

Automatic differentiation for hyperparameter selection in non-smooth convex learning

Quentin Bertrand (Mila, Université de Montréal)
<https://QB3.github.io>

Joint work with Q. Klopfenstein, M. Massias, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon

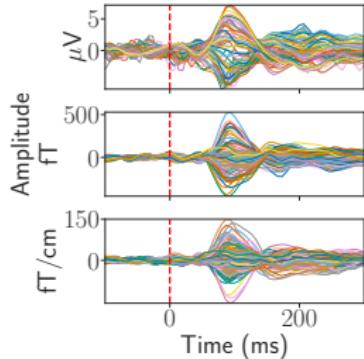
Neuroimaging data: EEG⁽¹⁾ and MEG⁽²⁾



EEG



MEG



M/EEG data Y

► Data Y : electric and magnetic fields at the head surface

⁽¹⁾ H. Berger. "Über das elektroenzephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).

⁽²⁾ D. Cohen. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).

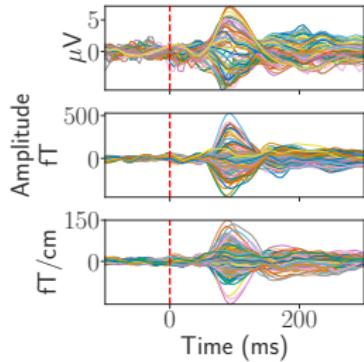
Neuroimaging data: EEG⁽¹⁾ and MEG⁽²⁾



EEG



MEG



M/EEG data Y

- ▶ **Data Y :** electric and magnetic fields at the head surface
- ▶ **Goal:** which parts of the brain are responsible for the signals?

⁽¹⁾H. Berger. "Über das elektroenzephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).

⁽²⁾D. Cohen. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).

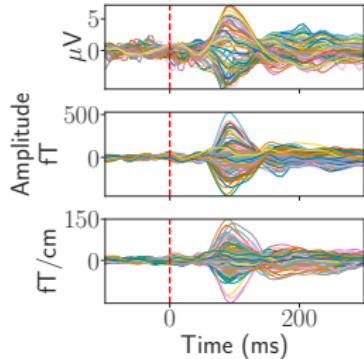
Neuroimaging data: EEG⁽¹⁾ and MEG⁽²⁾



EEG



MEG



M/EEG data Y

- ▶ **Data Y :** electric and magnetic fields at the head surface
- ▶ **Goal:** which parts of the brain are responsible for the signals?
- ▶ **Applications:** clinical and cognitive experiments

⁽¹⁾H. Berger. "Über das elektroenzephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).

⁽²⁾D. Cohen. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).

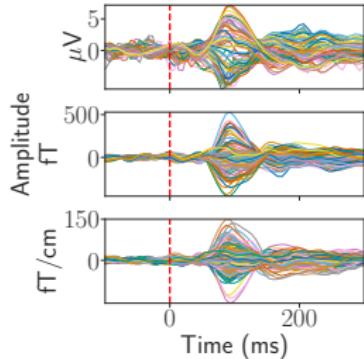
Neuroimaging data: EEG⁽¹⁾ and MEG⁽²⁾



EEG



MEG



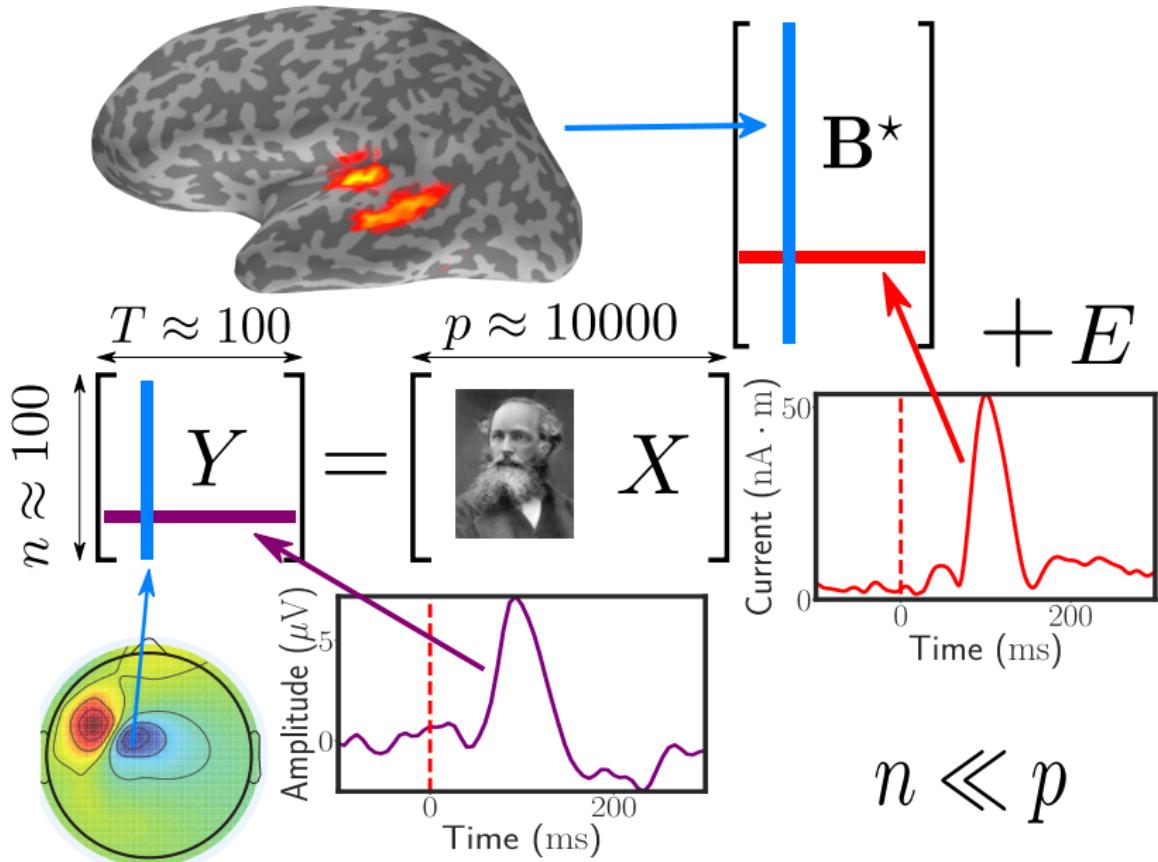
M/EEG data Y

- ▶ **Data Y :** electric and magnetic fields at the head surface
- ▶ **Goal:** which parts of the brain are responsible for the signals?
- ▶ **Applications:** clinical and cognitive experiments

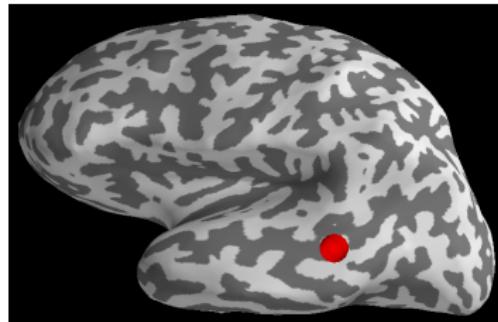
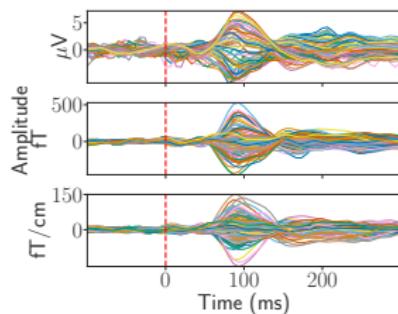
⁽¹⁾H. Berger. "Über das elektroenzephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).

⁽²⁾D. Cohen. "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science* (1968).

The M/EEG inverse problem



Summary of the problem setting



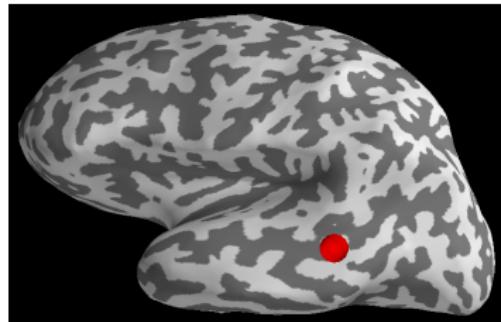
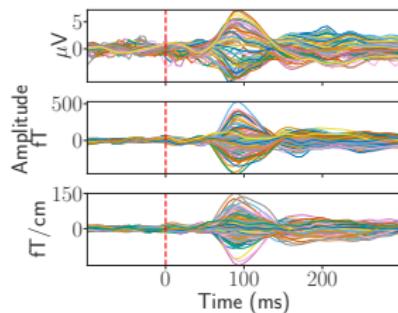
What you have: $\mathbf{Y} \in \mathbb{R}^{n \times T}$

What you want: $\mathbf{B} \in \mathbb{R}^{p \times T}$

This is typically done using optimization based estimators

$$\hat{\mathbf{B}} \in \arg \min_{\mathbf{B} \in \mathbb{R}^{p \times T}} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{X}\mathbf{B}\|_F^2 + \lambda \Omega(\mathbf{B}) \right)$$

Summary of the problem setting



What you have: $\mathbf{Y} \in \mathbb{R}^{n \times T}$

What you want: $\mathbf{B} \in \mathbb{R}^{p \times T}$

This is typically done using optimization based estimators

$$\hat{\mathbf{B}} \in \arg \min_{\mathbf{B} \in \mathbb{R}^{p \times T}} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{XB}\|_F^2 + \lambda \Omega(\mathbf{B}) \right)$$

Plan for today

$$\hat{\mathbf{B}} \in \arg \min_{\mathbf{B} \in \mathbb{R}^{p \times T}} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{X}\mathbf{B}\|_F^2 + \lambda \Omega(\mathbf{B}) \right)$$

Covered in this presentation

- ▶ ~~How to efficiently solve this optimization problem?~~⁽³⁾⁽⁴⁾
- ▶ How to efficiently select the regularization parameter λ ?^{(5),(6)}

⁽³⁾ Q. Bertrand and M. Massias. "Anderson acceleration of coordinate descent". In: *AISTATS*. 2021.

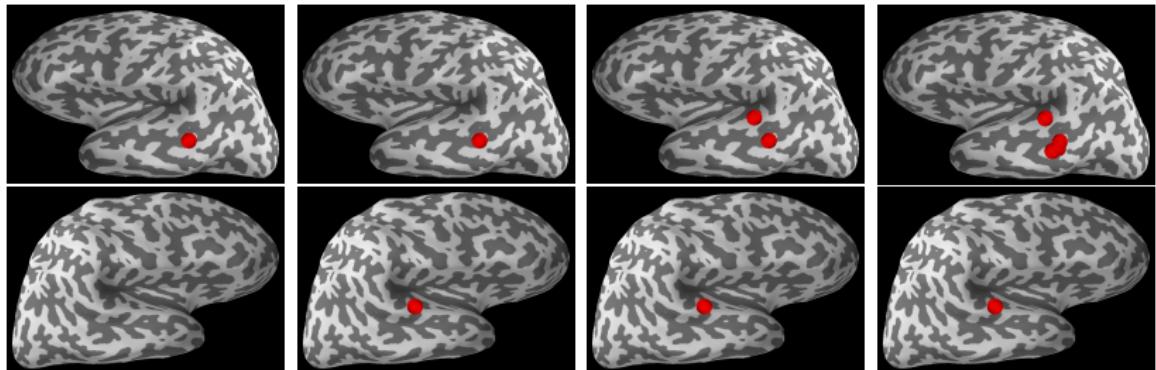
⁽⁴⁾ Q. Bertrand et al. "Beyond L1: Faster and Better Sparse Models with skglm". In: *arXiv preprint arXiv:2204.07826* (2022).

⁽⁵⁾ Q. Bertrand et al. "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML* (2020).

⁽⁶⁾ Q. Bertrand et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *JMLR* (2022).

Which λ to pick?

$$\hat{\mathbf{B}} \in \arg \min_{\mathbf{B} \in \mathbb{R}^{p \times T}} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{X}\mathbf{B}\|_F^2 + \lambda \|\mathbf{B}\|_{2,1} \right)$$



$$\lambda = 0.85\lambda_{\max}$$

$$\lambda = 0.82\lambda_{\max}$$

$$\lambda = 0.80\lambda_{\max}$$

$$\lambda = 0.75\lambda_{\max}$$

Real MEEG data. Brain source reconstruction using multitask Lasso with multiple λ . Which λ to pick? How to *automatically* select λ ?

- When $\lambda \geq \lambda_{\max}$, $\hat{\mathbf{B}} = 0$ no sources are recovered

Model selection techniques

- ▶ Statistical route^{(7), (8)}:
assumptions on the design matrix X
- ▶ Bayesian statistics^{(9), (10)}:
prior on λ
- ▶ Hyperparameter optimization^{(11), (12)}:
minimize a given criterion $\mathcal{C}(\hat{\beta}^{(\lambda)})$

(7) K. Lounici. "Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators". In: *Electron. J. Stat.* (2008).

(8) K. Lounici et al. "Taking Advantage of Sparsity in Multi-Task Learning". In: *arXiv preprint arXiv:0903.1468* (2009).

(9) M. E. Tipping. "Sparse Bayesian learning and the relevance vector machine". In: *Journal of Machine Learning Research* (2001).

(10) M. Figueiredo. "Adaptive Sparseness Using Jeffreys Prior.". In: *NeurIPS*. 2001.

(11) R. Kohavi and G. H. John. "Automatic parameter selection by minimizing estimated error". In: *Machine Learning Proceedings*. 1995.

(12) F. Hutter, J. Lücke, and L. Schmidt-Thieme. "Beyond manual tuning of hyperparameters". In: *KI-Künstliche Intelligenz* (2015).

Hyperparameter optimization (HO)

Possible selection criterion:

- ▶ Good generalization^{(13), (14)} of $\hat{\beta}^{(\lambda)}$
- ▶ AIC/BIC,⁽¹⁵⁾ SURE⁽¹⁶⁾ that controls model complexity

⁽¹³⁾ L. R. A. Stone and J.C. Ramer. "Estimating WAIS IQ from Shipley Scale scores: Another cross-validation". In: *Journal of clinical psychology* 21.3 (1965), pp. 297–297.

⁽¹⁴⁾ K. Lounici, K. Meziani, and B. Riu. "Muddling Labels for Regularization, a novel approach to generalization". In: *arXiv preprint arXiv:2102.08769* (2021).

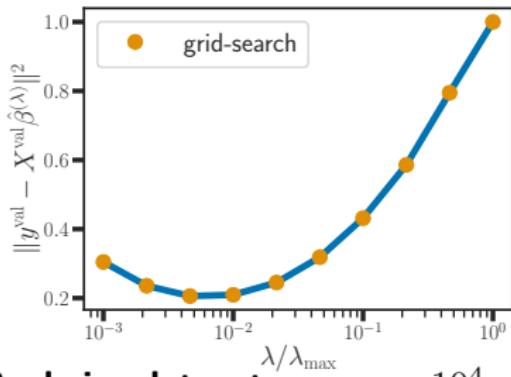
⁽¹⁵⁾ W. Liu, Y. Yang, et al. "Parametric or nonparametric? A parametricness index for model selection". In: *Ann. Statist.* 39.4 (2011), pp. 2074–2102.

⁽¹⁶⁾ C. M. Stein. "Estimation of the mean of a multivariate normal distribution". In: *Ann. Statist.* 9.6 (1981), pp. 1135–1151.

Hyperparameter optimization (HO)

Possible selection criterion:

- ▶ Good generalization^{(13), (14)} of $\hat{\beta}^{(\lambda)}$
- ▶ AIC/BIC,⁽¹⁵⁾ SURE⁽¹⁶⁾ that controls model complexity



Real-sim dataset, $n \approx p \approx 10^4$
Validation loss as a function of λ .

Example

Model: Lasso

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \frac{\|y^{\text{train}} - X^{\text{train}} \beta\|^2}{2n} + \lambda \|\beta\|_1$$

Criterion: held-out loss

$$\arg \min_{\lambda} \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2$$

⁽¹³⁾ L. R. A. Stone and J.C. Ramer. "Estimating WAIS IQ from Shipley Scale scores: Another cross-validation". In: *Journal of clinical psychology* 21.3 (1965), pp. 297–297.

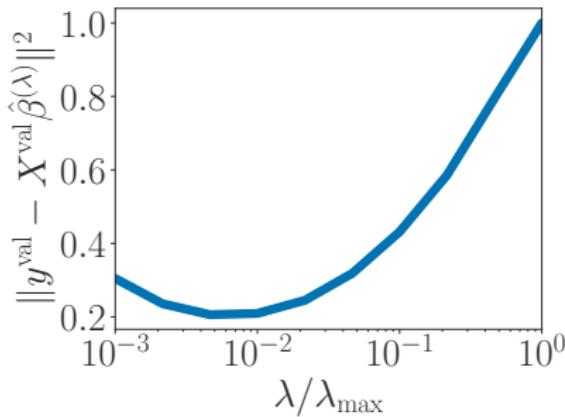
⁽¹⁴⁾ K. Lounici, K. Meziani, and B. Riu. "Muddling Labels for Regularization, a novel approach to generalization". In: *arXiv preprint arXiv:2102.08769* (2021).

⁽¹⁵⁾ W. Liu, Y. Yang, et al. "Parametric or nonparametric? A parametricness index for model selection". In: *Ann. Statist.* 39.4 (2011), pp. 2074–2102.

⁽¹⁶⁾ C. M. Stein. "Estimation of the mean of a multivariate normal distribution". In: *Ann. Statist.* 9.6 (1981), pp. 1135–1151.

HO as a bilevel optimization problem⁽¹⁷⁾⁽¹⁸⁾

$$\underbrace{\arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}}_{\text{outer optimization problem}}$$
$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \underbrace{\arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

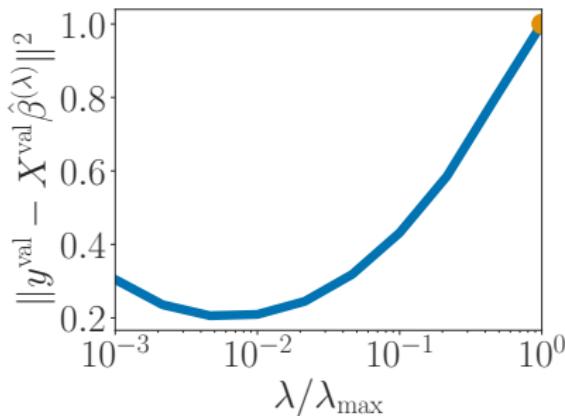


⁽¹⁷⁾ P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

⁽¹⁸⁾ F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

HO as a bilevel optimization problem⁽¹⁷⁾⁽¹⁸⁾

$$\underbrace{\arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}}_{\text{outer optimization problem}}$$
$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \underbrace{\arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

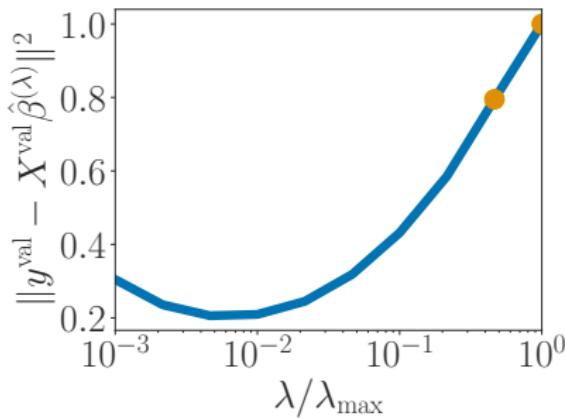


⁽¹⁷⁾ P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

⁽¹⁸⁾ F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

HO as a bilevel optimization problem⁽¹⁷⁾⁽¹⁸⁾

$$\underbrace{\arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}}_{\text{outer optimization problem}}$$
$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \underbrace{\arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

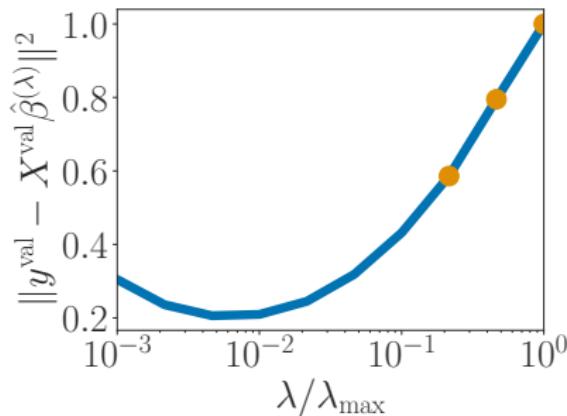


⁽¹⁷⁾ P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

⁽¹⁸⁾ F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

HO as a bilevel optimization problem⁽¹⁷⁾⁽¹⁸⁾

$$\underbrace{\arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}}_{\text{outer optimization problem}}$$
$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \underbrace{\arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

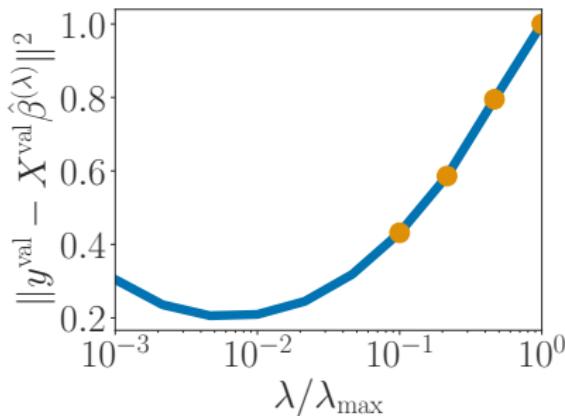


⁽¹⁷⁾ P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

⁽¹⁸⁾ F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

HO as a bilevel optimization problem⁽¹⁷⁾⁽¹⁸⁾

$$\underbrace{\arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}}_{\text{outer optimization problem}}$$
$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \underbrace{\arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

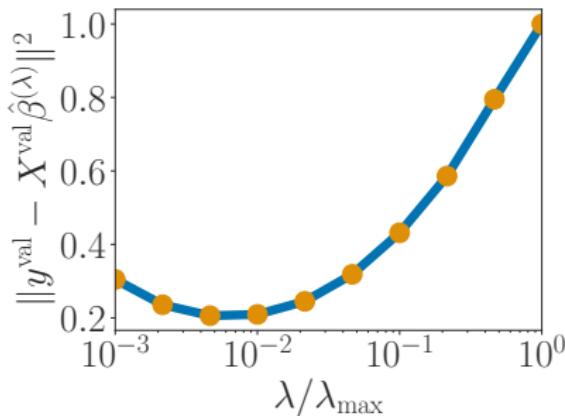


⁽¹⁷⁾ P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

⁽¹⁸⁾ F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

HO as a bilevel optimization problem⁽¹⁷⁾⁽¹⁸⁾

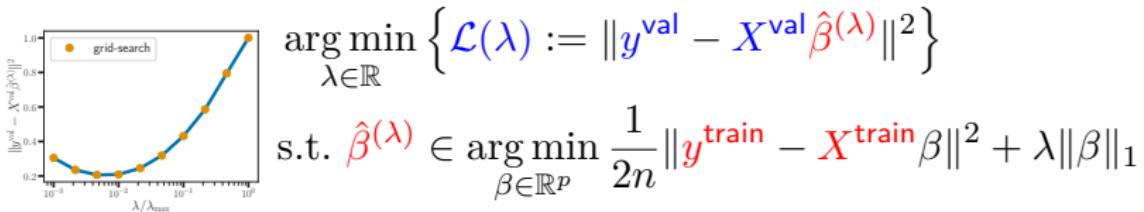
$$\underbrace{\arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}}_{\text{outer optimization problem}}$$
$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \underbrace{\arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$



⁽¹⁷⁾ P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

⁽¹⁸⁾ F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

Grid-search as a 0-order optimization method



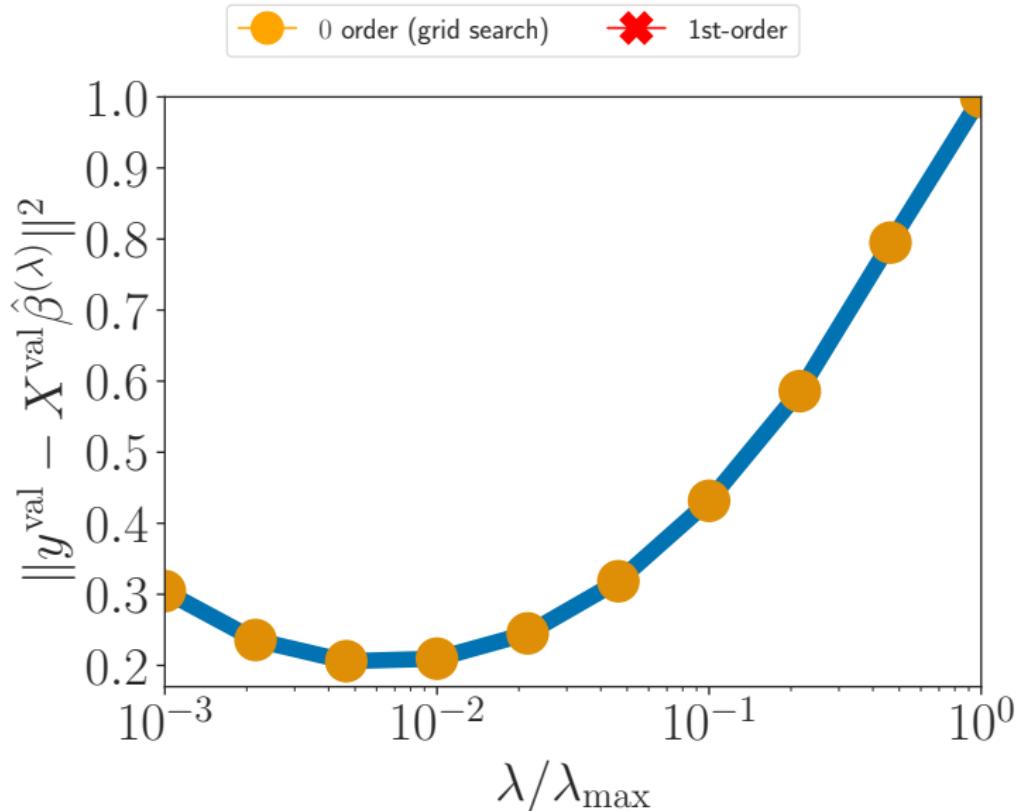
- ▶ Grid-search, random-search,⁽¹⁹⁾ SMBO⁽²⁰⁾: 0-order methods to solve bilevel optimization problem
- ▶ **Idea:** if \mathcal{L} is differentiable, use first-order optimization, i.e., compute $\nabla_\lambda \mathcal{L}$
- ▶ Once $\nabla_\lambda \mathcal{L}(\lambda)$ is computed, use gradient descent⁽²¹⁾:
$$\lambda^{(t+1)} = \lambda^{(t)} - \rho \nabla_\lambda \mathcal{L}(\lambda^{(t)}) \quad \text{with } \rho > 0$$

(19) J. Bergstra and Y. Bengio. "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* (2012).

(20) E. Brochu, V. M. Cora, and N. De Freitas. "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning". In: *arXiv preprint arXiv:1012.2599* (2010).

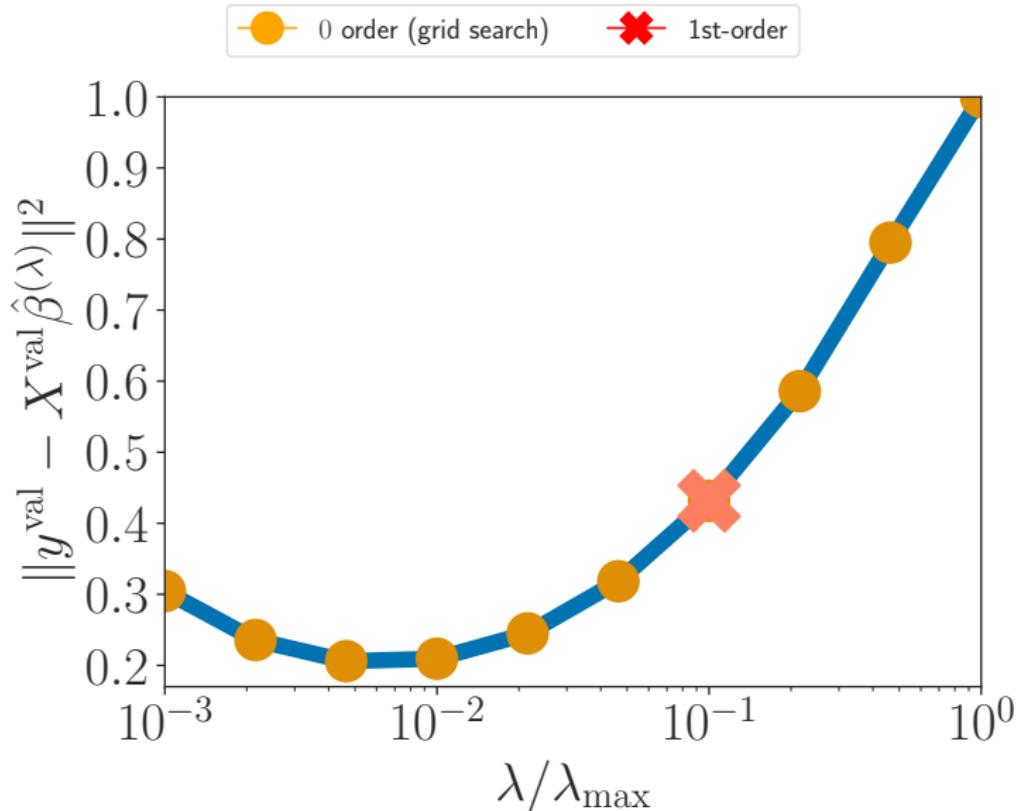
(21) F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

First-order optimization in λ , Lasso



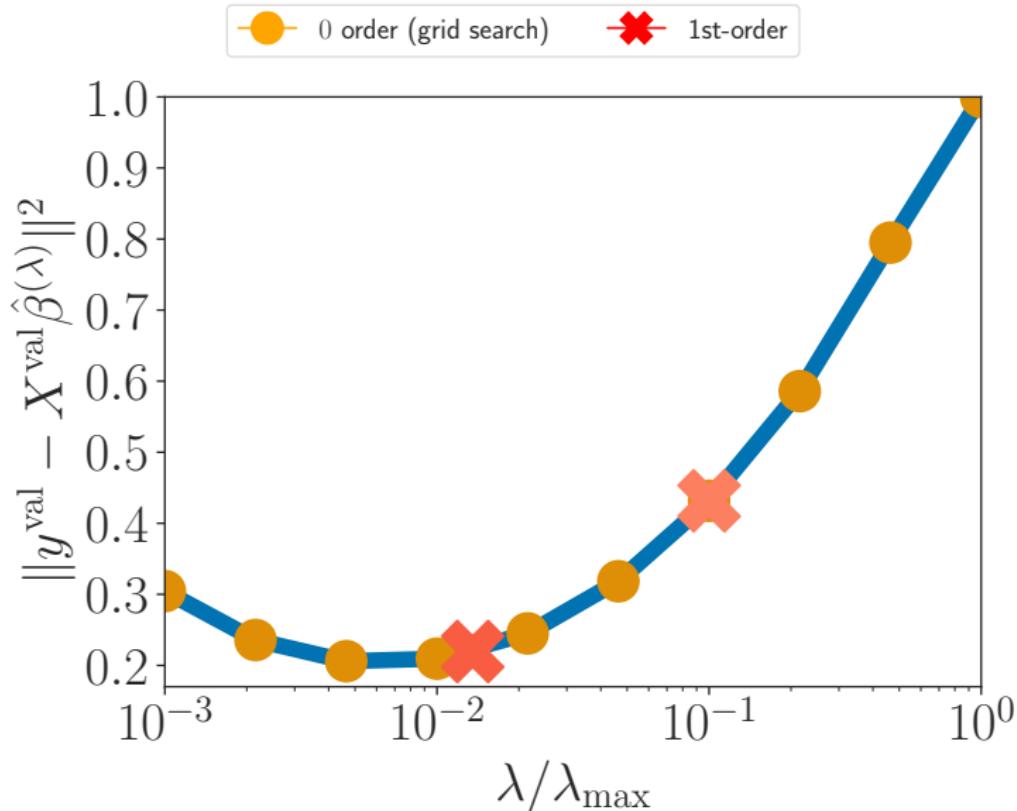
Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ .

First-order optimization in λ , Lasso



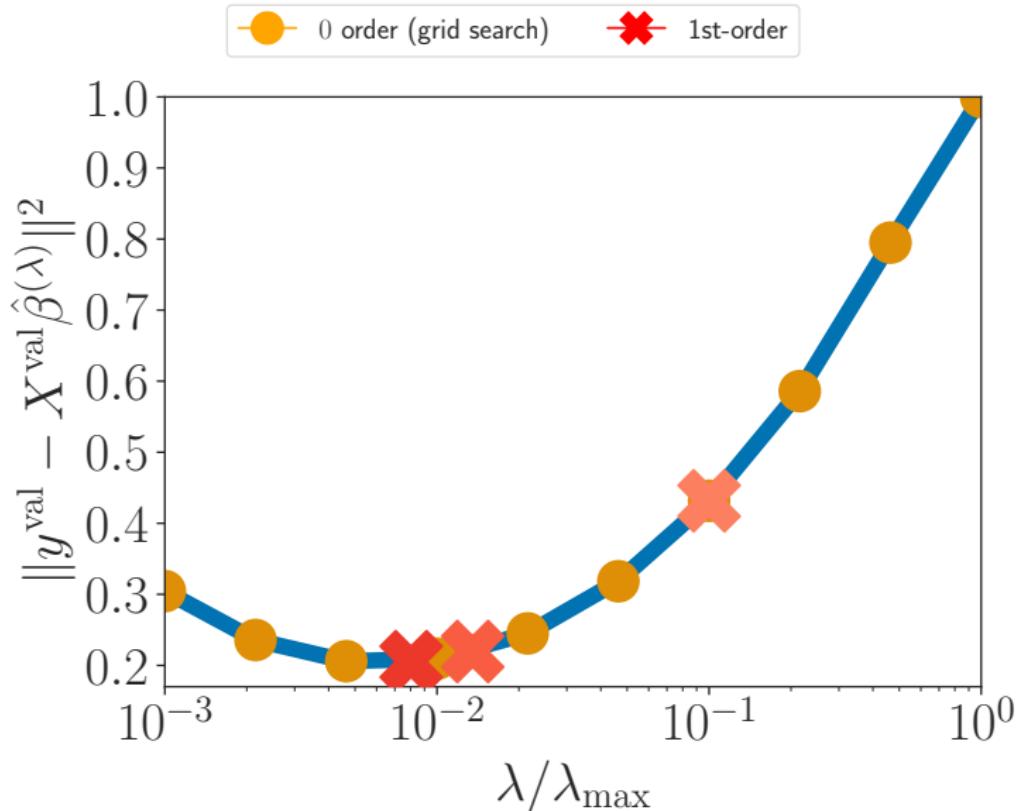
Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ .

First-order optimization in λ , Lasso



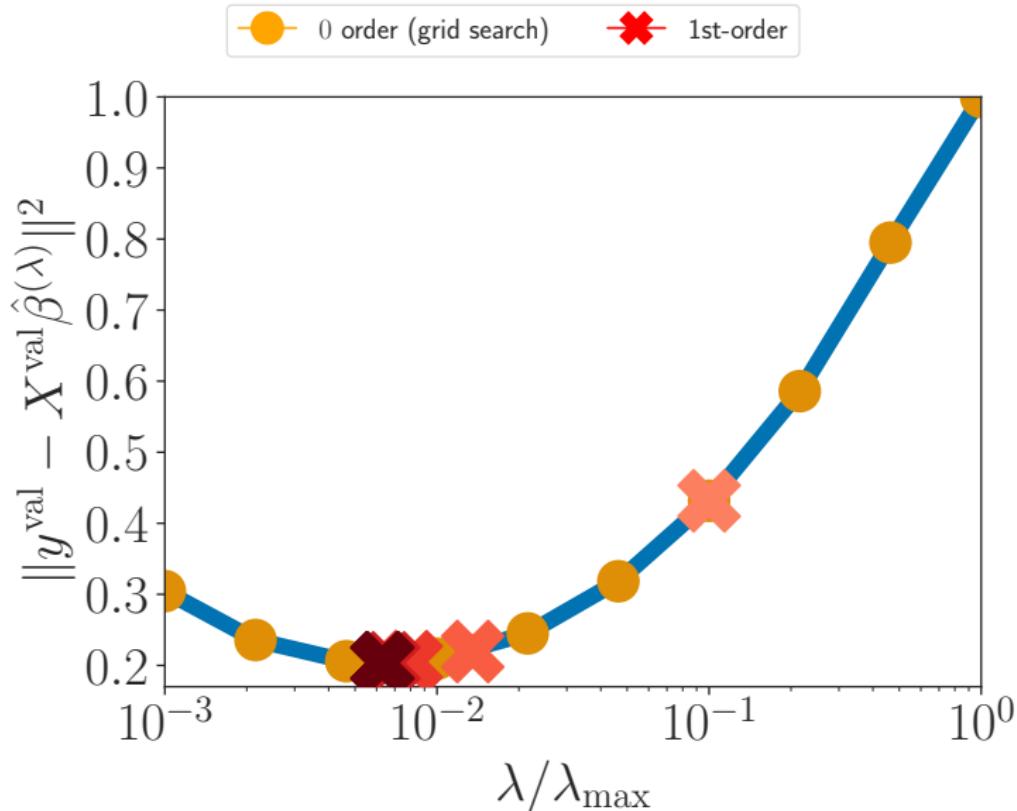
Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ .

First-order optimization in λ , Lasso



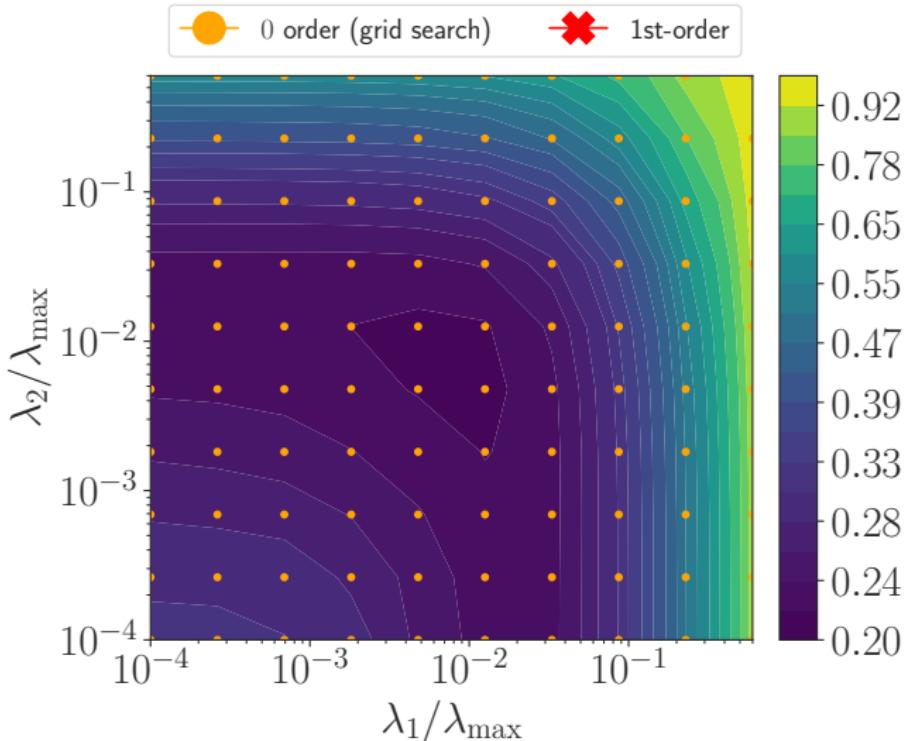
Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ .

First-order optimization in λ , Lasso



Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ .

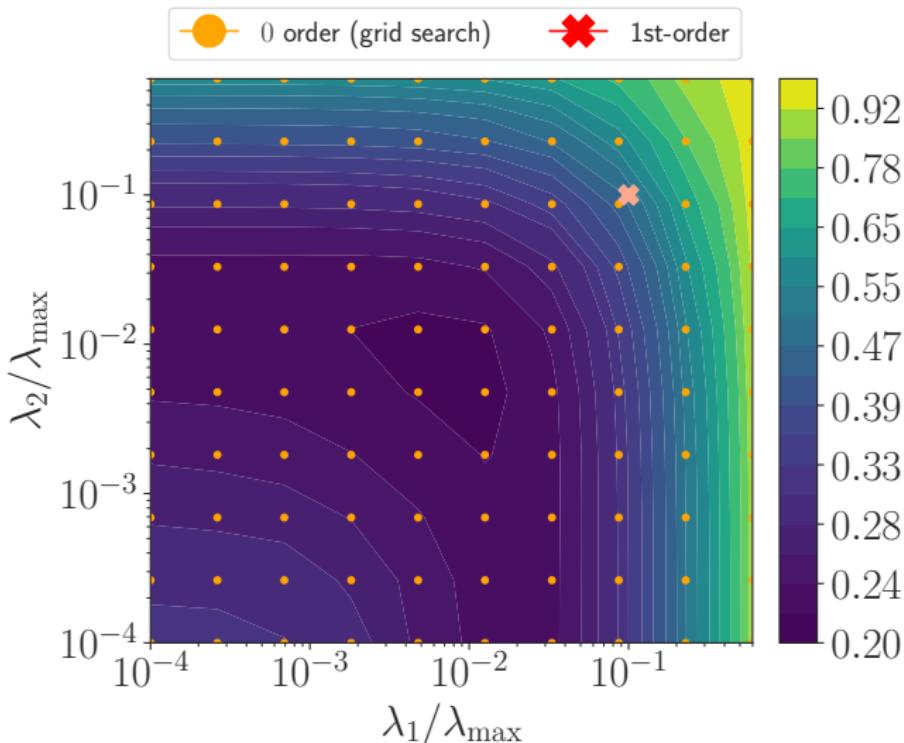
First-order optimization in λ , Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg \min_{\beta} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|^2$$

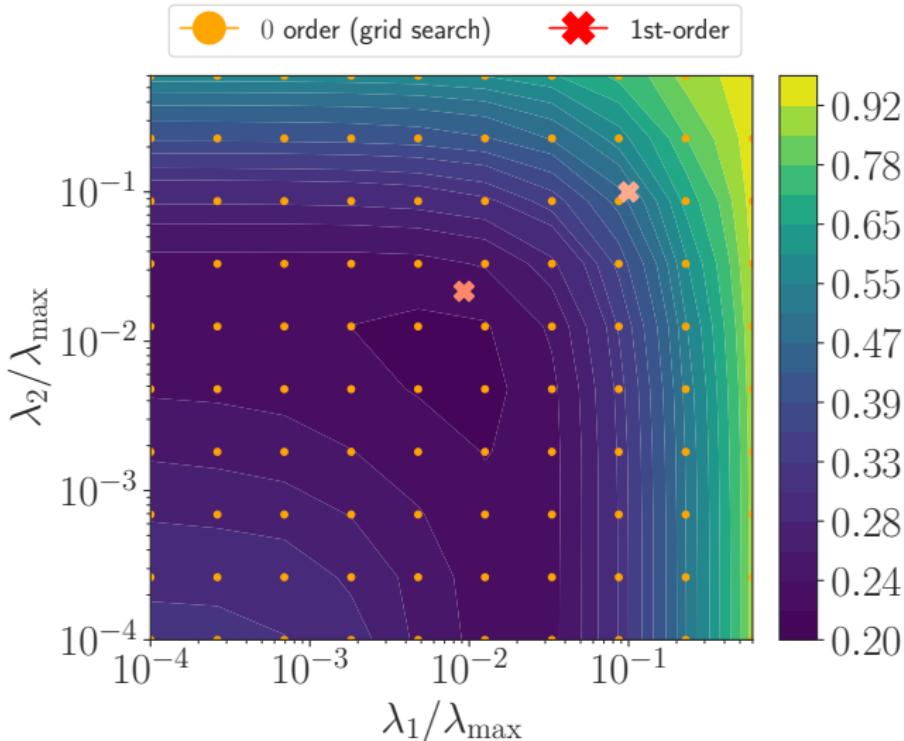
First-order optimization in λ , Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg \min_{\beta} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|^2$$

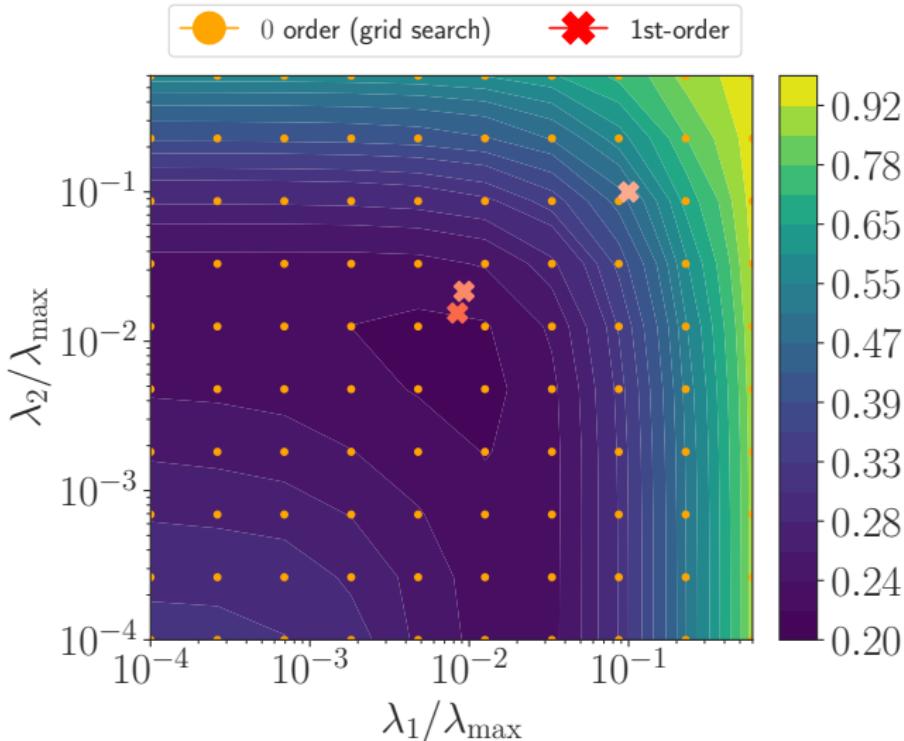
First-order optimization in λ , Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg \min_{\beta} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|^2$$

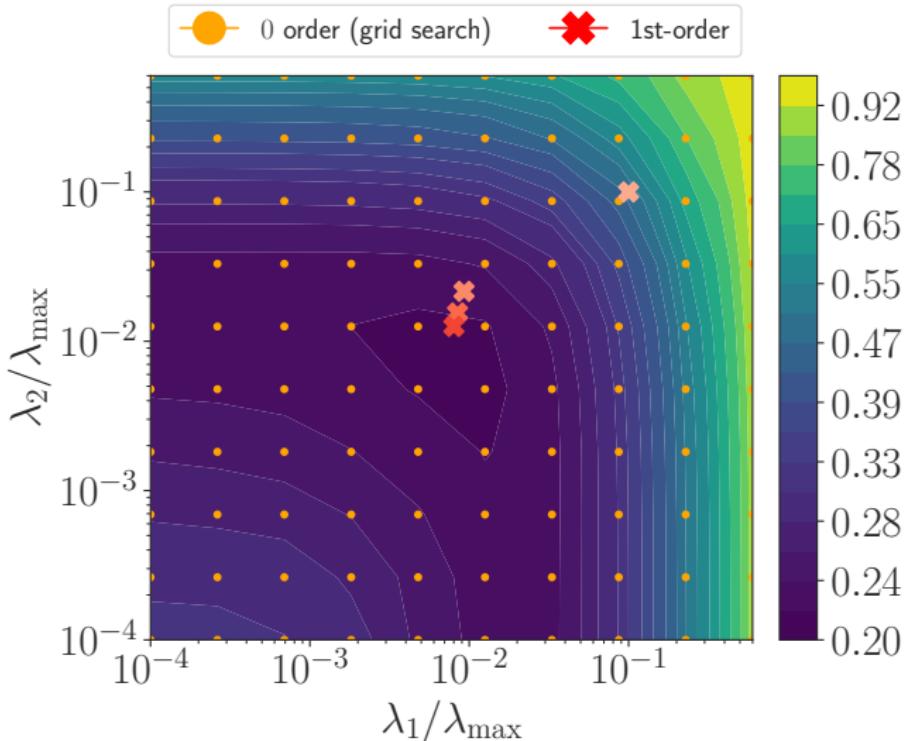
First-order optimization in λ , Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg \min_{\beta} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|^2$$

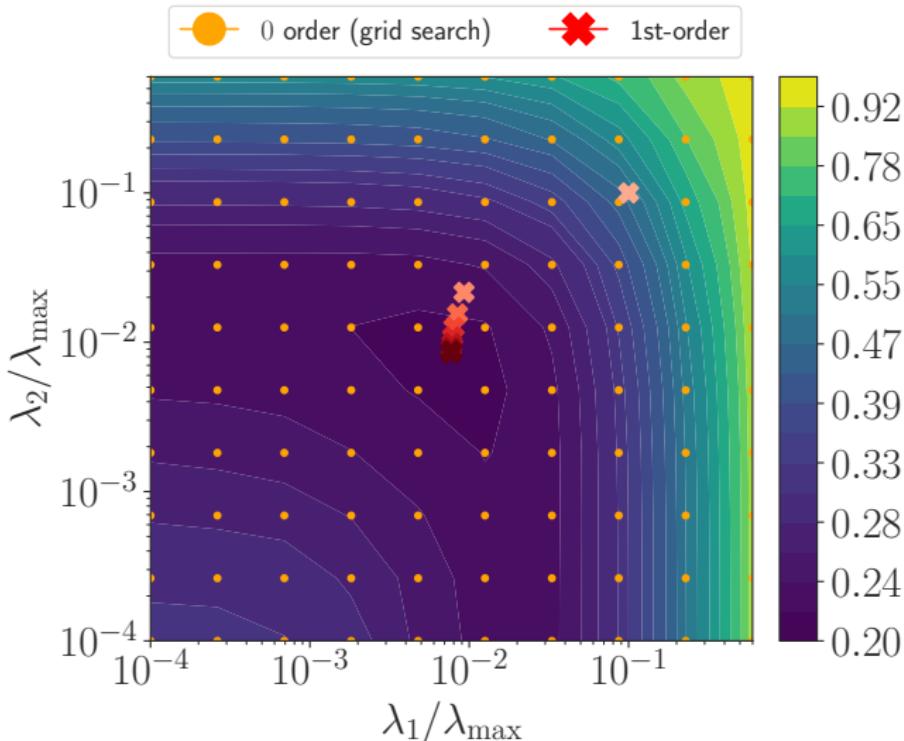
First-order optimization in λ , Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg \min_{\beta} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|^2$$

First-order optimization in λ , Enet



Real-sim dataset, level sets of the validation loss (hold-out)

$$\arg \min_{\beta} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}}\beta\|^2 + \lambda_1 \|\beta\|_1 + \frac{\lambda_2}{2} \|\beta\|^2$$

What's hard? Computing $\nabla_{\lambda} \mathcal{L}(\lambda)$

$$\begin{aligned} & \arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\} \\ & \text{s.t. } \hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1 \end{aligned}$$

Once $\nabla_{\lambda} \mathcal{L}(\lambda)$ is computed, one can use standard first-order methods:

- ▶ Line-search⁽²²⁾
- ▶ L-BFGS⁽²³⁾
- ▶ Gradient descent

⁽²²⁾ J. Nocedal and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006, Chap. 3.

⁽²³⁾ D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* (1989).

What's hard? Computing $\nabla_{\lambda} \mathcal{L}(\lambda)$

$$\begin{aligned} & \arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\} \\ & \text{s.t. } \hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1 \end{aligned}$$

Once $\nabla_{\lambda} \mathcal{L}(\lambda)$ is computed, one can use standard first-order methods:

- ▶ Line-search⁽²²⁾
- ▶ L-BFGS⁽²³⁾
- ▶ Gradient descent

Main challenge today: compute $\nabla_{\lambda} \mathcal{L}(\lambda)$ for a given λ

⁽²²⁾ J. Nocedal and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006, Chap. 3.

⁽²³⁾ D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* (1989).

What's hard? Computing $\nabla_{\lambda} \mathcal{L}(\lambda)$

$$\begin{aligned} & \arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\} \\ & \text{s.t. } \hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1 \end{aligned}$$

Once $\nabla_{\lambda} \mathcal{L}(\lambda)$ is computed, one can use standard first-order methods:

- ▶ Line-search⁽²²⁾
- ▶ L-BFGS⁽²³⁾
- ▶ Gradient descent

Main challenge today: compute $\nabla_{\lambda} \mathcal{L}(\lambda)$ for a given λ

⁽²²⁾ J. Nocedal and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006, Chap. 3.

⁽²³⁾ D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* (1989).

How to compute $\nabla_{\lambda} \mathcal{L}(\lambda)$?

$$\arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1$$

Chain rule:

$$\begin{aligned}\nabla_{\lambda} \mathcal{L}(\lambda) &= \underbrace{\hat{\mathcal{J}}_{(\lambda)}^\top}_{:= (\nabla_{\lambda} \hat{\beta}_1^{(\lambda)}, \dots, \nabla_{\lambda} \hat{\beta}_p^{(\lambda)})} \nabla_{\beta} C(\hat{\beta}^{(\lambda)}) \\ &\rightarrow \text{main challenge}\end{aligned}$$

► Boils down to:

how to compute the Jacobian $\hat{\mathcal{J}}_{(\lambda)} \in \mathbb{R}^{p \times 1}$ efficiently?

How to compute $\nabla_{\lambda} \mathcal{L}(\lambda)$?

$$\arg \min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1$$

Chain rule:

$$\begin{aligned} \nabla_{\lambda} \mathcal{L}(\lambda) &= \underbrace{\hat{\mathcal{J}}_{(\lambda)}^{\top}}_{:= (\nabla_{\lambda} \hat{\beta}_1^{(\lambda)}, \dots, \nabla_{\lambda} \hat{\beta}_p^{(\lambda)})} \nabla_{\beta} C(\hat{\beta}^{(\lambda)}) \\ &\rightarrow \text{main challenge} \end{aligned}$$

- ▶ Boils down to:

how to compute the Jacobian $\hat{\mathcal{J}}_{(\lambda)} \in \mathbb{R}^{p \times 1}$ efficiently?

How to compute $\hat{\mathcal{J}}_{(\lambda)} := (\nabla_{\lambda} \hat{\beta}_1^{(\lambda)}, \dots, \nabla_{\lambda} \hat{\beta}_p^{(\lambda)})^{\top}$?

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \underbrace{\frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \frac{\lambda}{2} \|\beta\|^2}_{\text{inner optimization problem}}$$

Smooth inner optimization problems, **well studied**:

- ▶ *Implicit differentiation* (**closed-form** formula)⁽²⁴⁾⁽²⁵⁾:
need to solve a $p \times p$ linear system ($p = \#\text{features}$)
- ▶ *Automatic differentiation*, *reverse*⁽²⁶⁾ or *forward*⁽²⁷⁾ mode

⁽²⁴⁾ J. Larsen et al. "Design and regularization of neural networks: the optimal use of a validation set". In: *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*. 1996.

⁽²⁵⁾ Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

⁽²⁶⁾ J. Domke. "Generic methods for optimization-based modeling". In: *AISTATS*. vol. 22. 2012.

⁽²⁷⁾ L. Franceschi et al. "Forward and reverse gradient-based hyperparameter optimization". In: *ICML*. 2017.

How to compute $\hat{\mathcal{J}}_{(\lambda)} := (\nabla_{\lambda} \hat{\beta}_1^{(\lambda)}, \dots, \nabla_{\lambda} \hat{\beta}_p^{(\lambda)})^{\top}$?

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \underbrace{\frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

Non-smooth inner optimization problems, **scarcer literature**:

- ▶ Smooth the non-smooth term⁽²⁸⁾
- ▶ Use algorithms with differentiable updates^{(29), (30)} (Bregman)

Our contributions:

- ▶ Iterative differentiation can be applied on proximal algorithms
- ▶ Equivalent of the implicit differentiation in the non-smooth case

⁽²⁸⁾ G. Peyré and J. Fadili. "Learning analysis sparsity priors". In: *Sampta*. 2011.

⁽²⁹⁾ P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

⁽³⁰⁾ J. Frecon, S. Salzo, and M. Pontil. "Bilevel learning of the group lasso structure". In: *NeurIPS*. 2018.

How to compute $\hat{\mathcal{J}}_{(\lambda)} := (\nabla_{\lambda} \hat{\beta}_1^{(\lambda)}, \dots, \nabla_{\lambda} \hat{\beta}_p^{(\lambda)})^{\top}$?

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \underbrace{\frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$

Non-smooth inner optimization problems, **scarcer literature**:

- ▶ Smooth the non-smooth term⁽²⁸⁾
- ▶ Use algorithms with differentiable updates^{(29), (30)} (Bregman)

Our contributions:

- ▶ Iterative differentiation can be applied on proximal algorithms
- ▶ Equivalent of the implicit differentiation in the non-smooth case

⁽²⁸⁾ G. Peyré and J. Fadili. "Learning analysis sparsity priors". In: *Sampta*. 2011.

⁽²⁹⁾ P. Ochs et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.

⁽³⁰⁾ J. Frecon, S. Salzo, and M. Pontil. "Bilevel learning of the group lasso structure". In: *NeurIPS*. 2018.

Forward-mode differentiation^{(31), (32)} of PGD

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \overbrace{f}^{\text{smooth}}(\beta) + \lambda \overbrace{g}^{\text{non-smooth}}(\beta)$$

Algorithm: Proximal gradient descent PGD

init : $\beta = 0_p$, , L

for $\text{iter} = 1, \dots, \text{do}$

$z \leftarrow \beta - \frac{1}{L} \nabla f(\beta)$ *// gradient step*

$\beta \leftarrow \text{prox}_{\lambda g / L}(z)$ *// proximal step*

return β

⁽³¹⁾R. E. Wengert. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

⁽³²⁾C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Forward-mode differentiation^{(31), (32)} of PGD

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \overbrace{f}^{\text{smooth}}(\beta) + \lambda \overbrace{g}^{\text{non-smooth}}(\beta)$$

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p$, $\mathcal{J} = 0_p$, L

for $\text{iter} = 1, \dots$, **do**

$z \leftarrow \beta - \frac{1}{L} \nabla f(\beta)$ *// gradient step*

$dz \leftarrow \left(\text{Id}_p - \frac{1}{L} \nabla^2 f(\beta) \right) \mathcal{J}$ *// diff w.r.t. λ : chain rule*

$\beta \leftarrow \text{prox}_{\lambda g / L}(z)$ *// proximal step*

return β

⁽³¹⁾R. E. Wengert. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

⁽³²⁾C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Forward-mode differentiation^{(31), (32)} of PGD

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \overbrace{f}^{\text{smooth}}(\beta) + \lambda \overbrace{g}^{\text{non-smooth}}(\beta)$$

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p$, $\mathcal{J} = 0_p$, L

for $\text{iter} = 1, \dots$, **do**

$z \leftarrow \beta - \frac{1}{L} \nabla f(\beta)$ // gradient step

$dz \leftarrow \left(\text{Id}_p - \frac{1}{L} \nabla^2 f(\beta) \right) \mathcal{J}$ // diff w.r.t. λ : chain rule

$\beta \leftarrow \text{prox}_{\lambda g/L}(z)$ // proximal step

$\mathcal{J} \leftarrow \partial_z \text{prox}_{\lambda g/L}(z) dz$ // diff w.r.t. λ : chain rule

return β , \mathcal{J}

⁽³¹⁾R. E. Wengert. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

⁽³²⁾C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Forward-mode differentiation^{(31), (32)} of PGD

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \overbrace{f}^{\text{smooth}}(\beta) + \lambda \overbrace{g}^{\text{non-smooth}}(\beta)$$

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p$, $\mathcal{J} = 0_p$, L

for iter = 1, ..., **do**

$z \leftarrow \beta - \frac{1}{L} \nabla f(\beta)$ // gradient step

$dz \leftarrow \left(\text{Id}_p - \frac{1}{L} \nabla^2 f(\beta) \right) \mathcal{J}$ // diff w.r.t. λ : chain rule

$\beta \leftarrow \text{prox}_{\lambda g/L}(z)$ // proximal step

$\mathcal{J} \leftarrow \partial_z \text{prox}_{\lambda g/L}(z) dz + \partial_\lambda \text{prox}_{\lambda g/L}(z)$ // diff w.r.t. λ : chain rule
// do not forget this term!

return β , \mathcal{J}

⁽³¹⁾R. E. Wengert. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

⁽³²⁾C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

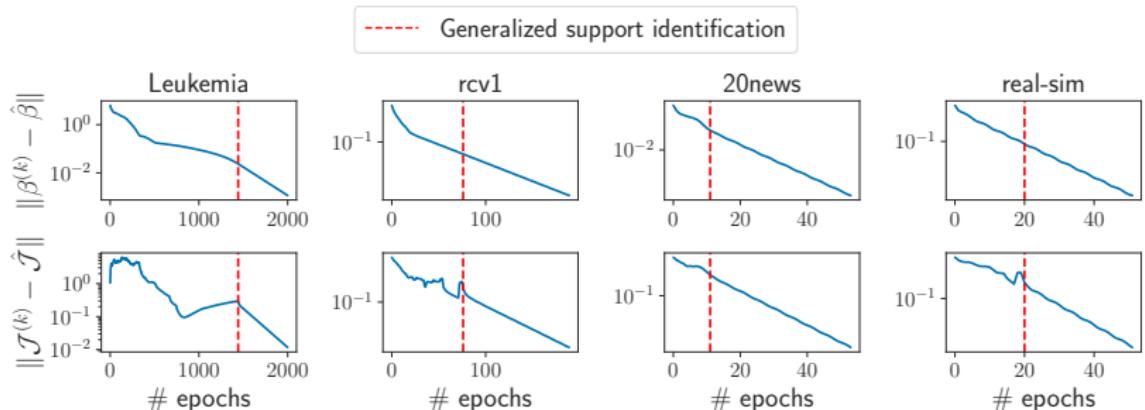
Forward-mode differentiation

Forward diff. PCD convergence, Lasso

Assume

- ▶ The sequence $(\beta^{(k)})$ generated by PCD converges to $\hat{\beta}$
- ▶ The problem is not degenerated: $-X^\top(X\hat{\beta} - y) \in \text{ri}(\lambda\partial\|\cdot\|_1)$
- ▶ Restricted injectivity holds: $X_{:,S}^\top X_{:,S} \succ 0$

Then the Jacobian sequence based on forward diff. of PCD converges to the true Jacobian. Once the support (the non-zeros coefs.) has been identified, convergence is linear.⁽³³⁾



(33) Q. Bertrand et al. "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: ICML (2020).

Implicit differentiation (smooth ψ)⁽³⁴⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \psi(\beta, \lambda)$$

$$\nabla_{\beta} \psi\left(\hat{\beta}^{(\lambda)}, \lambda\right) = 0$$

⁽³⁴⁾Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

Implicit differentiation (smooth ψ)⁽³⁴⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \psi(\beta, \lambda)$$

$$\nabla_{\beta} \psi\left(\hat{\beta}^{(\lambda)}, \lambda\right) = 0$$

$$\nabla_{\beta, \lambda}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda) + \hat{\mathcal{J}}_{(\lambda)}^\top \nabla_{\beta}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda) = 0$$

⁽³⁴⁾Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

Implicit differentiation (smooth ψ)⁽³⁴⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \psi(\beta, \lambda)$$

$$\nabla_{\beta} \psi(\hat{\beta}^{(\lambda)}, \lambda) = 0$$

$$\nabla_{\beta, \lambda}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda) + \hat{\mathcal{J}}_{(\lambda)}^\top \nabla_{\beta}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda) = 0$$

$$\hat{\mathcal{J}}_{(\lambda)}^\top = -\nabla_{\beta, \lambda}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda) \underbrace{\left(\nabla_{\beta}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda) \right)^{-1}}_{p \times p}$$

- ▶ Need to solve a linear **system of size p**

⁽³⁴⁾Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

Implicit differentiation (smooth ψ)⁽³⁴⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} \psi(\beta, \lambda)$$

$$\nabla_{\beta} \psi\left(\hat{\beta}^{(\lambda)}, \lambda\right) = 0$$

$$\nabla_{\beta, \lambda}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda) + \hat{\mathcal{J}}_{(\lambda)}^\top \nabla_{\beta}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda) = 0$$

$$\hat{\mathcal{J}}_{(\lambda)}^\top = -\nabla_{\beta, \lambda}^2 \psi\left(\hat{\beta}^{(\lambda)}, \lambda\right) \underbrace{\left(\nabla_{\beta}^2 \psi(\hat{\beta}^{(\lambda)}, \lambda)\right)^{-1}}_{p \times p}$$

- ▶ Need to solve a linear **system of size p**

⁽³⁴⁾Y. Bengio. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).

Implicit differentiation ($f + \lambda \sum_j |\beta_j|$)⁽³⁵⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)$$

⁽³⁵⁾ Q. Bertrand et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *JMLR* (2022).

Implicit differentiation ($f + \lambda \sum_j |\beta_j|$)⁽³⁵⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)$$

$$\begin{aligned}\hat{\mathcal{J}} &= \partial_{\beta} \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) \left(\text{Id} - \frac{\nabla^2 f}{L} \right) \hat{\mathcal{J}} \\ &\quad + \partial_{\lambda} \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)\end{aligned}$$

⁽³⁵⁾ Q. Bertrand et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *JMLR* (2022).

Implicit differentiation ($f + \lambda \sum_j |\beta_j|$)⁽³⁵⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)$$

$$\begin{aligned} \hat{\mathcal{J}} &= \partial_{\beta} \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) \left(\text{Id} - \frac{\nabla^2 f}{L} \right) \hat{\mathcal{J}} \\ &\quad + \partial_{\lambda} \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) \end{aligned}$$

Key observation, if $\hat{\beta}_j^{(\lambda)} = 0$ + non-degeneracy assumption:

$$\partial_{\beta} \text{ST} \left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L} \nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) = 0 = \partial_{\lambda} \text{ST} \left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L} \nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)$$

⁽³⁵⁾ Q. Bertrand et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *JMLR* (2022).

Implicit differentiation ($f + \lambda \sum_j |\beta_j|$)⁽³⁵⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)$$

$$\begin{aligned} \hat{\mathcal{J}} &= \partial_{\beta} \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) \left(\text{Id} - \frac{\nabla^2 f}{L} \right) \hat{\mathcal{J}} \\ &\quad + \partial_{\lambda} \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) \end{aligned}$$

Key observation, if $\hat{\beta}_j^{(\lambda)} = 0$ + non-degeneracy assumption:

$$\partial_{\beta} \text{ST} \left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L} \nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) = 0 = \partial_{\lambda} \text{ST} \left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L} \nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)$$

With $\mathcal{S} = \{j \in [p] : \hat{\beta}_j^{(\lambda)} = 0\}$ we have $\hat{\mathcal{J}}_{\mathcal{S}^c} = 0$

$$\hat{\mathcal{J}}_{\mathcal{S}} = \partial_{\beta} \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) s \hat{\mathcal{J}}_{\mathcal{S}} + \partial_{\lambda} \text{ST} \left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L} \nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) s$$

⁽³⁵⁾ Q. Bertrand et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *JMLR* (2022).

Implicit differentiation ($f + \lambda \sum_j |\beta_j|$)⁽³⁵⁾

$$\hat{\beta}^{(\lambda)} \in \arg \min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$$

$$\hat{\beta}^{(\lambda)} = \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)$$

$$\begin{aligned} \hat{\mathcal{J}} &= \partial_\beta \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) \left(\text{Id} - \frac{\nabla^2 f}{L} \right) \hat{\mathcal{J}} \\ &\quad + \partial_\lambda \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) \end{aligned}$$

Key observation, if $\hat{\beta}_j^{(\lambda)} = 0$ + non-degeneracy assumption:

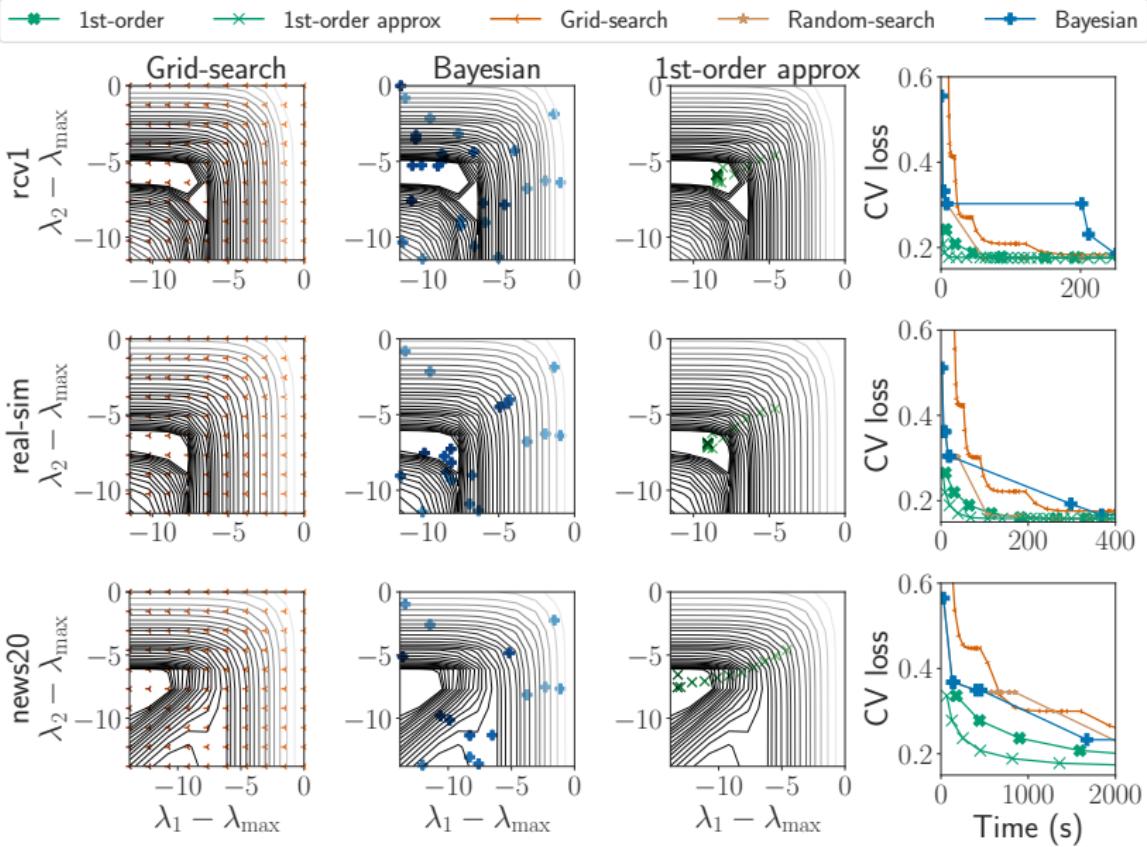
$$\partial_\beta \text{ST} \left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L} \nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right) = 0 = \partial_\lambda \text{ST} \left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L} \nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)$$

With $\mathcal{S} = \{j \in [p] : \hat{\beta}_j^{(\lambda)} = 0\}$ we have $\hat{\mathcal{J}}_{\mathcal{S}^c} = 0$

$$\hat{\mathcal{J}}_{\mathcal{S}} = \partial_\beta \text{ST} \left(\hat{\beta}^{(\lambda)} - \frac{1}{L} \nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)_{\mathcal{S}} \hat{\mathcal{J}}_{\mathcal{S}} + \partial_\lambda \text{ST} \left(\hat{\beta}_j^{(\lambda)} - \frac{1}{L} \nabla_j f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L} \right)_{\mathcal{S}}$$

⁽³⁵⁾ Q. Bertrand et al. "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: JMLR (2022).

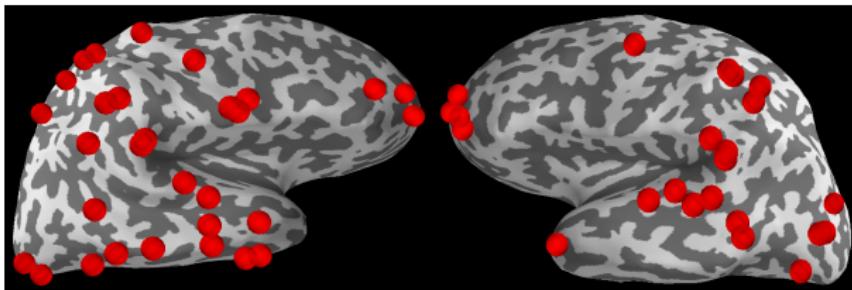
Experiments - Enet cross-validation



$$\arg \min_{\beta} \frac{1}{2n} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + e^{\lambda_1} \|\beta\|_1 + \frac{e^{\lambda_2}}{2} \|\beta\|^2$$

Experiments - Real MEEG data

- ▶ **Outer criterion:** FDMC SURE⁽³⁶⁾
- ▶ **Inner problems:** vanilla Lasso



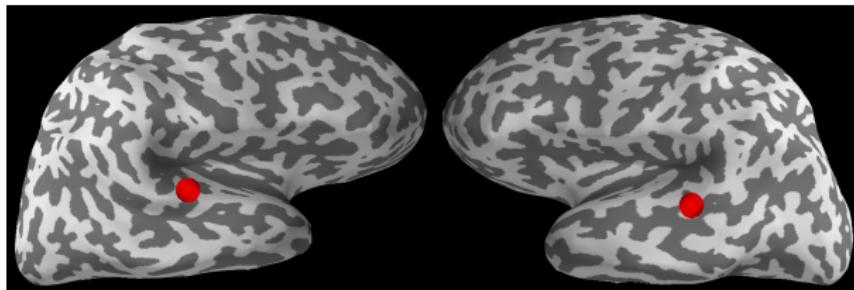
Real M/EEG data, vanilla Lasso (1 hyperparameter λ)

$$\arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\beta\|_2^2 + e^{\lambda} \|\beta\|_1$$

⁽³⁶⁾ C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Experiments - Real MEEG data

- ▶ **Outer criterion:** FDMC SURE⁽³⁶⁾
- ▶ **Inner problems:** weighted Lasso ($\sim 10^4$ hyperparameters)



Real M/EEG data, weighted Lasso (p hyperparameters)

$$\arg \min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\beta\|_2^2 + \sum_{j=1}^p e^{\lambda_j} |\beta_j|$$

⁽³⁶⁾ C.-A. Deledalle et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).

Summary

- ▶ Paper **Q. Bertrand et al.** “Implicit differentiation for fast hyperparameter selection in non-smooth convex learning”. In: *JMLR* (2022)
- ▶ Open source code <https://github.com/QB3/sparse-ho> (in jax <https://github.com/google/jaxopt/pull/274>)

Other frameworks:

- ▶ J. Bolte et al. “Nonsmooth implicit differentiation for machine-learning and optimization”. In: *NeurIPS* (2021)
- ▶ J. Bolte, E. Pauwels, and S. Vaiter. “Automatic differentiation of nonsmooth iterative algorithms”. In: *arXiv preprint arXiv:2206.00457* (2022)
- ▶ S. Mehmood and P. Ochs. “Fixed-Point Automatic Differentiation of Forward–Backward Splitting Algorithms for Partly Smooth Functions”. In: *arXiv preprint arXiv:2208.03107* (2022)

Summary

- ▶ Paper **Q. Bertrand et al.** “Implicit differentiation for fast hyperparameter selection in non-smooth convex learning”. In: *JMLR* (2022)
- ▶ Open source code <https://github.com/QB3/sparse-ho> (in jax <https://github.com/google/jaxopt/pull/274>)

Other frameworks:

- ▶ **J. Bolte et al.** “Nonsmooth implicit differentiation for machine-learning and optimization”. In: *NeurIPS* (2021)
- ▶ **J. Bolte, E. Pauwels, and S. Vaiter.** “Automatic differentiation of nonsmooth iterative algorithms”. In: *arXiv preprint arXiv:2206.00457* (2022)
- ▶ **S. Mehmood and P. Ochs.** “Fixed-Point Automatic Differentiation of Forward–Backward Splitting Algorithms for Partly Smooth Functions”. In: *arXiv preprint arXiv:2208.03107* (2022)

Perspectives I: Machine Learning

Opportunities for statisticians and the bilevel community

- ▶ Can have access to much more complex estimators
- ▶ With a very large number of hyperparameters

Bilevel optimization goes much beyond hyperparameter selection

- ▶ Meta-learning^{(37), (38)}
- ▶ Dataset distillation⁽³⁹⁾
- ▶ Deep equilibrium networks^{(40), (41)}

⁽³⁷⁾ L. Franceschi et al. "Forward and reverse gradient-based hyperparameter optimization". In: *ICML*. 2017.

⁽³⁸⁾ L. Franceschi et al. "Bilevel Programming for Hyperparameter Optimization and Meta-Learning". In: *ICML*. 2018.

⁽³⁹⁾ J. Lorraine, P. Vicol, and D. Duvenaud. "Optimizing Millions of Hyperparameters by Implicit Differentiation". In: *arXiv preprint arXiv:1911.02590* (2019).

⁽⁴⁰⁾ S. Bai, J. Z. Kolter, and V. Koltun. "Deep equilibrium models". In: *NeurIPS* (2019).

⁽⁴¹⁾ S. Bai, V. Koltun, and J. Z. Kolter. "Multiscale deep equilibrium models". In: *NeurIPS* (2020).

Perspectives I: Machine Learning

Opportunities for statisticians and the bilevel community

- ▶ Can have access to much more complex estimators
- ▶ With a very large number of hyperparameters

Bilevel optimization goes much beyond hyperparameter selection

- ▶ Meta-learning^{(37), (38)}
- ▶ Dataset distillation⁽³⁹⁾
- ▶ Deep equilibrium networks^{(40), (41)}

⁽³⁷⁾ L. Franceschi et al. "Forward and reverse gradient-based hyperparameter optimization". In: *ICML*. 2017.

⁽³⁸⁾ L. Franceschi et al. "Bilevel Programming for Hyperparameter Optimization and Meta-Learning". In: *ICML*. 2018.

⁽³⁹⁾ J. Lorraine, P. Vicol, and D. Duvenaud. "Optimizing Millions of Hyperparameters by Implicit Differentiation". In: *arXiv preprint arXiv:1911.02590* (2019).

⁽⁴⁰⁾ S. Bai, J. Z. Kolter, and V. Koltun. "Deep equilibrium models". In: *NeurIPS* (2019).

⁽⁴¹⁾ S. Bai, V. Koltun, and J. Z. Kolter. "Multiscale deep equilibrium models". In: *NeurIPS* (2020).

Perspectives II: Optimization

- ▶ For smooth inner problems, HO packages exist⁽⁴²⁾⁽⁴³⁾
- ▶ But practitioners mostly rely on 0-order methods⁽⁴⁴⁾⁽⁴⁵⁾

Algorithmic problems

- ▶ Hard to tune *hyperhyperparameters*
- ▶ Hard to calibrate nested *for* loops

⁽⁴²⁾F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

⁽⁴³⁾L. Franceschi et al. "Far-HO: A Bilevel Programming Package for Hyperparameter Optimization and Meta-Learning". In: *arXiv preprint arXiv:1806.04941* (2018).

⁽⁴⁴⁾L. Li et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: *Journal of Machine Learning Research* (2017).

⁽⁴⁵⁾T. Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019.

Perspectives II: Optimization

- ▶ For smooth inner problems, HO packages exist⁽⁴²⁾⁽⁴³⁾
- ▶ But practitioners mostly rely on 0-order methods⁽⁴⁴⁾⁽⁴⁵⁾

Algorithmic problems

- ▶ Hard to tune *hyperhyperparameters*
- ▶ Hard to calibrate nested *for* loops

Next challenges

- ▶ Automatic single-loop algorithms
- ▶ Single-loop algorithms for non-smooth inner problems

⁽⁴²⁾ F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

⁽⁴³⁾ L. Franceschi et al. "Far-HO: A Bilevel Programming Package for Hyperparameter Optimization and Meta-Learning". In: *arXiv preprint arXiv:1806.04941* (2018).

⁽⁴⁴⁾ L. Li et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: *Journal of Machine Learning Research* (2017).

⁽⁴⁵⁾ T. Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019.

Perspectives II: Optimization

- ▶ For smooth inner problems, HO packages exist⁽⁴²⁾⁽⁴³⁾
- ▶ But practitioners mostly rely on 0-order methods⁽⁴⁴⁾⁽⁴⁵⁾

Algorithmic problems

- ▶ Hard to tune *hyperhyperparameters*
- ▶ Hard to calibrate nested *for* loops

Next challenges

- ▶ Automatic single-loop algorithms
- ▶ Single-loop algorithms for non-smooth inner problems

⁽⁴²⁾ F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.

⁽⁴³⁾ L. Franceschi et al. "Far-HO: A Bilevel Programming Package for Hyperparameter Optimization and Meta-Learning". In: *arXiv preprint arXiv:1806.04941* (2018).

⁽⁴⁴⁾ L. Li et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: *Journal of Machine Learning Research* (2017).

⁽⁴⁵⁾ T. Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019.

Thank you!

Alexandre, Joseph, Samuel, Mathieu, Mathurin, Quentin K. and Pierre-Antoine



- ▶ Akiba, T. et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019.
- ▶ Bai, S., J. Z. Kolter, and V. Koltun. "Deep equilibrium models". In: *NeurIPS* (2019).
- ▶ Bai, S., V. Koltun, and J. Z. Kolter. "Multiscale deep equilibrium models". In: *NeurIPS* (2020).
- ▶ Bengio, Y. "Gradient-based optimization of hyperparameters". In: *Neural computation* (2000).
- ▶ Berger, H. "Über das elektroenkephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* (1929).
- ▶ Bergstra, J. and Y. Bengio. "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* (2012).
- ▶ Bertrand, Q. et al. "Beyond L1: Faster and Better Sparse Models with skglm". In: *arXiv preprint arXiv:2204.07826* (2022).

- ▶ **Bertrand, Q.** and M. Massias. “Anderson acceleration of coordinate descent”. In: *AISTATS*. 2021.
- ▶ **Bertrand, Q.** et al. “Implicit differentiation for fast hyperparameter selection in non-smooth convex learning”. In: *JMLR* (2022).
- ▶ **Bertrand, Q.** et al. “Implicit differentiation of Lasso-type models for hyperparameter optimization”. In: *ICML* (2020).
- ▶ Bolte, J., E. Pauwels, and S. Vaiter. “Automatic differentiation of nonsmooth iterative algorithms”. In: *arXiv preprint arXiv:2206.00457* (2022).
- ▶ Bolte, J. et al. “Nonsmooth implicit differentiation for machine-learning and optimization”. In: *NeurIPS* (2021).
- ▶ Brochu, E., V. M. Cora, and N. De Freitas. “A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning”. In: *arXiv preprint arXiv:1012.2599* (2010).
- ▶ Cohen, D. “Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents”. In: *Science* (1968).

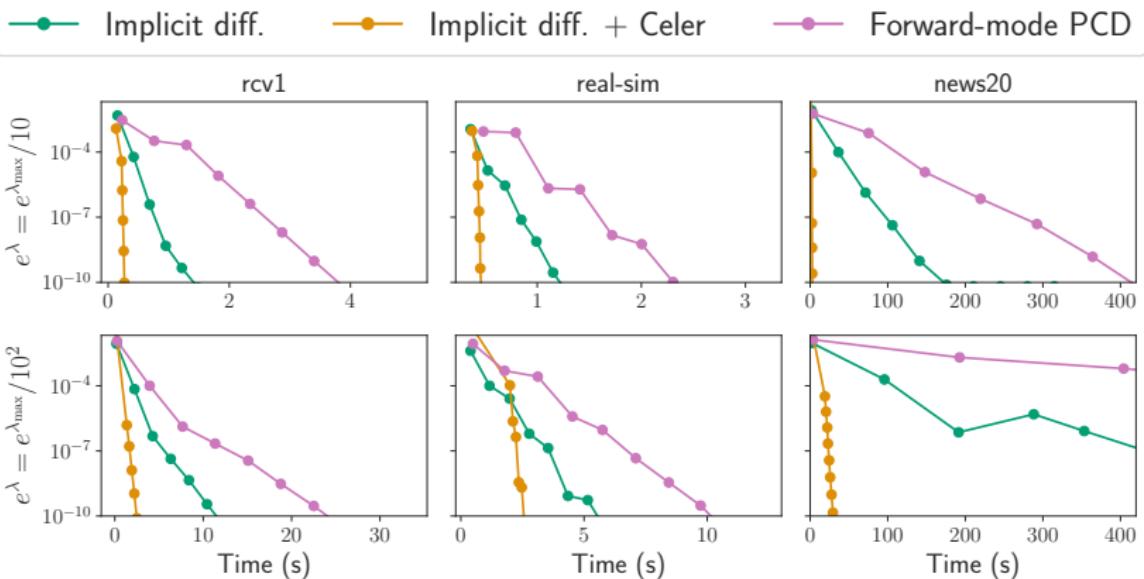
- Deledalle, C.-A. et al. "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* (2014).
- Domke, J. "Generic methods for optimization-based modeling". In: *AISTATS*. Vol. 22. 2012.
- Figueiredo, M. "Adaptive Sparseness Using Jeffreys Prior.". In: *NeurIPS*. 2001.
- Franceschi, L. et al. "Bilevel Programming for Hyperparameter Optimization and Meta-Learning". In: *ICML*. 2018.
- Franceschi, L. et al. "Far-HO: A Bilevel Programming Package for Hyperparameter Optimization and Meta-Learning". In: *arXiv preprint arXiv:1806.04941* (2018).
- Franceschi, L. et al. "Forward and reverse gradient-based hyperparameter optimization". In: *ICML*. 2017.
- Frecon, J., S. Salzo, and M. Pontil. "Bilevel learning of the group lasso structure". In: *NeurIPS*. 2018.
- Hutter, F., J. Lücke, and L. Schmidt-Thieme. "Beyond manual tuning of hyperparameters". In: *KI-Künstliche Intelligenz* (2015).

- ▶ Kohavi, R. and G. H. John. "Automatic parameter selection by minimizing estimated error". In: *Machine Learning Proceedings*. 1995.
- ▶ Larsen, J. et al. "Design and regularization of neural networks: the optimal use of a validation set". In: *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*. 1996.
- ▶ Li, L. et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: *Journal of Machine Learning Research* (2017).
- ▶ Liu, D. C. and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* (1989).
- ▶ Liu, W., Y. Yang, et al. "Parametric or nonparametric? A parametricness index for model selection". In: *Ann. Statist.* 39.4 (2011), pp. 2074–2102.

- ▶ Lorraine, J., P. Vicol, and D. Duvenaud. "Optimizing Millions of Hyperparameters by Implicit Differentiation". In: *arXiv preprint arXiv:1911.02590* (2019).
- ▶ Lounici, K. "Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators". In: *Electron. J. Stat.* (2008).
- ▶ Lounici, K., K. Meziani, and B. Riu. "Muddling Labels for Regularization, a novel approach to generalization". In: *arXiv preprint arXiv:2102.08769* (2021).
- ▶ Lounici, K. et al. "Taking Advantage of Sparsity in Multi-Task Learning". In: *arXiv preprint arXiv:0903.1468* (2009).
- ▶ Mehmood, S. and P. Ochs. "Fixed-Point Automatic Differentiation of Forward–Backward Splitting Algorithms for Partly Smooth Functions". In: *arXiv preprint arXiv:2208.03107* (2022).
- ▶ Nocedal, J. and S. J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006.

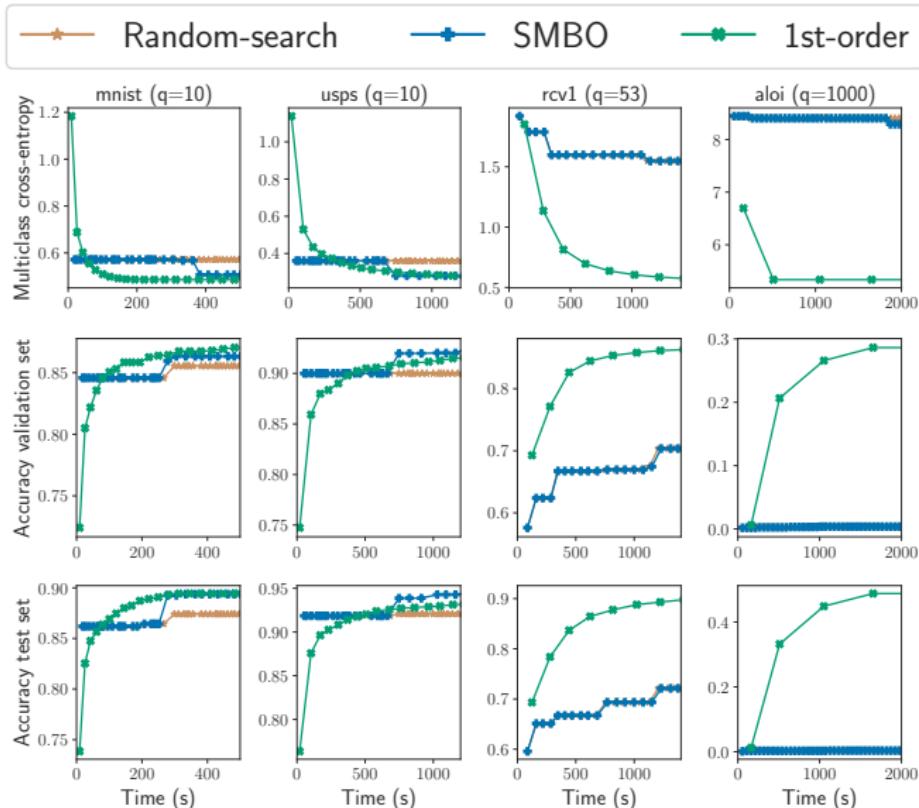
- ▶ Ochs, P. et al. "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. 2015.
- ▶ Pedregosa, F. "Hyperparameter optimization with approximate gradient". In: *ICML*. 2016.
- ▶ Peyré, G. and J. Fadili. "Learning analysis sparsity priors". In: *Sampta*. 2011.
- ▶ Stein, C. M. "Estimation of the mean of a multivariate normal distribution". In: *Ann. Statist.* 9.6 (1981), pp. 1135–1151.
- ▶ Stone, L. R. A. and J.C. Ramer. "Estimating WAIS IQ from Shipley Scale scores: Another cross-validation". In: *Journal of clinical psychology* 21.3 (1965), pp. 297–297.
- ▶ Tipping, M. E. "Sparse Bayesian learning and the relevance vector machine". In: *Journal of Machine Learning Research* (2001).
- ▶ Wengert, R. E. "A simple automatic derivative evaluation program". In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

Backup - Implicit vs forward-mode



Lasso with hold-out criterion: absolute difference between the exact hypergradient (using $\hat{\beta}$) and the iterate hypergradient (using $\beta^{(k)}$) of the Lasso as a function of time.

Backup - Multiclass logistic regression



Multiclass sparse logistic regression hold-out, time comparison (# classes = # hyperparameters).

Backup - Outer procedure

Algorithm: OUTER PROCEDURE

input : $\lambda \in \mathbb{R}^r, (\epsilon_i)$

init : use_adaptive_step_size = True

for $i = 1, \dots, \text{iter}$ **do**

$\lambda^{\text{old}} \leftarrow \lambda$

 // compute the value and the gradient

$\mathcal{L}(\lambda), \nabla \mathcal{L}(\lambda) \leftarrow \text{Implicit diff}(X, y, \lambda, \epsilon_i)$

if use_adaptive_step_size **then**

$\alpha = 1/\|\nabla \mathcal{L}(\lambda)\|$

 // gradient step

$\lambda -= \alpha \nabla \mathcal{L}(\lambda)$

if $\mathcal{L}(\lambda) > \mathcal{L}(\lambda^{\text{old}})$ **then**

 use_adaptive_step_size = False

$\alpha /= 10$

return λ
