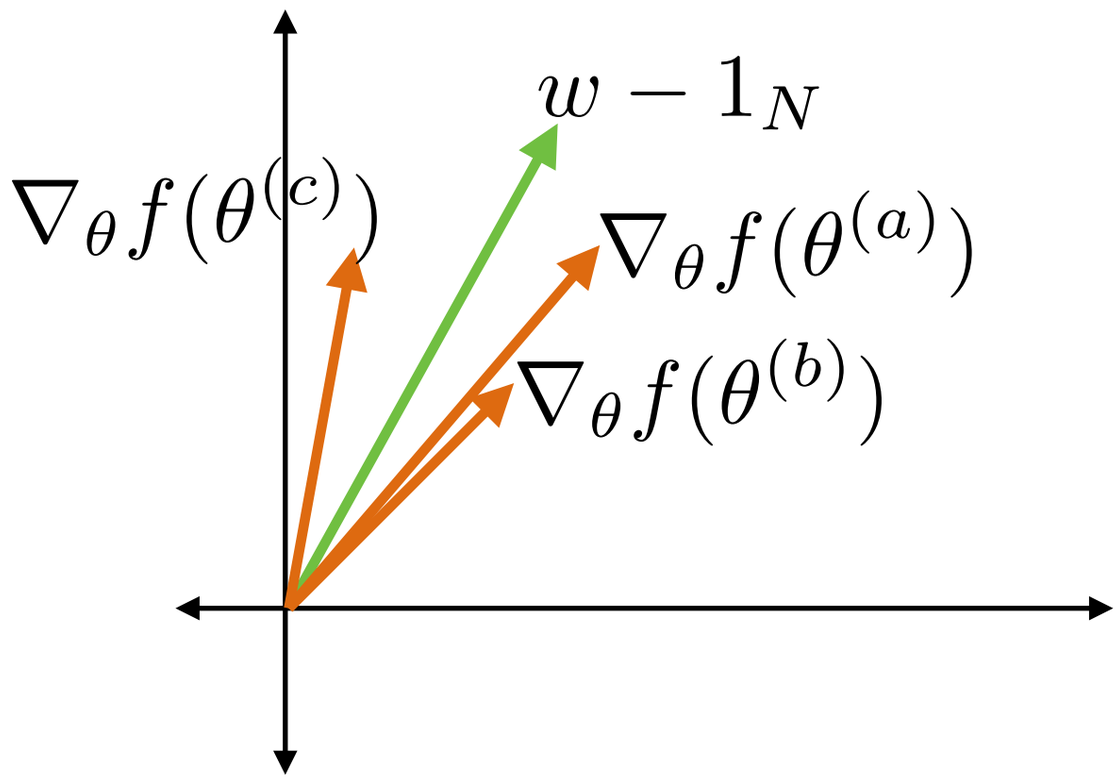
- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$

- + Analogous assumption on the Hessian terms



Theory condition

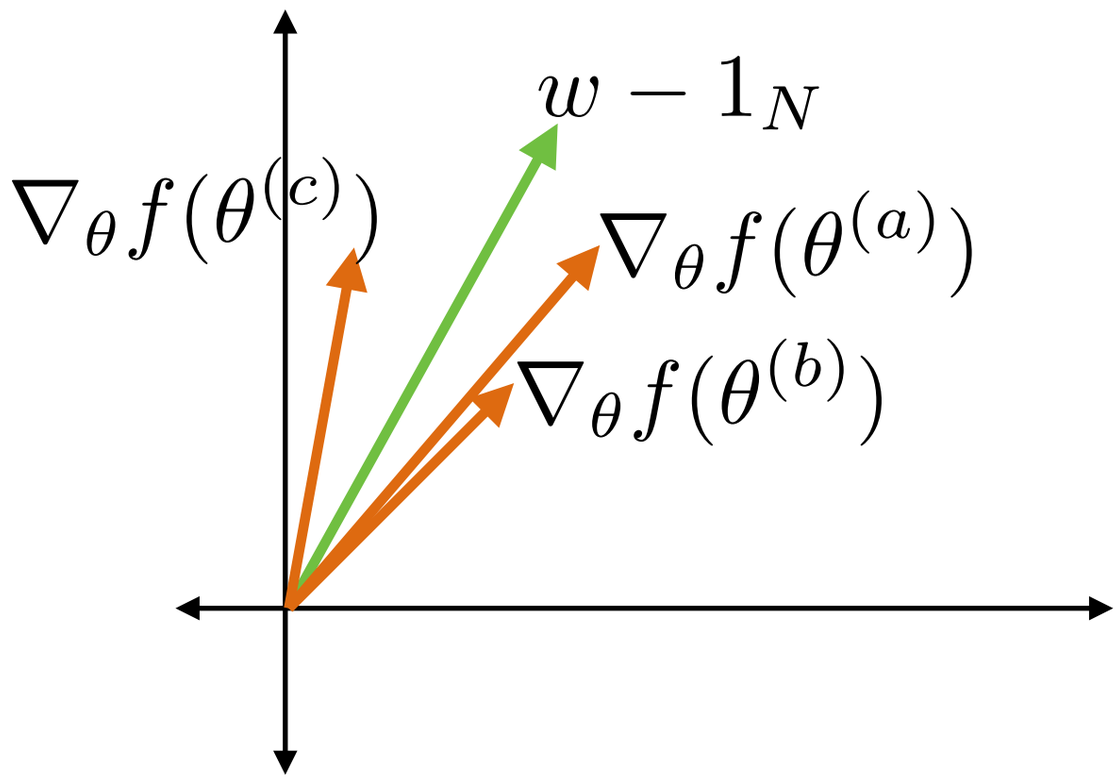
- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$

- + Analogous assumption on the Hessian terms
- Always a non-empty weight set for any delta



Theory condition

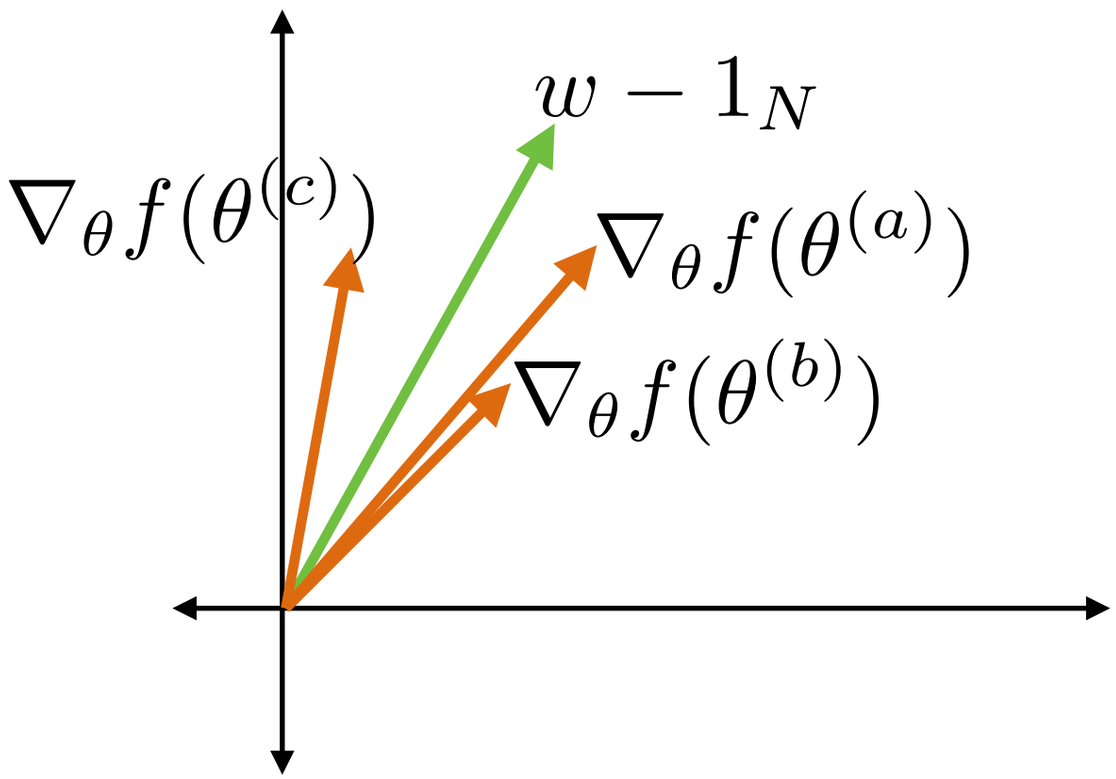
- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$

- + Analogous assumption on the Hessian terms
- Always a non-empty weight set for any delta
- CV works via Holder's inequality



Theory condition

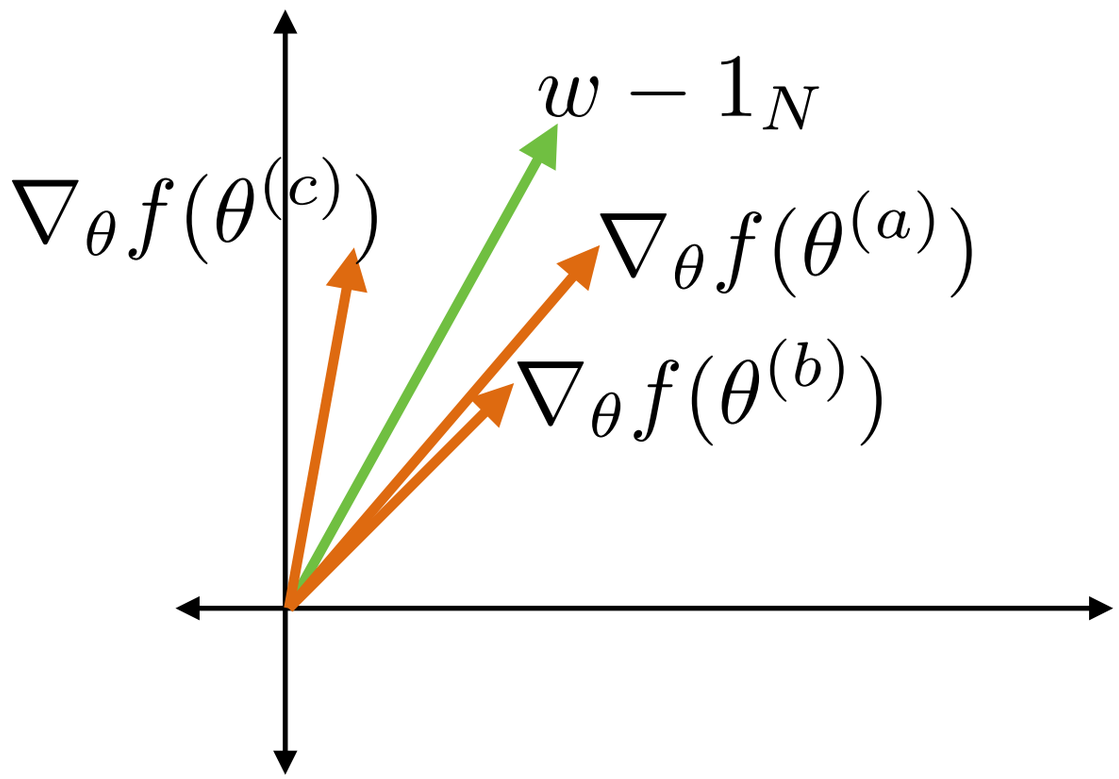
- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$

- + Analogous assumption on the Hessian terms
- Always a non-empty weight set for any delta
- CV works via Holder's inequality



Theory condition

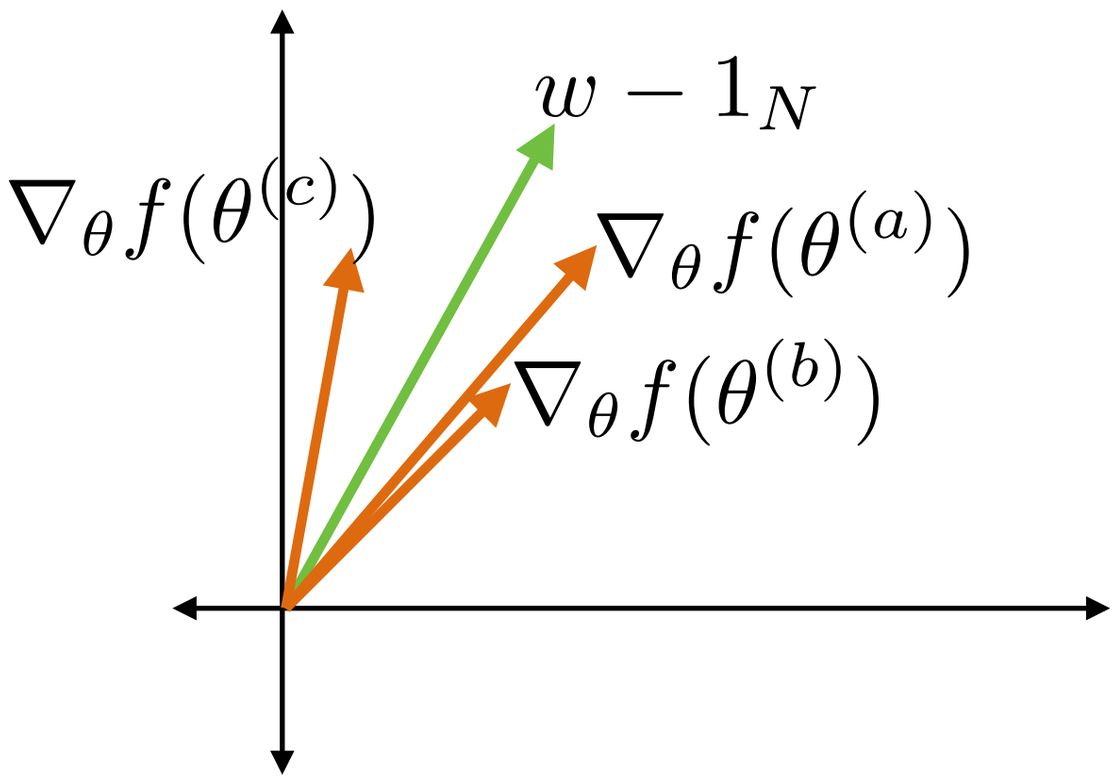
- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- A set complexity condition:

$$\max_{w \in W_\delta} \sup_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f_n(\theta) \right\|_1 \leq \delta$$

- + Analogous assumption on the Hessian terms
- Always a non-empty weight set for any delta
- CV works via Holder's inequality



Theory

Theorem: Under our assumptions,

$$\delta \leq \Delta \Rightarrow \max_{w \in W_\delta} \left\| \hat{\theta}_{IJ}(w) - \hat{\theta}(w) \right\|_2 \leq C\delta^2$$

Explicit functions of constants from our assumptions

Theory

Theorem: Under our assumptions,

$$\delta \leq \Delta \Rightarrow \max_{w \in W_\delta} \left\| \hat{\theta}_{IJ}(w) - \hat{\theta}(w) \right\|_2 \leq C\delta^2$$

Explicit functions of constants from our assumptions

Corollary: For leave-*k*-out CV,

$$\max_{w \in W^{(k)}} \|\hat{\theta}_{IJ}(w) - \hat{\theta}(w)\|_2 \leq \frac{C_1 \|\nabla_\theta f\|_\infty^2}{N^2} \leq C_2 N^{-1}$$

Swiss Army IJ: Experiments

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

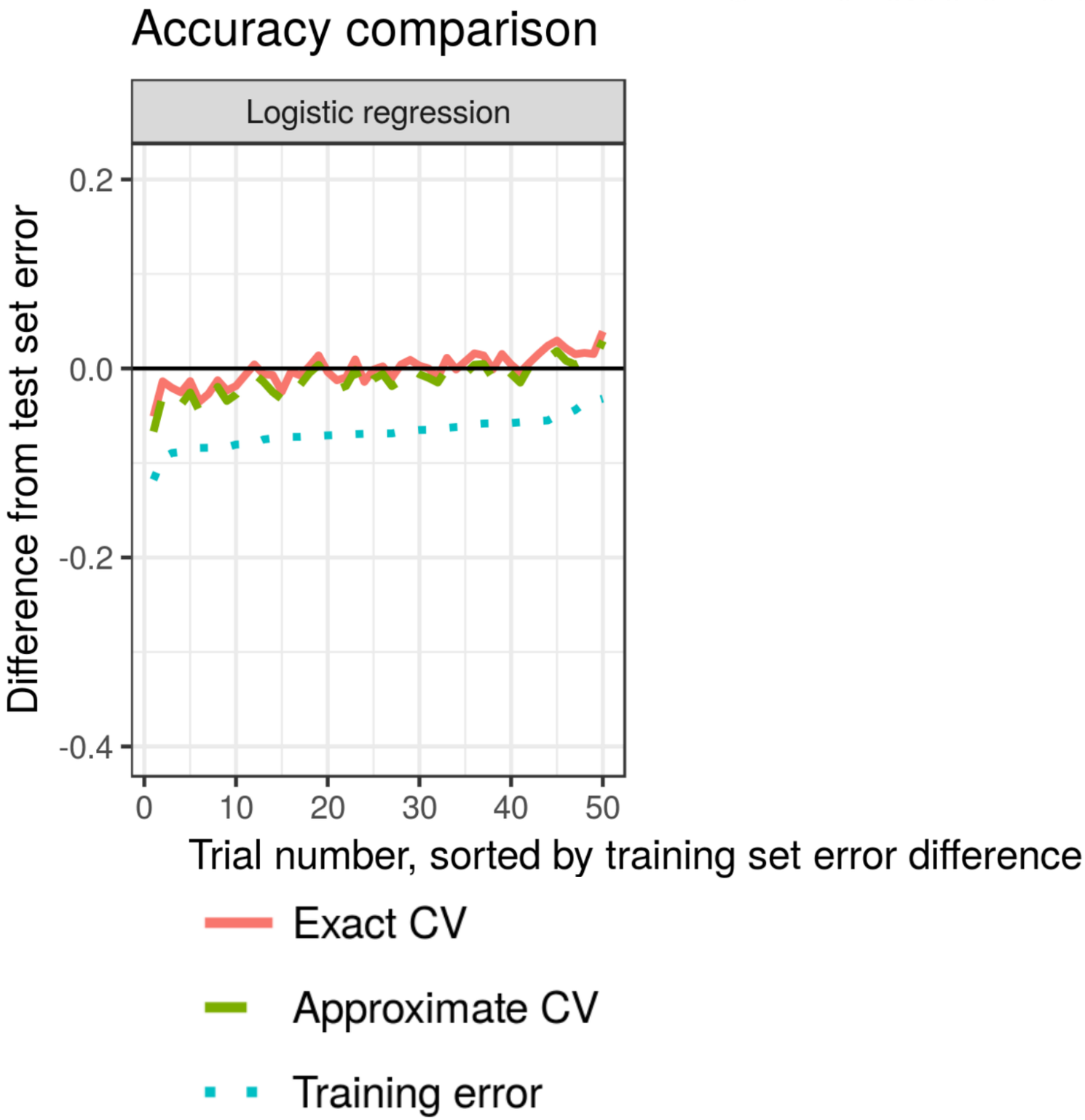
Swiss Army IJ: Experiments

- Simulated data

[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Swiss Army IJ: Experiments

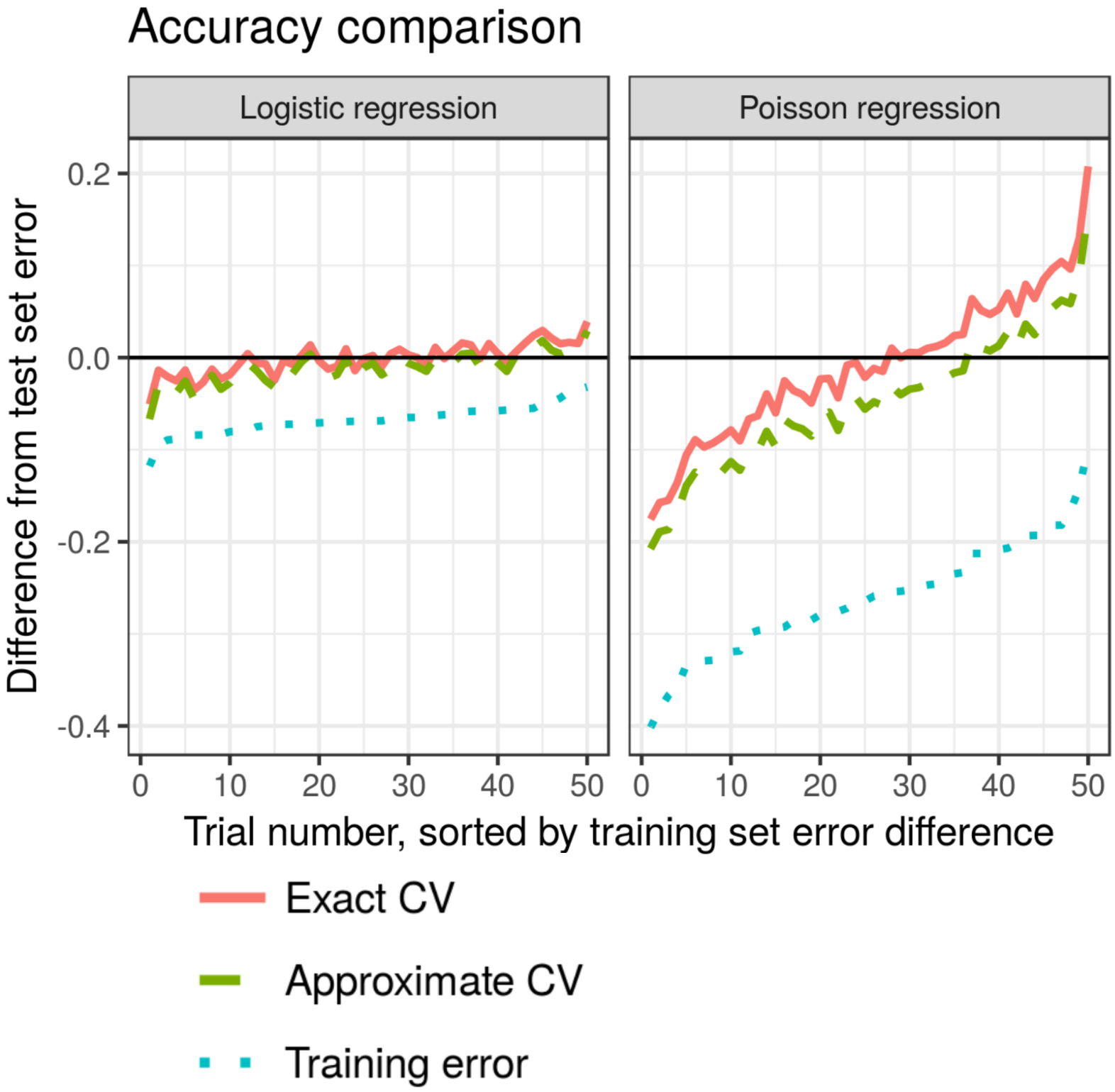
- Simulated data



[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Swiss Army IJ: Experiments

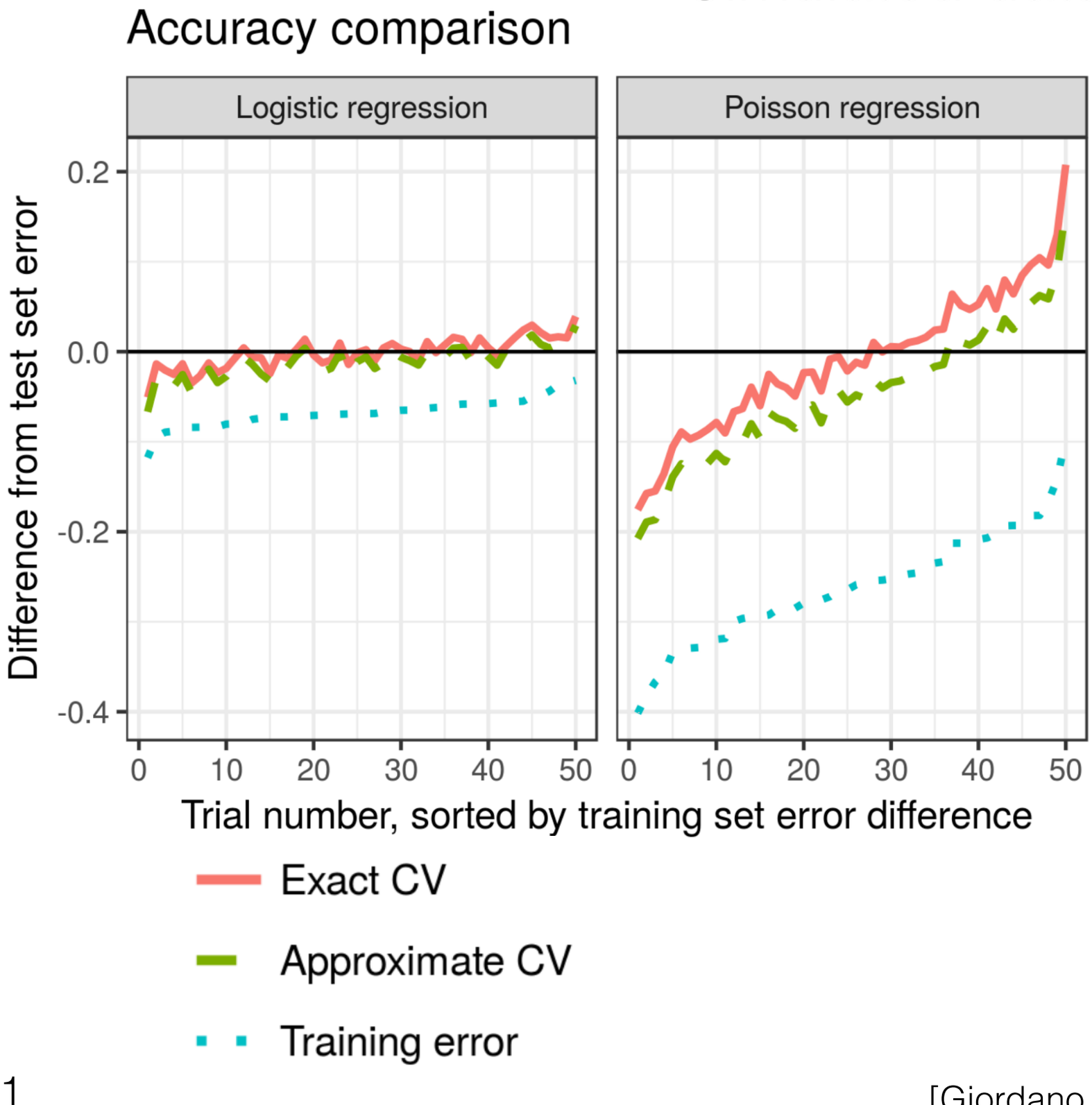
- Simulated data

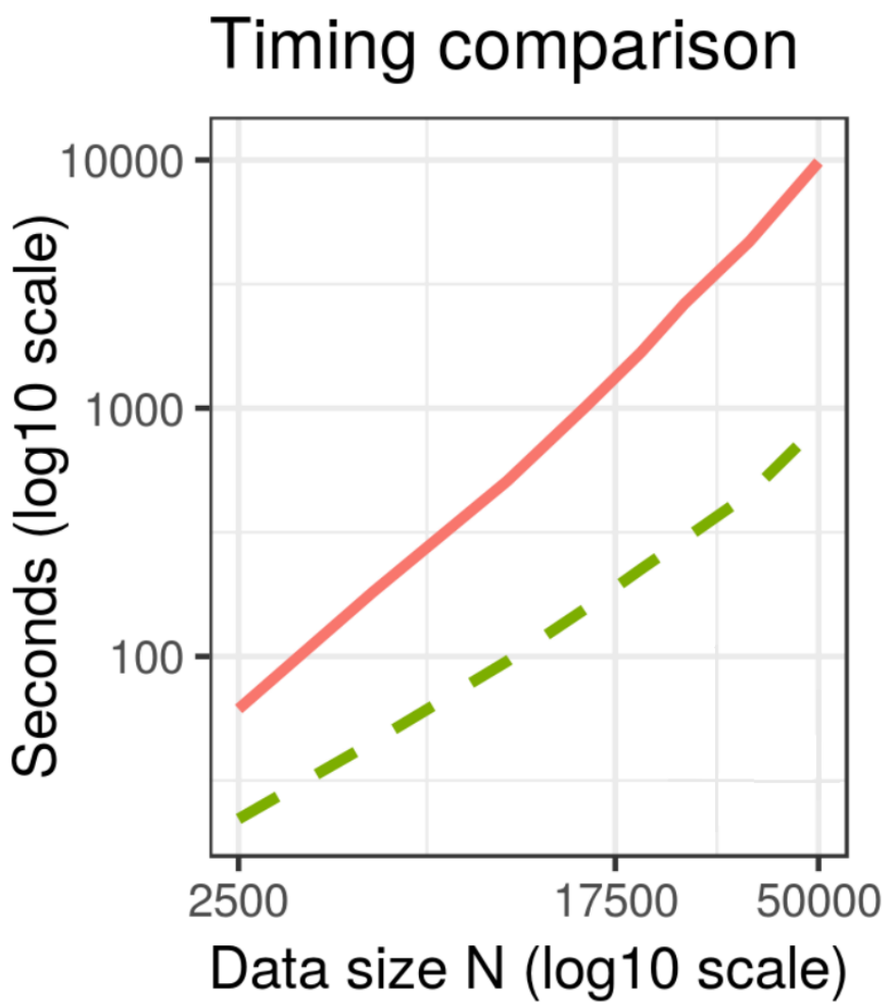


[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Swiss Army IJ: Experiments

- Simulated data





[Giordano, Stephenson, Liu, Jordan, Broderick 2019]

Swiss Army IJ: Experiments

- Mice gene expression (*N*=1000) time series, after influenza infection

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Luan, Li 2003; Shoemaker et al 2015]

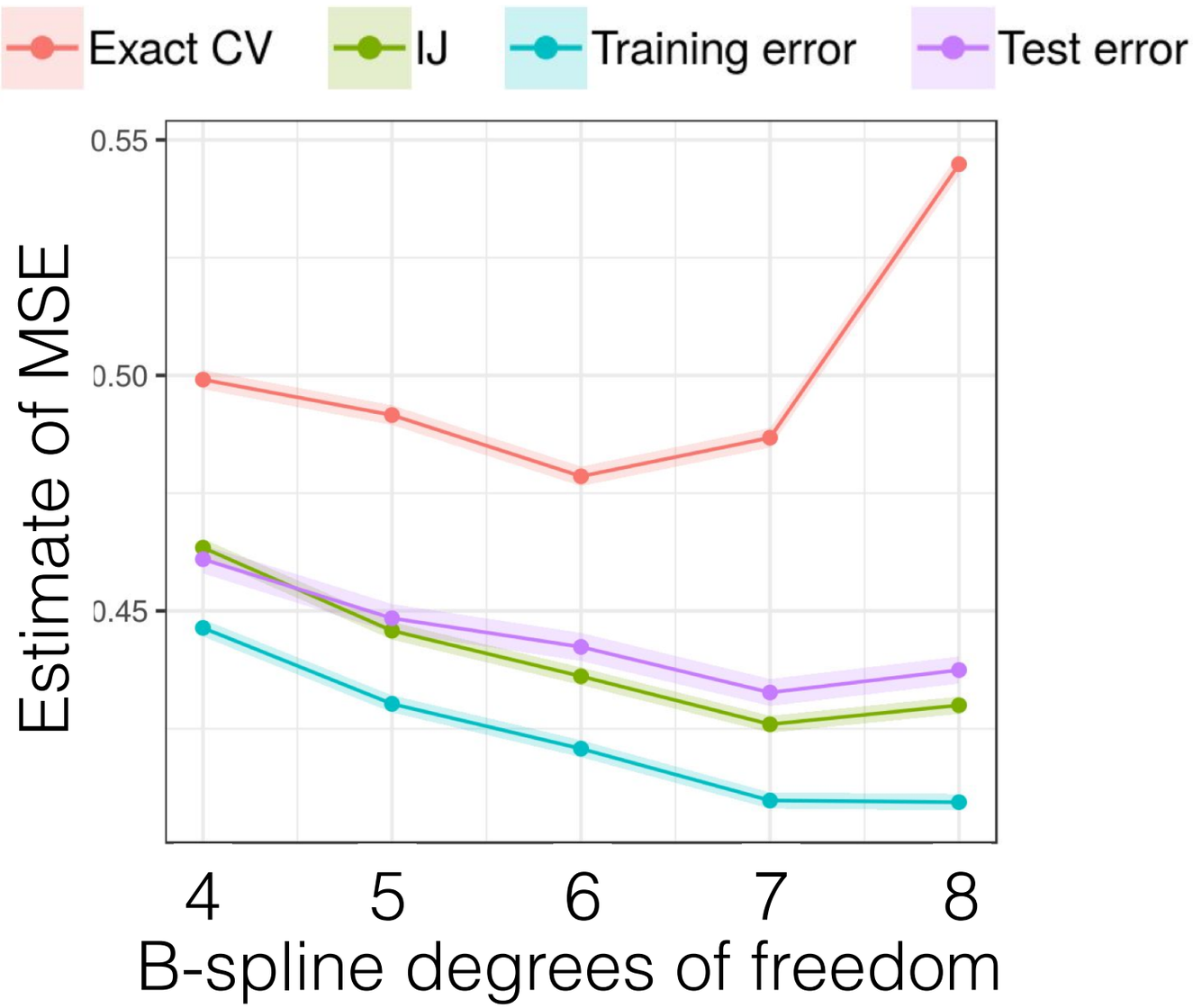
Swiss Army IJ: Experiments

- Mice gene expression (N=1000) time series, after influenza infection
- Two-stage analysis: Regress gene expression on B-spline basis, then cluster (transformed) fits

[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Luan, Li 2003; Shoemaker et al 2015]

Swiss Army IJ: Experiments

- Mice gene expression (N=1000) time series, after influenza infection
- Two-stage analysis: Regress gene expression on B-spline basis, then cluster (transformed) fits



[Giordano, Stephenson, Liu, Jordan, Broderick 2019; Luan, Li 2003; Shoemaker et al 2015]

Roadmap

- Cross validation (CV) setup
- CV Challenge #1: an expensive data analysis
- Solution #1: a Taylor series approximation ("Swiss Army infinitesimal jackknife")
- CV Challenge #2: high dimensions
- Solution #2: using sparsity (carefully)

Roadmap

- Cross validation (CV) setup
- CV Challenge #1: an expensive data analysis
- Solution #1: a Taylor series approximation ("Swiss Army infinitesimal jackknife")
- CV Challenge #2: high dimensions
- Solution #2: using sparsity (carefully)

Approximation for high dimensions

- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis)

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis)

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis)
- Recall:

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis)

- Recall:

$$H_{d_1 d_2} = \frac{\partial^2}{\partial \theta_{d_1} \partial \theta_{d_2}} \text{objective}$$

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis)

- Recall:

$$H_{d_1 d_2} = \frac{\partial^2}{\partial \theta_{d_1} \partial \theta_{d_2}} \text{objective}$$

D x *D*

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis)

- Recall:

$$H_{d_1 d_2} = \frac{\partial^2}{\partial \theta_{d_1} \partial \theta_{d_2}} \text{objective}$$

D x *D*

$$G_{dn} = \frac{\partial^2}{\partial \theta_d \partial w_n} \text{objective}$$

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis)

- Recall:

$$H_{d_1 d_2} = \frac{\partial^2}{\partial \theta_{d_1} \partial \theta_{d_2}} \text{objective}$$

D x *D*

$$G_{dn} = \frac{\partial^2}{\partial \theta_d \partial w_n} \text{objective}$$

D x *N*

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis) + O(D^3)

- Recall:

$$H_{d_1 d_2} = \frac{\partial^2}{\partial\theta_{d_1}\partial\theta_{d_2}}\text{objective}$$

D x *D*

$$G_{dn} = \frac{\partial^2}{\partial\theta_d \partial w_n}\text{objective}$$

D x *N*

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis) + O(D^3)

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis) + O(D^3)

- Accuracy also bad in high dims

[Stephenson & Broderick 2019]

Approximation for high dimensions

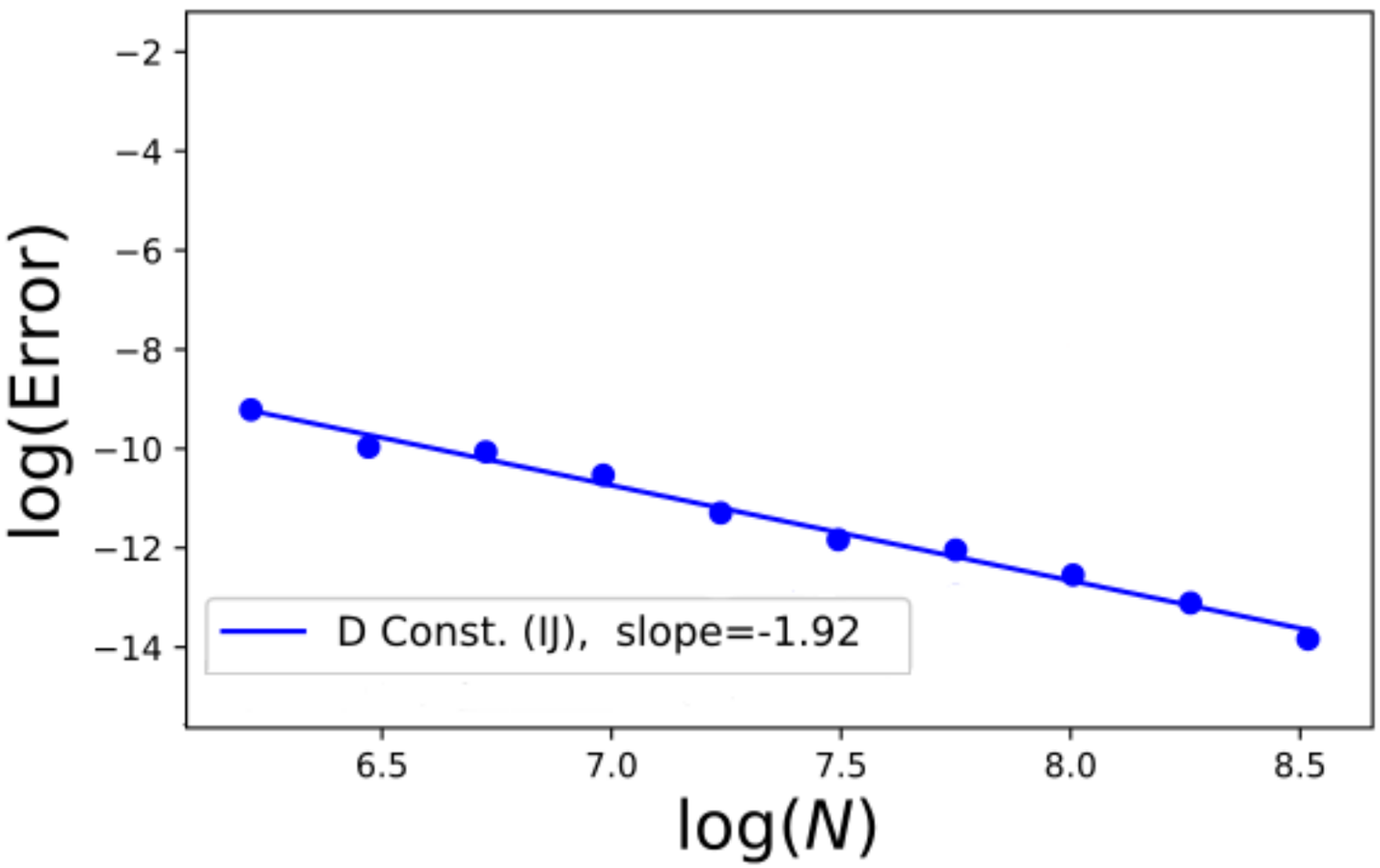
- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis) + O(D^3)



- Accuracy also bad in high dims

[Stephenson & Broderick 2019]

Approximation for high dimensions

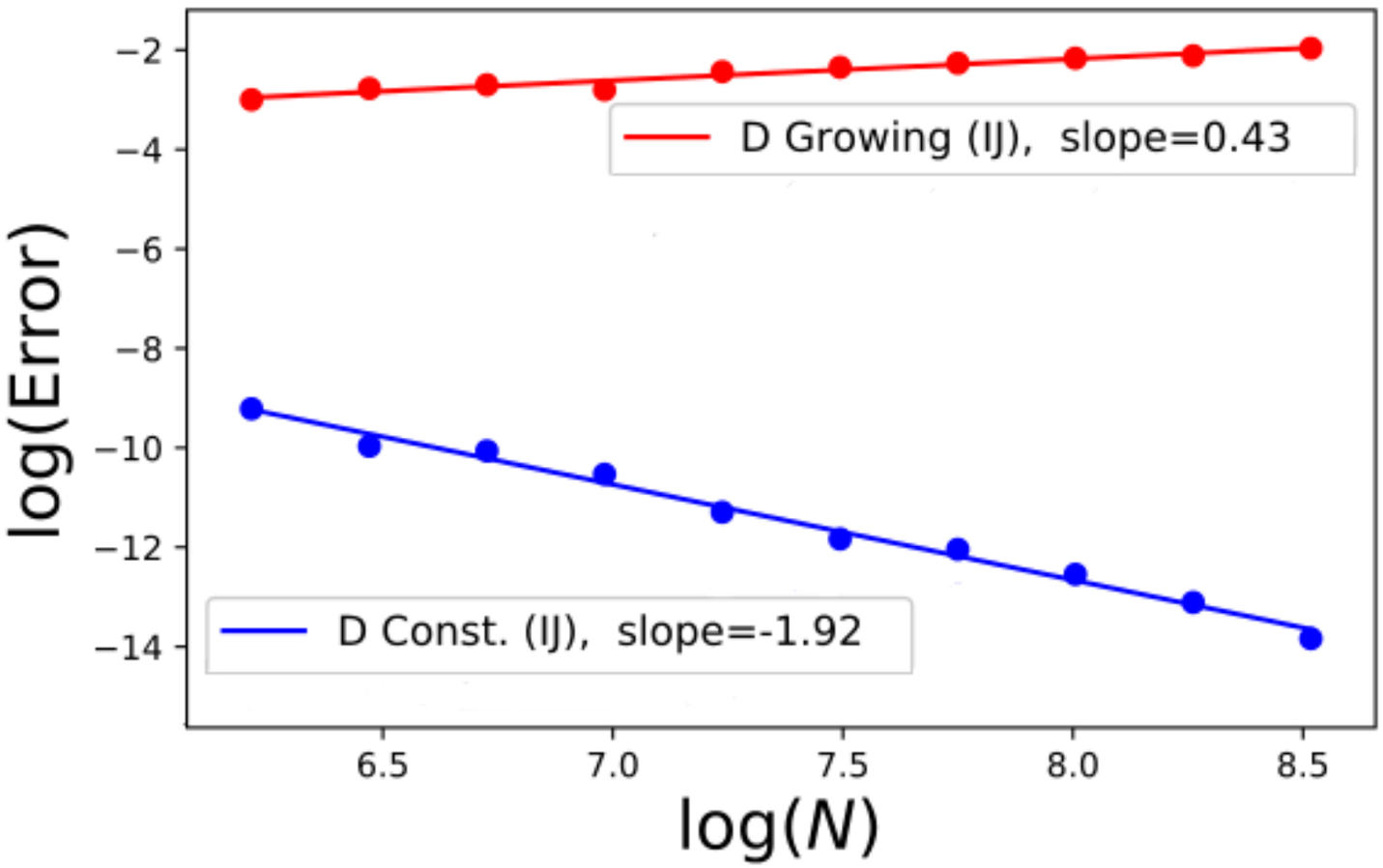
- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis) + O(D^3)



- Accuracy also bad in high dims

[Stephenson & Broderick 2019]

Approximation for high dimensions

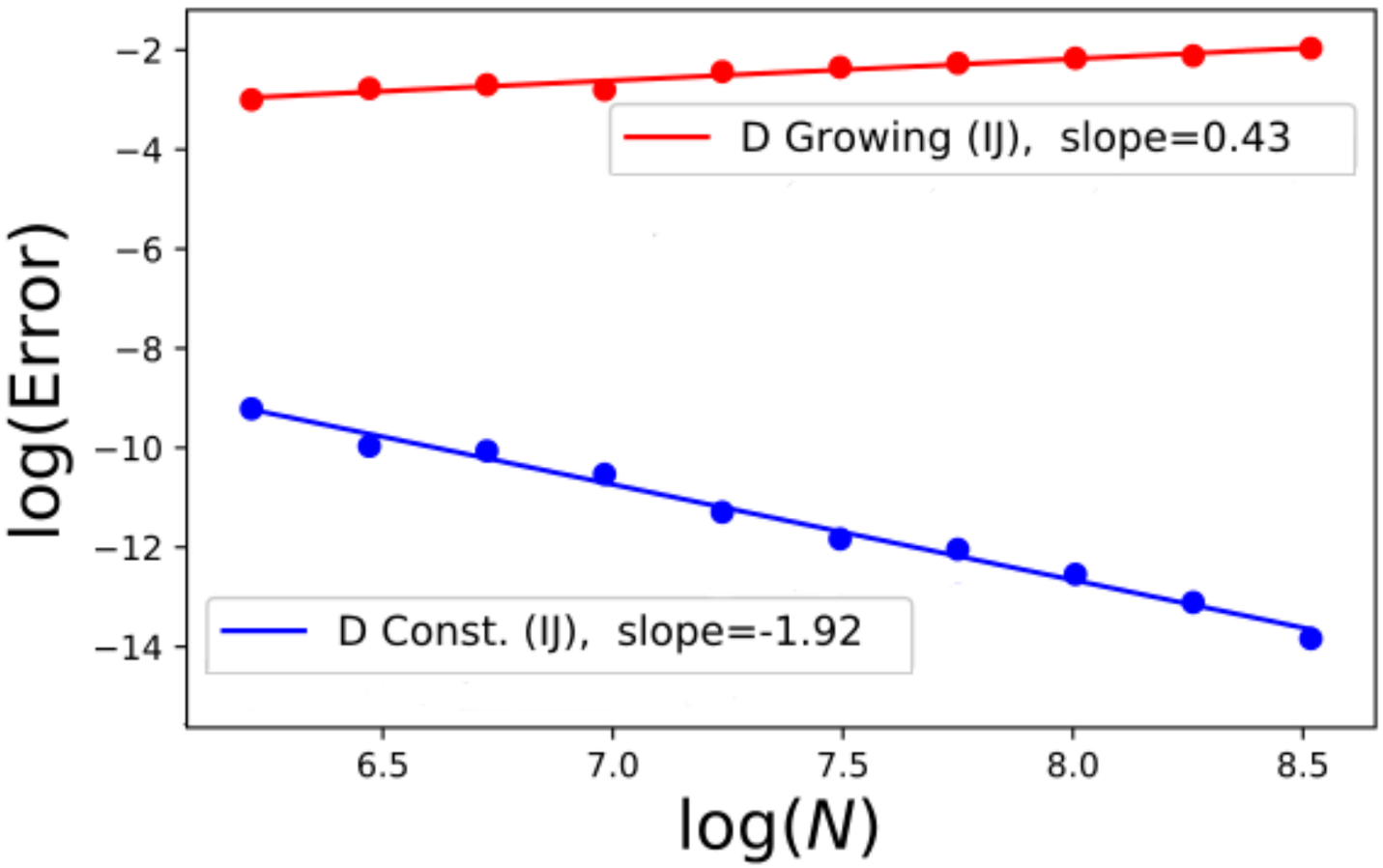
- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- Cost: (1 data analysis) + O(D^3)



- Accuracy also bad in high dims
- Problem doesn't go away on its own when there's a smaller "effective dimension" D_{eff}

[Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

Does it work?

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

- Further approximation via randomized solvers

Does it work?

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

- Further approximation via randomized solvers

Does it work?

- Can be fast but additional error

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

- Further approximation via randomized solvers
- Use Swiss Army IJ with L1 regularization

Does it work?

- Can be fast but additional error

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

- Further approximation via randomized solvers
- Use Swiss Army IJ with L1 regularization

Does it work?

- Can be fast but additional error
- L1 penalty not differentiable

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

- Further approximation via randomized solvers
- Use Swiss Army IJ with L1 regularization
- Use Swiss Army IJ with smooth approximation to L1

Does it work?

- Can be fast but additional error
- L1 penalty not differentiable

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

- Further approximation via randomized solvers
- Use Swiss Army IJ with L1 regularization
- Use Swiss Army IJ with smooth approximation to L1

Does it work?

- Can be fast but additional error
- L1 penalty not differentiable
- We show is inaccurate, slow

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

- Further approximation via randomized solvers
- Use Swiss Army IJ with L1 regularization
- Use Swiss Army IJ with smooth approximation to L1
- Run L1 regularization *then* IJ on the (small) support

Does it work?

- Can be fast but additional error
- L1 penalty not differentiable
- We show is inaccurate, slow

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

Approximation for high dimensions

- IJ approximation:

H: Hessian of the objective

G: matrix of gradients

$$\hat{\theta}_{IJ}(w) = \hat{\theta}(1_N) - H(1_N)^{-1} G(1_N)(w - 1_N)$$

- What to do?

Ideas:

- Further approximation via randomized solvers
- Use Swiss Army IJ with L1 regularization
- Use Swiss Army IJ with smooth approximation to L1
- Run L1 regularization *then* IJ on the (small) support

Does it work?

- Can be fast but additional error
- L1 penalty not differentiable
- We show is inaccurate, slow
- Yes! Accurate, fast, supported by our theory

[Koh and Liang 2017; Wang et al 2018; Rad & Maleki 2018; Stephenson & Broderick 2019]

High dimensions: Experiments

[Stephenson & Broderick 2019]

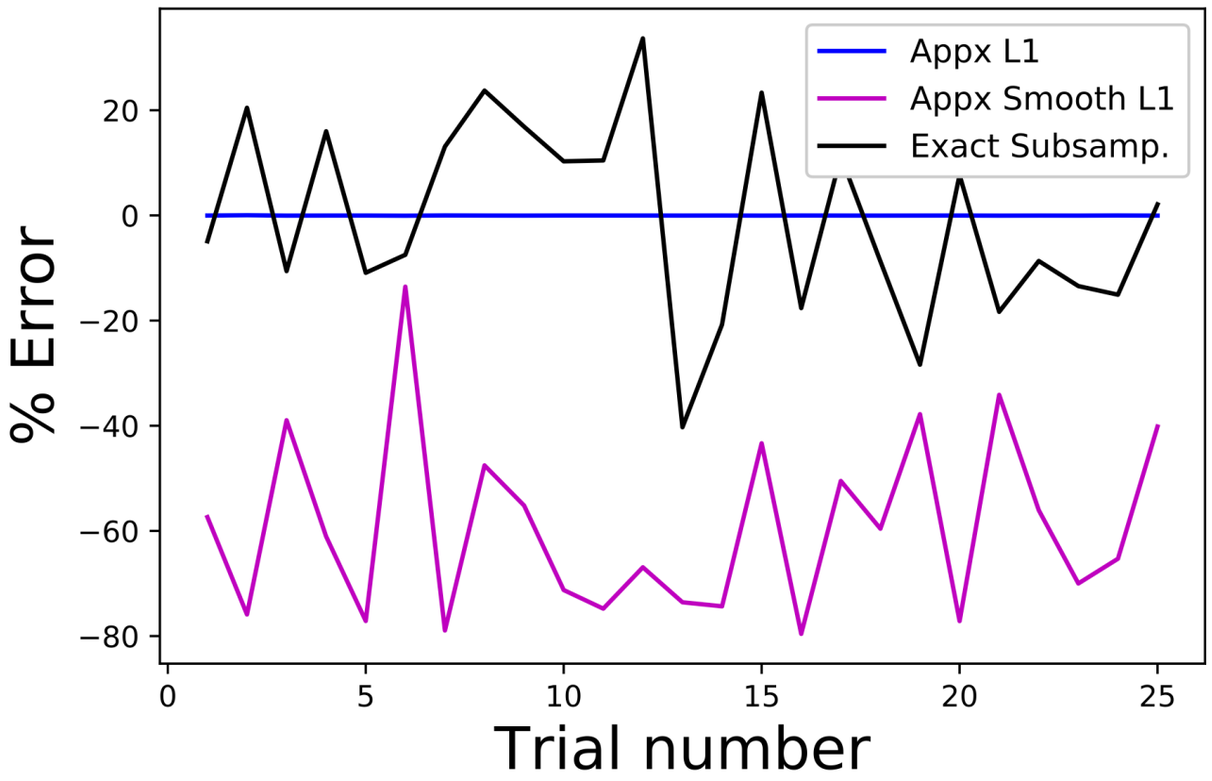
High dimensions: Experiments

- Simulated data (N=500, D=40,000)

[Stephenson & Broderick 2019]

High dimensions: Experiments

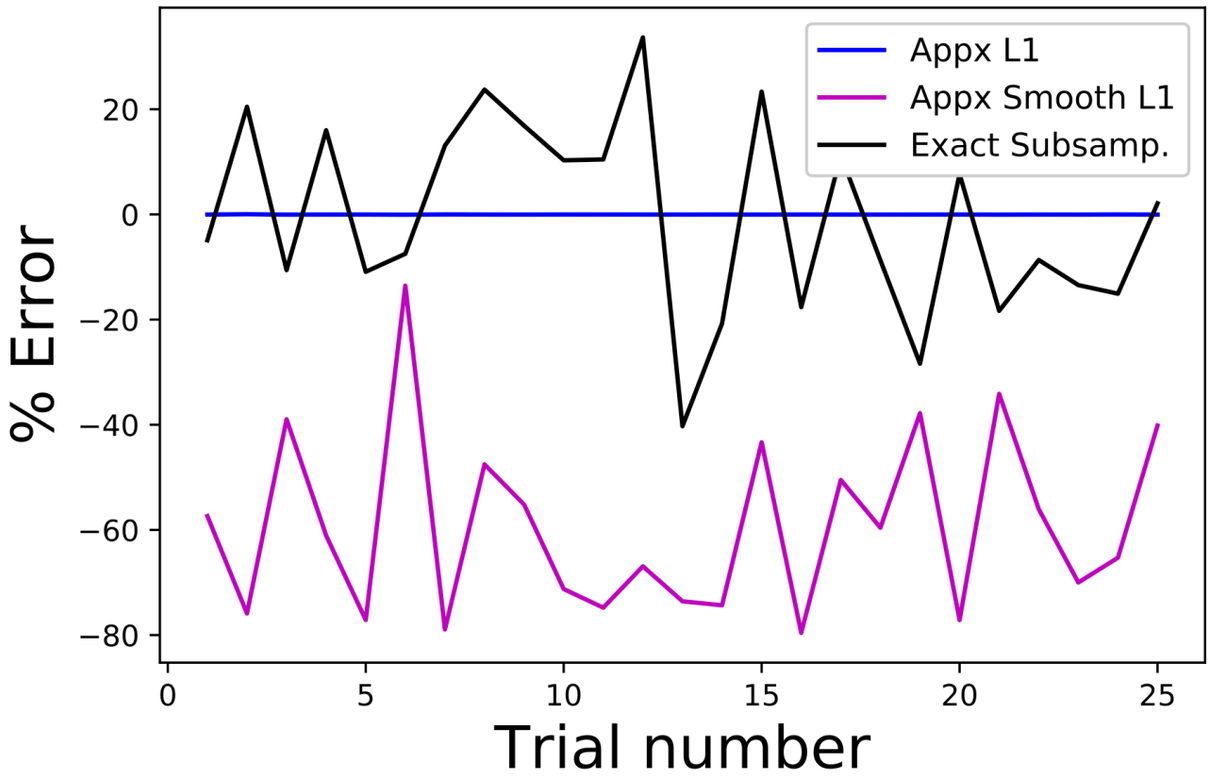
- Simulated data (N=500, D=40,000)

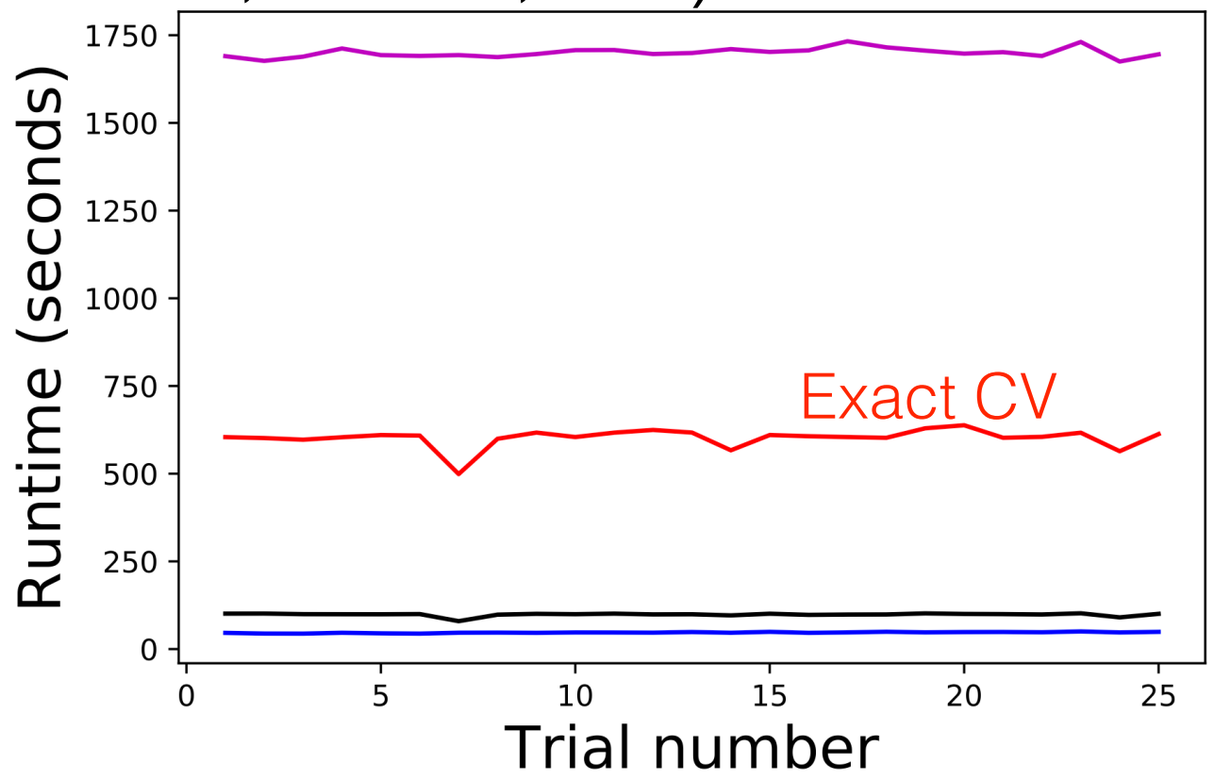


[Stephenson & Broderick 2019]

High dimensions: Experiments

- Simulated data (N=500, D=40,000)

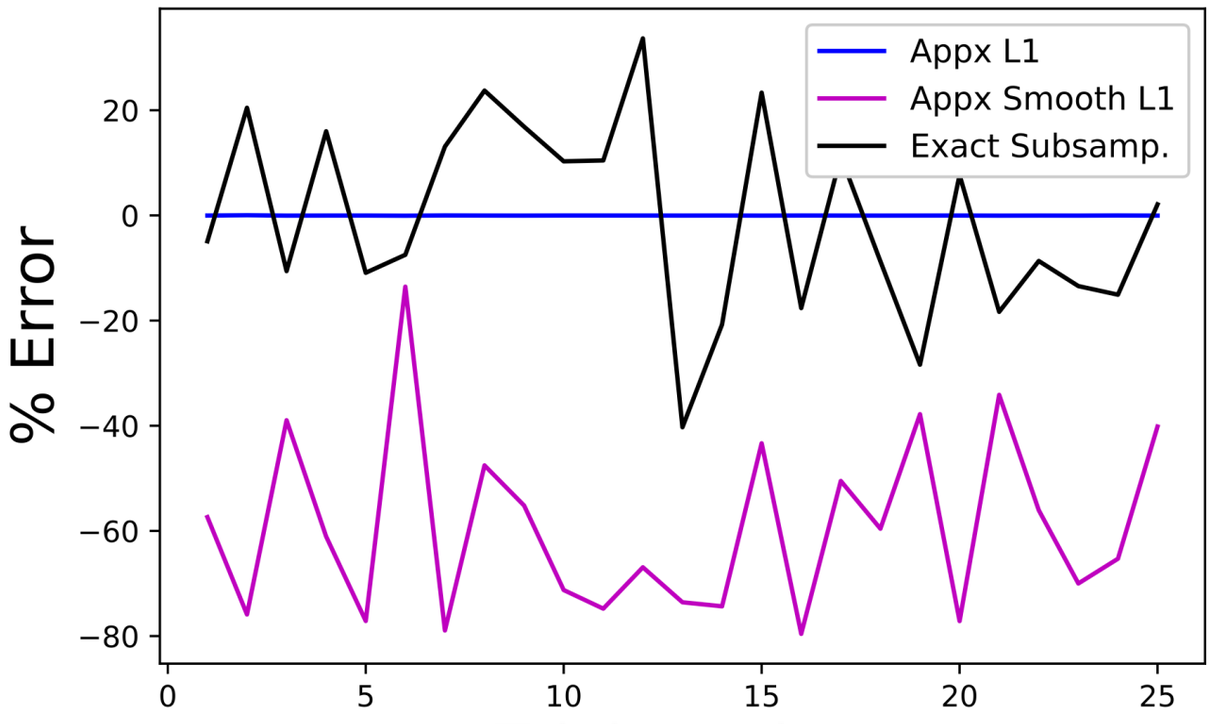


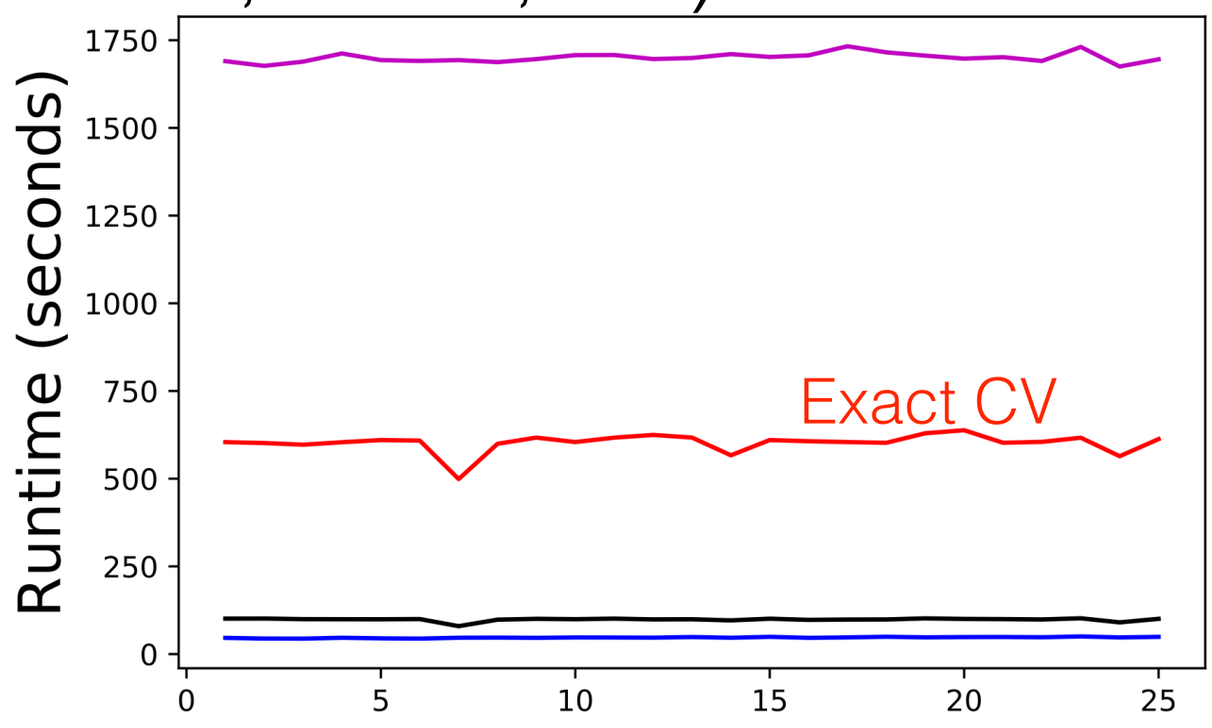


[Stephenson & Broderick 2019]

High dimensions: Experiments

- Simulated data (N=500, D=40,000)



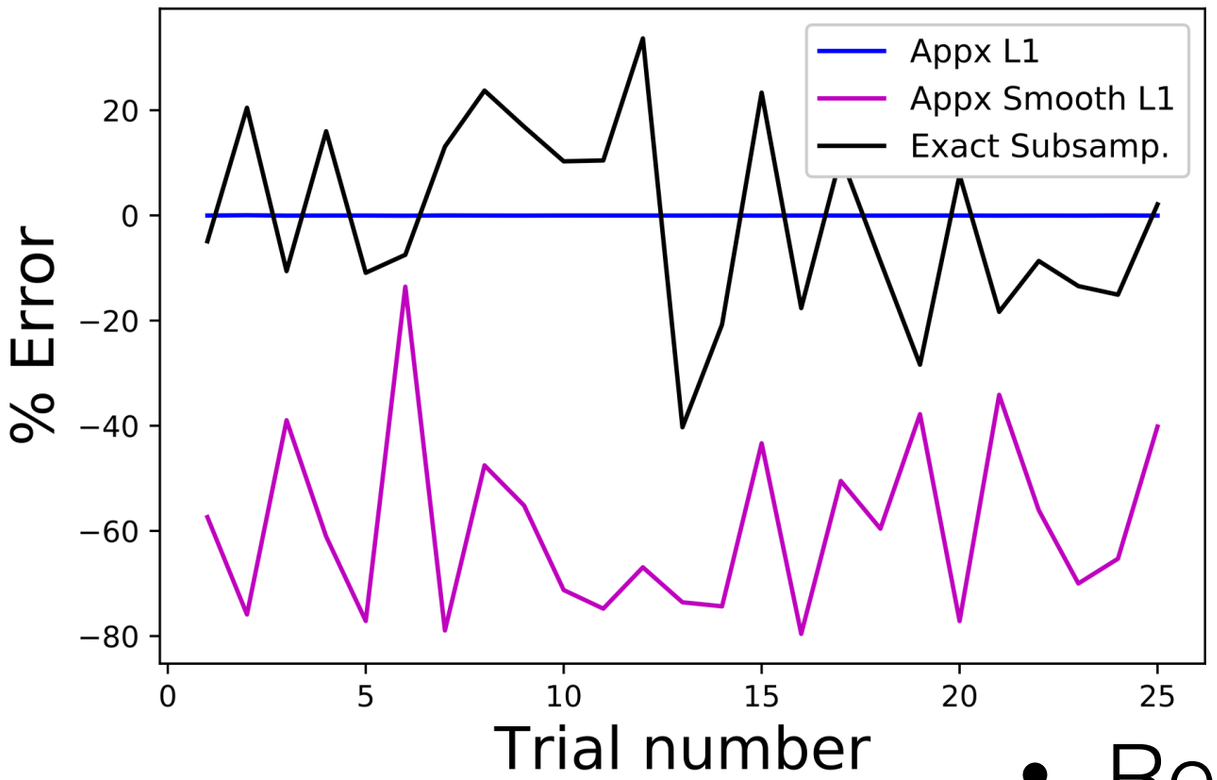


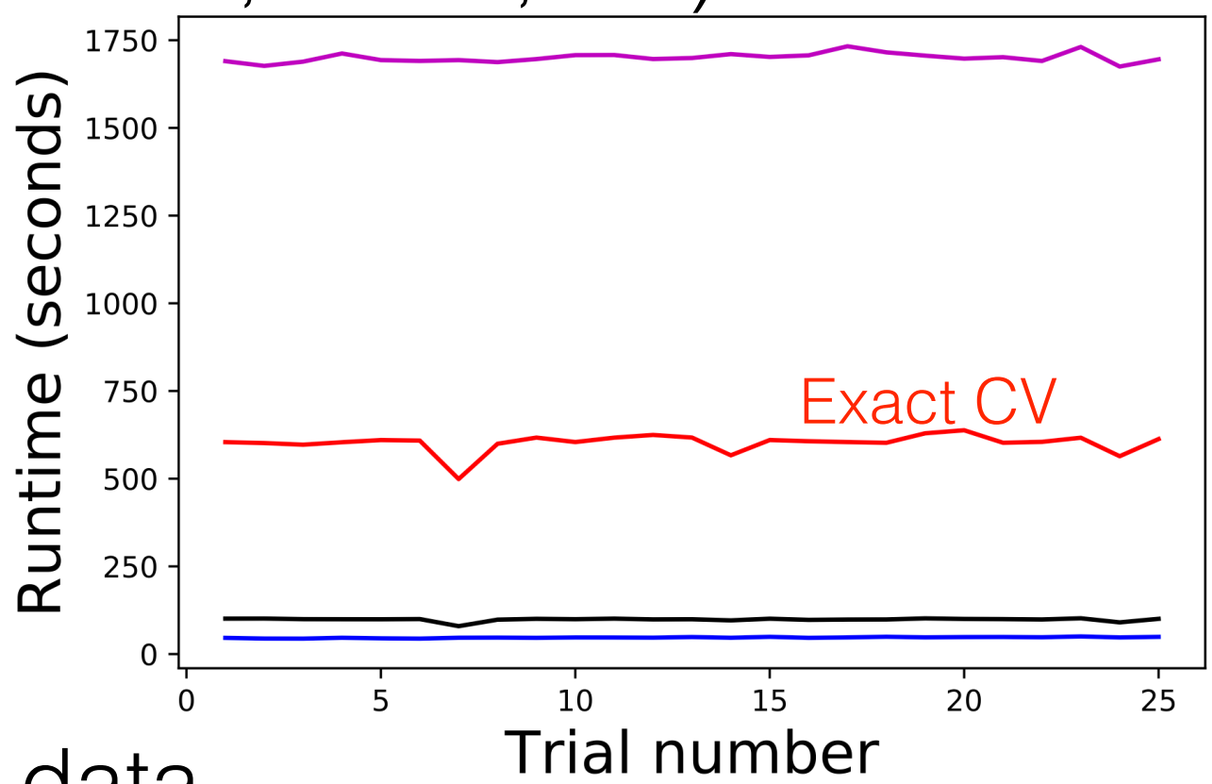
- Real data

[Stephenson & Broderick 2019]

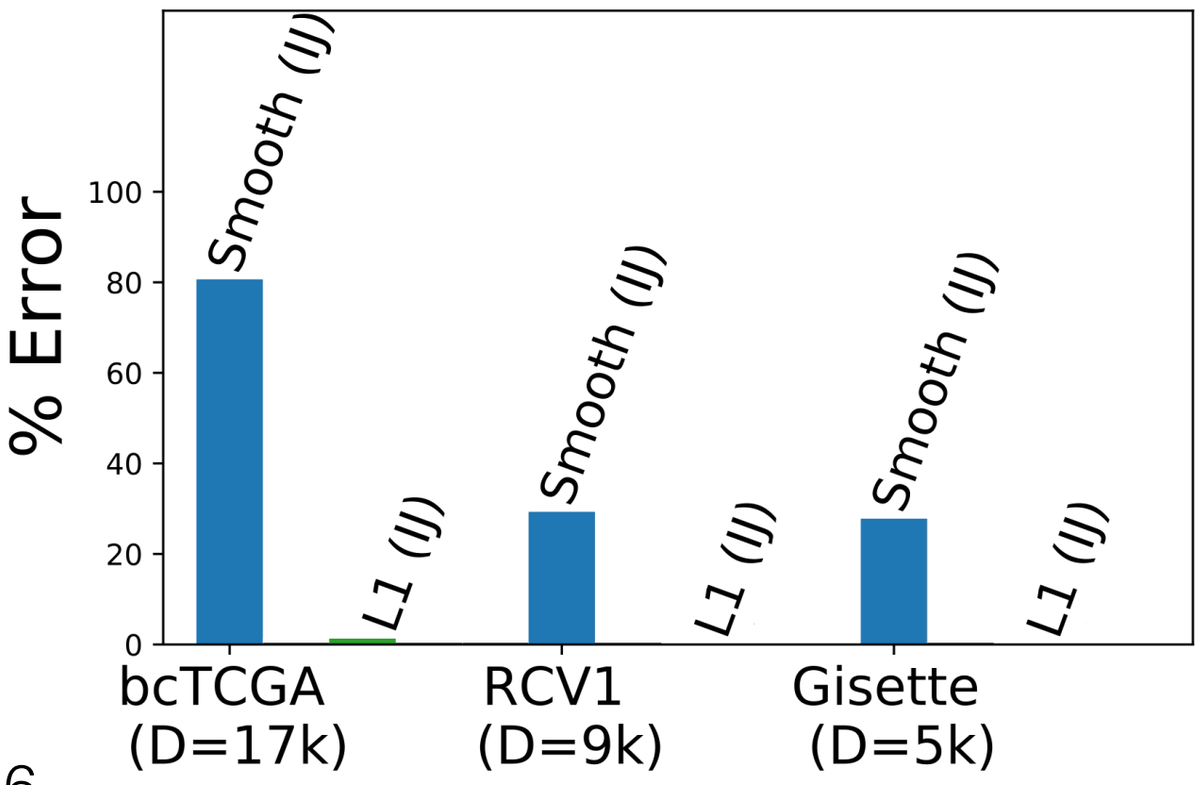
High dimensions: Experiments

- Simulated data (N=500, D=40,000)





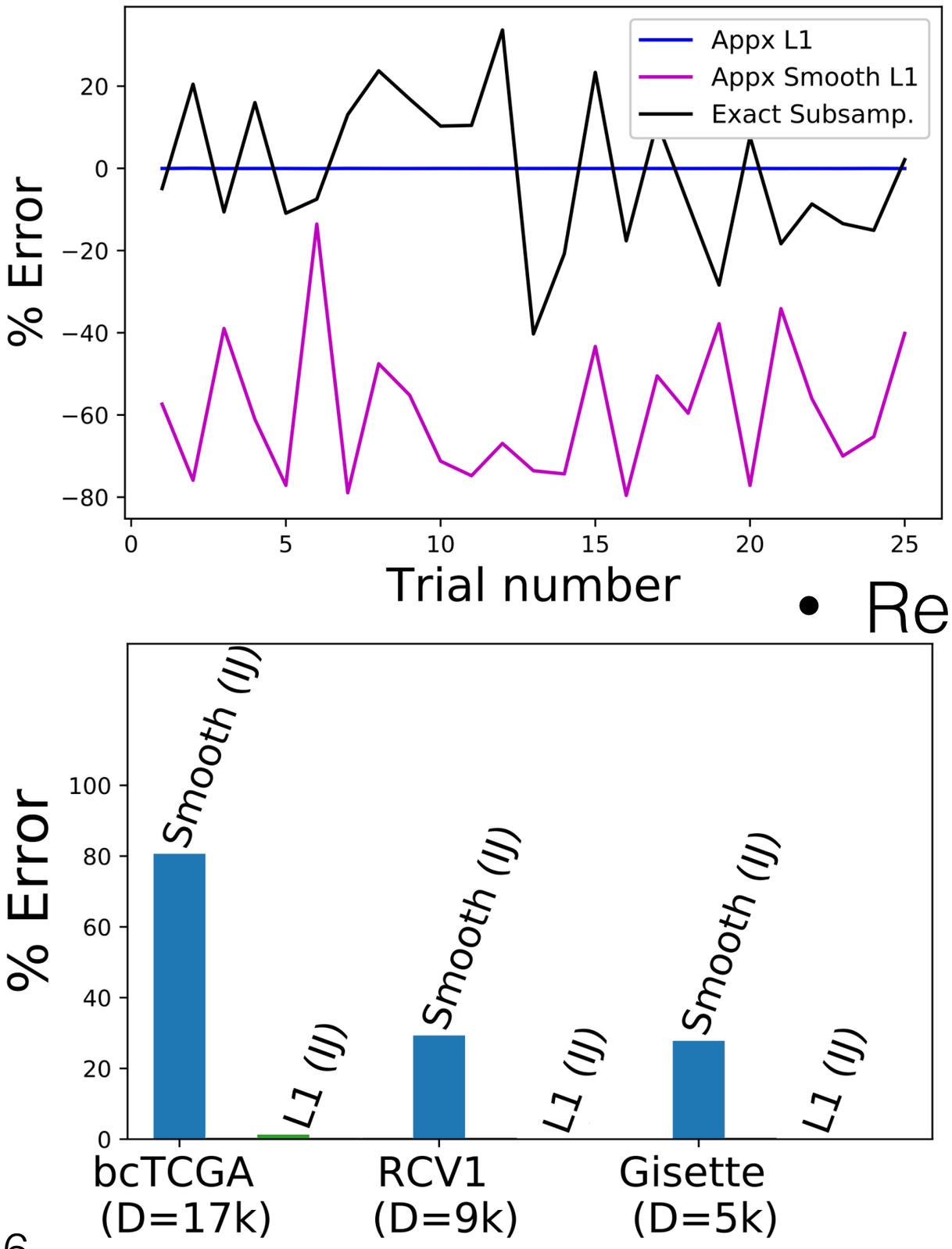
- Real data

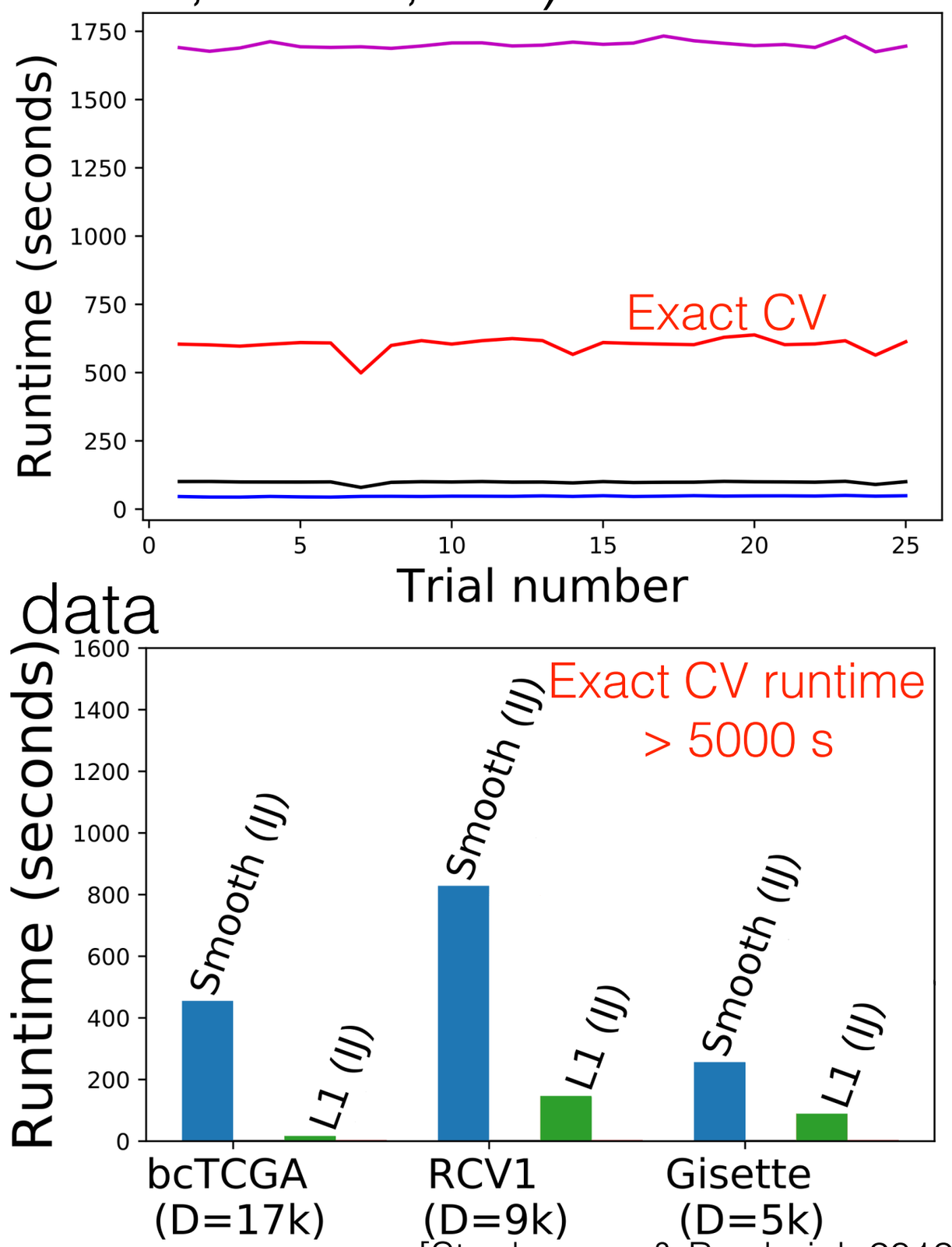


[Stephenson & Broderick 2019]

High dimensions: Experiments

- Simulated data (N=500, D=40,000)





- Real data

[Stephenson & Broderick 2019]

Conclusions

Conclusions

- Note: On a larger RCV1 data set, we estimate full CV takes > 2 weeks. Our method takes 3 minutes.

Conclusions

- Note: On a larger RCV1 data set, we estimate full CV takes > 2 weeks. Our method takes 3 minutes.
- Our CV approximations require *just 1 algorithm run*

Conclusions

- Note: On a larger RCV1 data set, we estimate full CV takes > 2 weeks. Our method takes 3 minutes.
- Our CV approximations require *just 1 algorithm run*
  - Taylor expansion + L1-regularization

Conclusions

- Note: On a larger RCV1 data set, we estimate full CV takes > 2 weeks. Our method takes 3 minutes.
- Our CV approximations require *just 1 algorithm run*
  - Taylor expansion + L1-regularization
  - We show: Fast, automatic, and high-quality

Conclusions

- Note: On a larger RCV1 data set, we estimate full CV takes > 2 weeks. Our method takes 3 minutes.
- Our CV approximations require *just 1 algorithm run*
  - Taylor expansion + L1-regularization
  - We show: Fast, automatic, and high-quality

R Giordano, W Stephenson, R Liu, MI Jordan, and T Broderick. A Swiss army infinitesimal jackknife. *AISTATS* 2019. (Notable Paper Award winner)

- https://github.com/rgiordan/AISTATS2019SwissArmyIJ
- https://github.com/rgiordan/vittles

W Stephenson and T Broderick. Approximate cross validation in high dimensions with guarantees. *AISTATS* 2020, to appear. ArXiv:1905.13657.

References

R Giordano, W Stephenson, R Liu, MI Jordan, and T Broderick. A Swiss army infinitesimal jackknife. *AISTATS* 2019. (Notable Paper Award winner)

- https://github.com/rgiordan/AISTATS2019SwissArmyIJ
- https://github.com/rgiordan/vittles

W Stephenson and T Broderick. Approximate cross validation in high dimensions with guarantees. *AISTATS* 2020, to appear. ArXiv:1905.13657.

References

R Agrawal, T Campbell, JH Huggins, and T Broderick. Data-dependent compression of random features for large-scale kernel approximation. *AISTATS* 2019.

T Campbell and T Broderick. Automated scalable Bayesian inference via Hilbert coresets. *Journal of Machine Learning Research*, 2019.

T Campbell and T Broderick. Bayesian Coreset Construction via Greedy Iterative Geodesic Ascent. *ICML* 2018.

JH Huggins, T Campbell, and T Broderick. Coresets for scalable Bayesian logistic regression. *NeurIPS* 2016.

R Giordano, W Stephenson, R Liu, MI Jordan, and T Broderick. A Swiss army infinitesimal jackknife. *AISTATS* 2019. (Notable Paper Award winner)

- https://github.com/rgiordan/AISTATS2019SwissArmyIJ
- https://github.com/rgiordan/vittles

W Stephenson and T Broderick. Approximate cross validation in high dimensions with guarantees. *AISTATS* 2020, to appear. ArXiv:1905.13657.