Hyperparameter selection for high dimensional sparse learning: application to neuroimaging

A gentle motivation

Joseph Salmon
IMAG, Univ Montpellier, CNRS
Institut Universitaire de France (IUF)

UNIVERSITÉ DE MONTPELLIER

Joint works with various colleagues

Quentin Bertrand (MILA)

Quentin Klopfenstein (Université du Luxembourg)

Mathurin Massias (INRIA, OCKHAM them)

Pierre-Antoine Bannier (M2 student, Parietal Team)

Samuel Vaiter (Université Côte d'Azur, CNRS)

Mathieu Blondel (Google Research, Brain team)

Alexandre Gramfort (INRIA, Parietal Team)

Quentin B.

Quentin K.

Mathurin

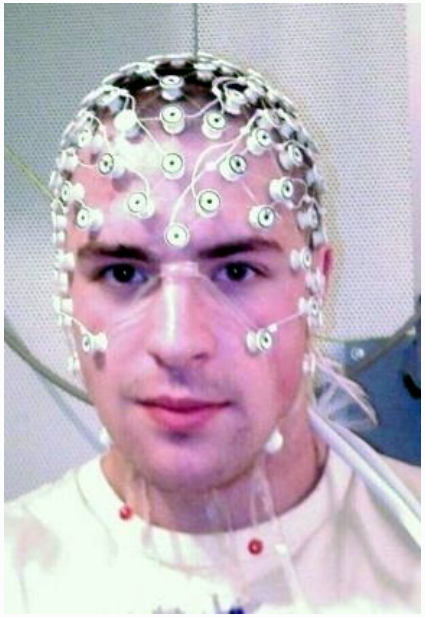
Pierre-Antoine

Mathieu

Samuel

Alexandre

Neuroimaging data: EEG [1] and MEG [2]



(a) EEG

(b) MEG=Mag.+Grad.

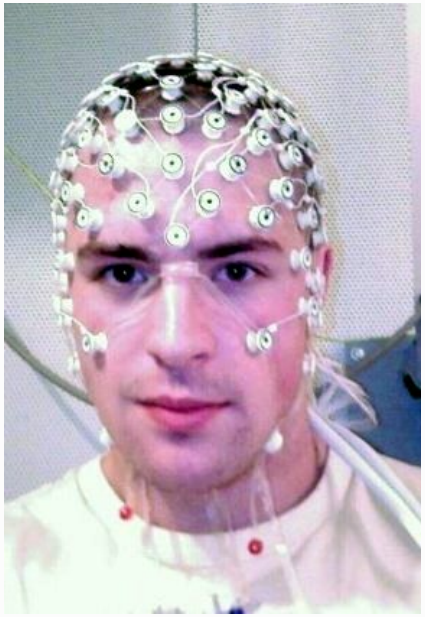
(c) M/EEG

Photo credit: S. Whitmarsh

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

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

Neuroimaging data: EEG [1] and MEG [2]



(a) EEG

(b) MEG=Mag.+Grad.

(c) M/EEG

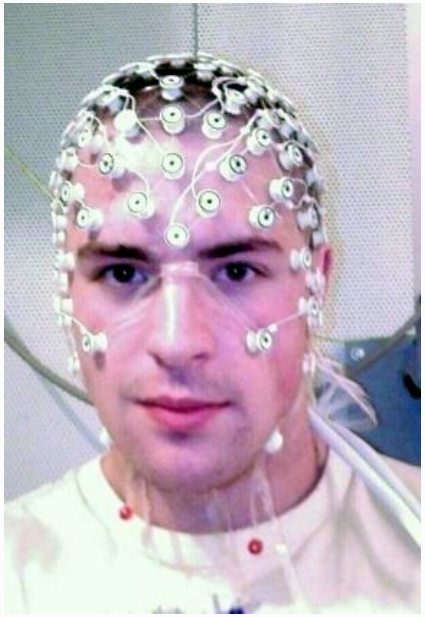
Photo credit: S. Whitmarsh

- **Data** Y: electric and magnetic fields at the head surface

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

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

Neuroimaging data: EEG [1] and MEG [2]



(a) EEG

(b) MEG=Mag.+Grad.

(c) M/EEG

Photo credit: S. Whitmarsh

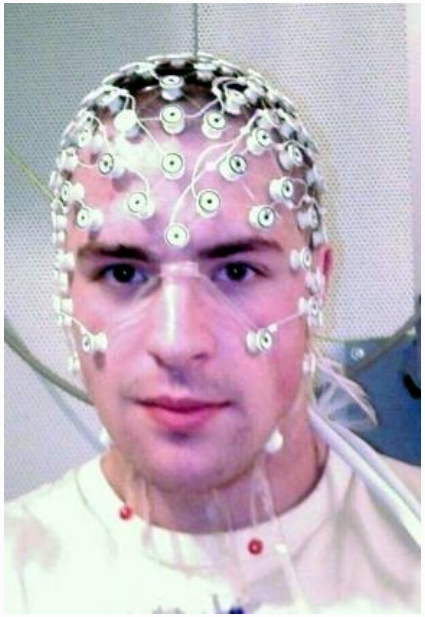
- **Data** Y: electric and magnetic fields at the head surface
- **Goal**: which parts of the brain are responsible for the signals?

---

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

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

Neuroimaging data: EEG [1] and MEG [2]



(a) EEG

(b) MEG=Mag.+Grad.

(c) M/EEG

Photo credit: S. Whitmarsh

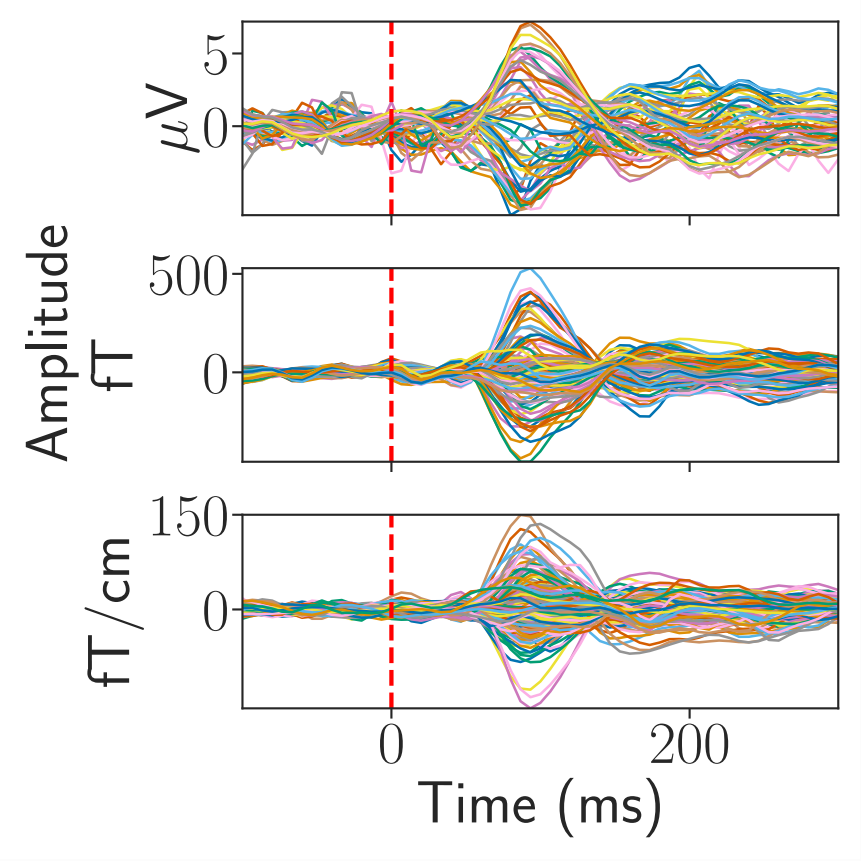
- **Data** Y: electric and magnetic fields at the head surface
- **Goal**: which parts of the brain are responsible for the signals?
- **Applications**: clinical and cognitive experiments

---

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

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

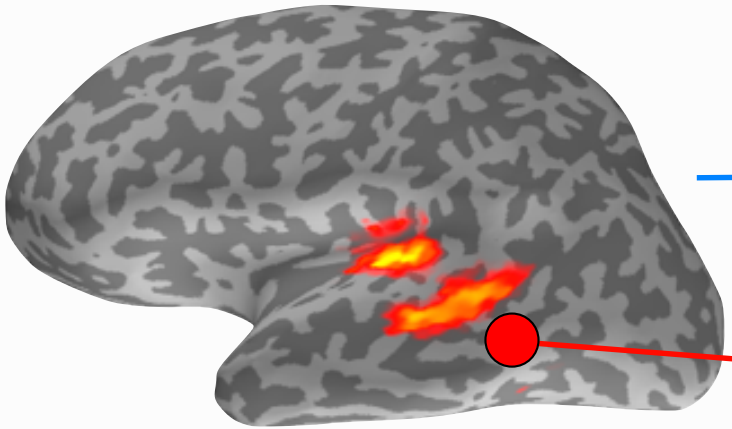
M/EEG DATA: $Y \in \mathbb{R}^{n \times T}$

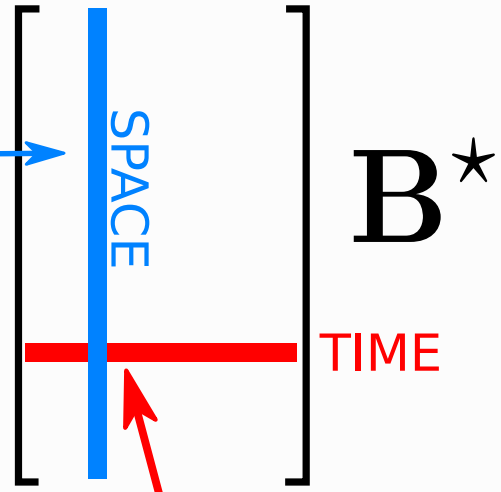


3 modalities:

- EEG
- MEG: magnometers (amplitude)
- MEG: gradiometers (gradients)

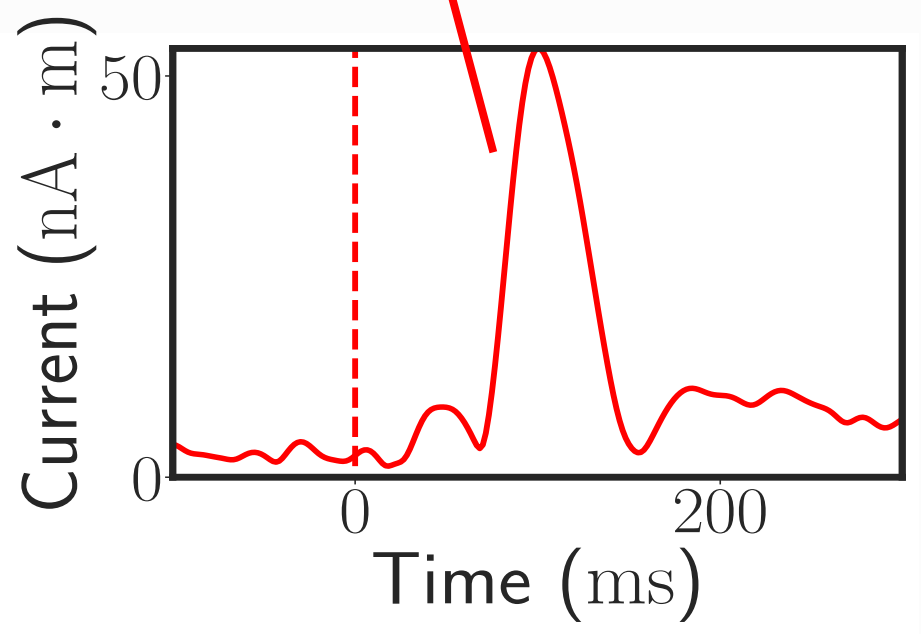
Source modeling



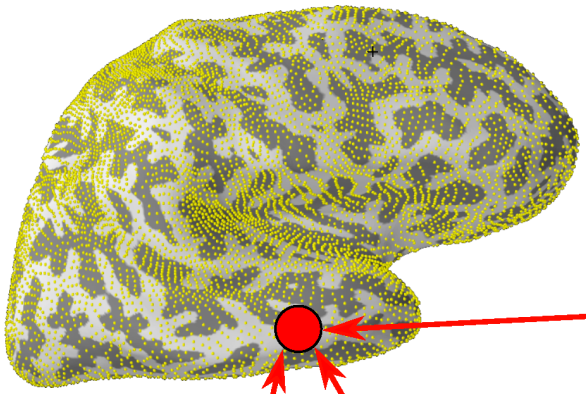


Source candidates regularly spaced in the brain (e.g., every 5mm)

$$\mathbf{B}^\star \in \mathbb{R}^{p \times T}$$



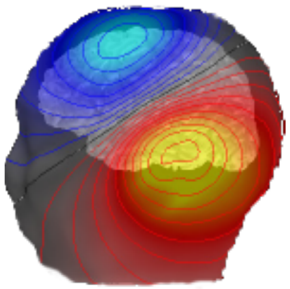
Design matrix - Forward operator

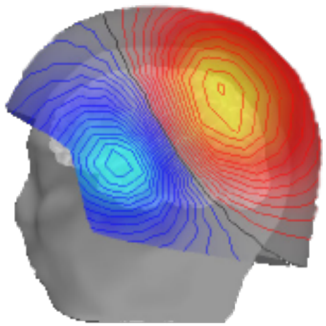


$$X = \left[\begin{array}{c|c} & X_{\text{EEG}} \\ \hline & X_{\text{MEG}} \end{array}\right] \in \mathbb{R}^{n \times p}$$

EEG:
Forward field of the electrodes

MEG:
Forward field of sensor

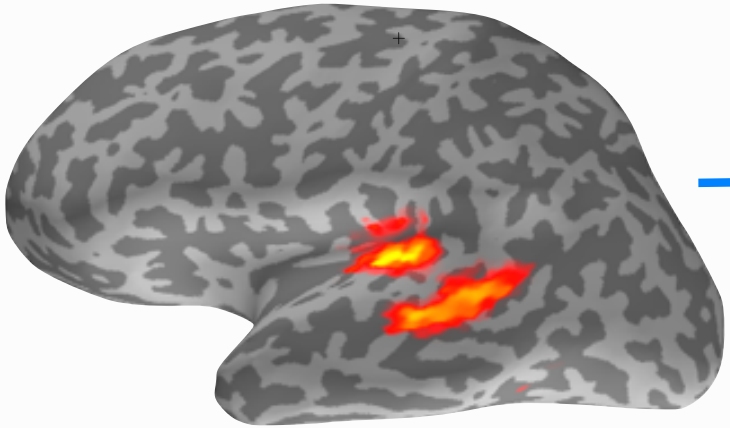




X: gain matrix / forward operator obtained by Maxwell's equations

The M/EEG inverse problem

Maxwell equations and (approx.) linearity



$$\mathbf{B}^\star$$

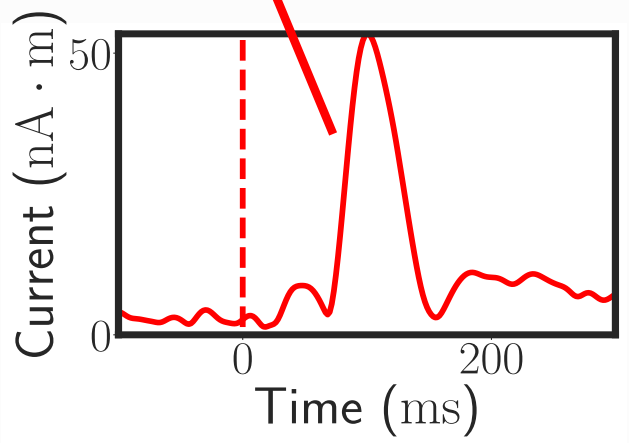
$$+ E$$

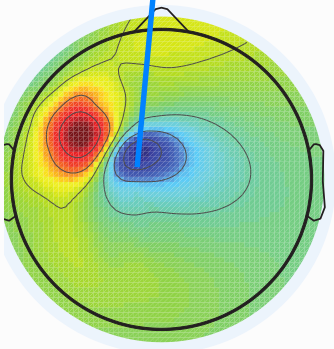
$$T \approx 100$$

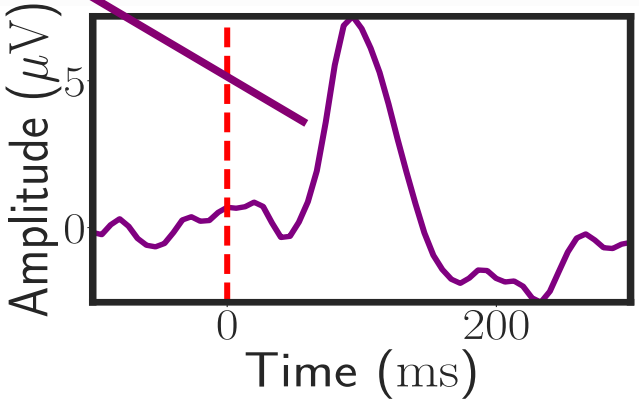
$$p \approx 10000$$

$$n \approx 100$$

$$Y = X$$





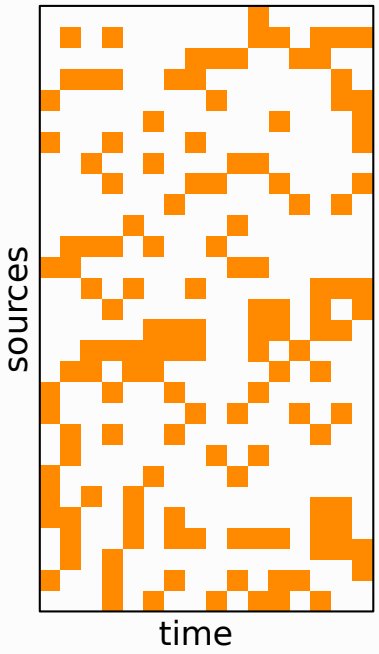


$$n \ll p$$

Multitask penalties[1] and MEG[2]

Popular convex penalties:

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



Parameter $\hat{\mathrm{B}} \in \mathbb{R}^{p \times T}$

Sparse support: no structure

Penalty: **Lasso**

$$\Omega(\mathrm{B}) = \|\mathrm{B}\|_1 = \sum_{j=1}^{p} \sum_{k=1}^{T} |\mathrm{B}_{j,k}|$$

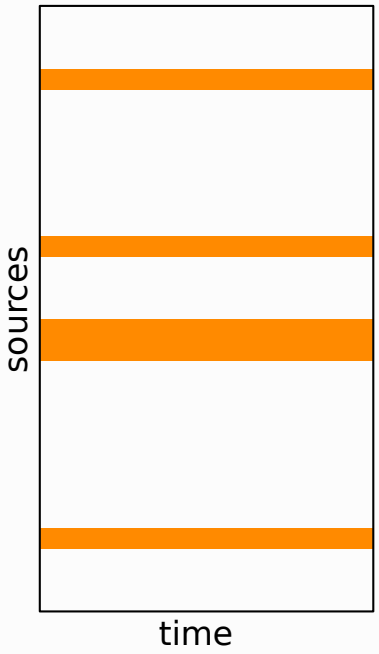
[1] A. Argyriou, T. Evgeniou, and M. Pontil (2008). "Convex multi-task feature learning". In: *Machine Learning*

[2] A. Gramfort, M. Kowalski, and M. Hämäläinen (2012). "Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods". In: *Phys. Med. Biol.*

Multitask penalties[1] and MEG[2]

Popular convex penalties: multitask Lasso (MTL)

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



Parameter $\hat{\mathrm{B}} \in \mathbb{R}^{p \times T}$

Sparse support: group structure ✓

Penalty: **Group-Lasso**

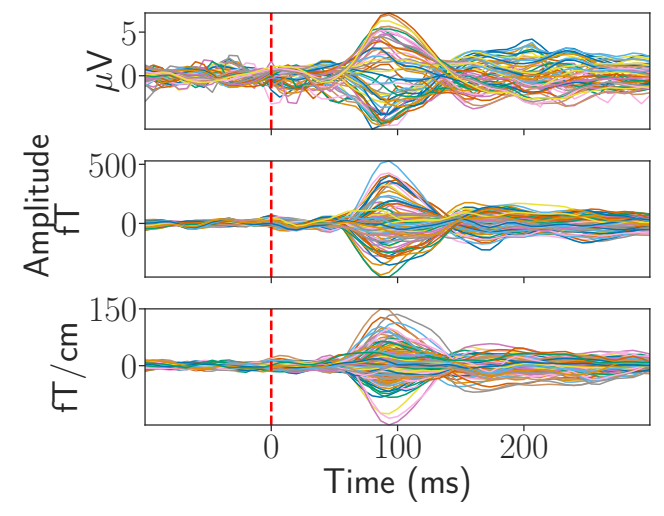
$$\Omega(\mathrm{B}) = \|\mathrm{B}\|_{2,1} = \sum_{j=1}^{p} \|\mathrm{B}_{j,:}\|_2$$

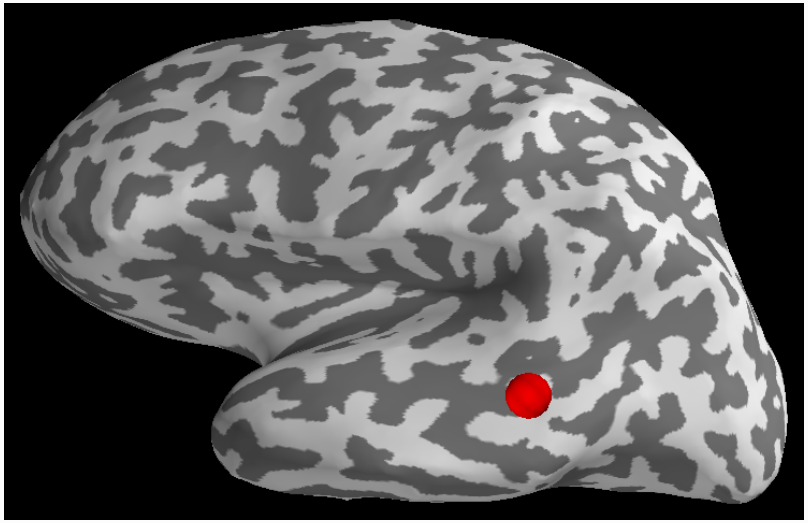
where $\mathrm{B}_{j,:}$ the j-th row of B

(1) A. Argyriou, T. Evgeniou, and M. Pontil (2008). "Convex multi-task feature learning". In: *Machine Learning*

(2) A. Gramfort, M. Kowalski, and M. Hämäläinen (2012). "Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods". In: *Phys. Med. Biol.*

Summary of the problem setting

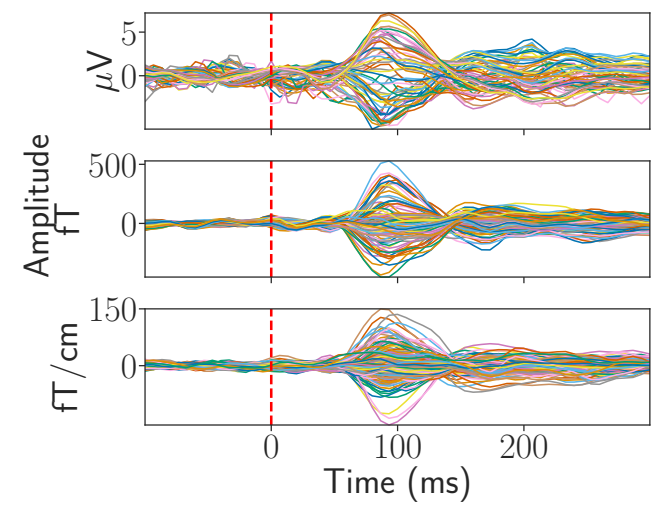


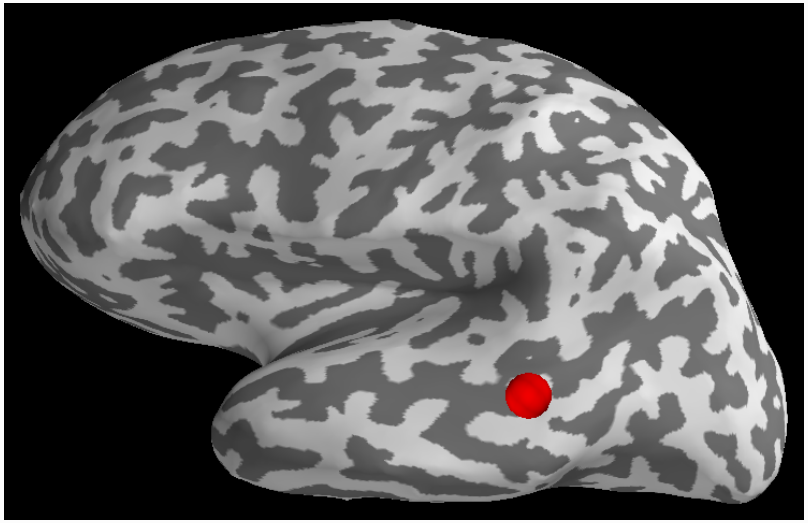


What you have: $Y \in \mathbb{R}^{n \times T}$

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

Summary of the problem setting





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

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

This is typically done using optimization based estimators

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

Summary of contributions

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

Covered in this presentation

- How to efficiently select the regularization parameter λ?[1], [2]

Not covered in this presentation

- How to efficiently solve this optimization problem?[3]
- How to handle spatial correlation?[4]

---

[1] Q. Bertrand, Q. Klopfenstein, M. Blondel, et al. (2020). "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML*.

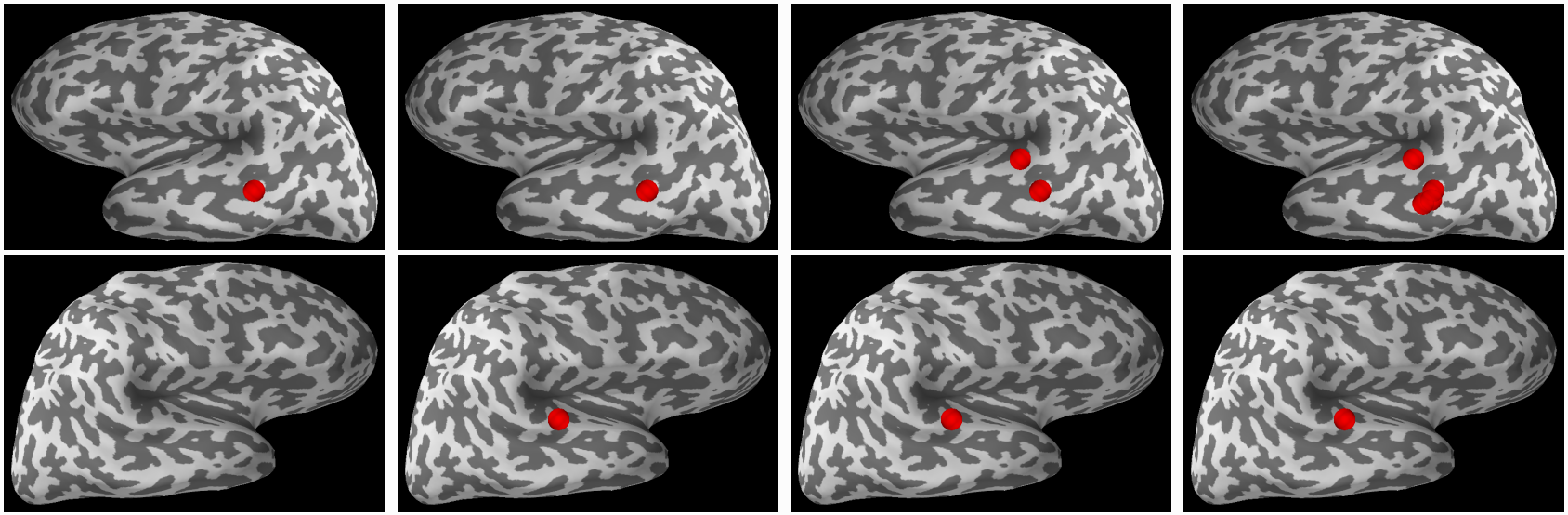
[2] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2021). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*.

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

[4] Q. Bertrand, M. Massias, et al. (2019). "Handling correlated and repeated measurements with the smoothed Multivariate square-root Lasso". In: *NeurIPS*.

Which λ to pick?

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



$\lambda = 0.85\lambda_{\max}$ $\lambda = 0.82\lambda_{\max}$ $\lambda = 0.80\lambda_{\max}$ $\lambda = 0.75\lambda_{\max}$

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

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

Model selection techniques

- Statistical route[1,2]: assumptions on the design matrix X, provide guarantees but are often too
- Bayesian statistics[3,4]: prior on λ
- Bayesian optimisation,[5] 0-th order method:

The road today:

- Hyperparameter optimization[6] : minimize a given criterion $\mathcal{C}(\hat{\beta}^{(\lambda)})$

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

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

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

(4) M. Figueiredo (2001). "Adaptive Sparseness Using Jeffreys Prior.". In: *Advances in Neural Information Processing Systems*.

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

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

Hyperparameter optimization (HO)

Possible selection criterion:

- Good generalization[1,2] of $\hat{\beta}(\lambda)$
- AIC/BIC,[3] SURE[4] that controls model complexity

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

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

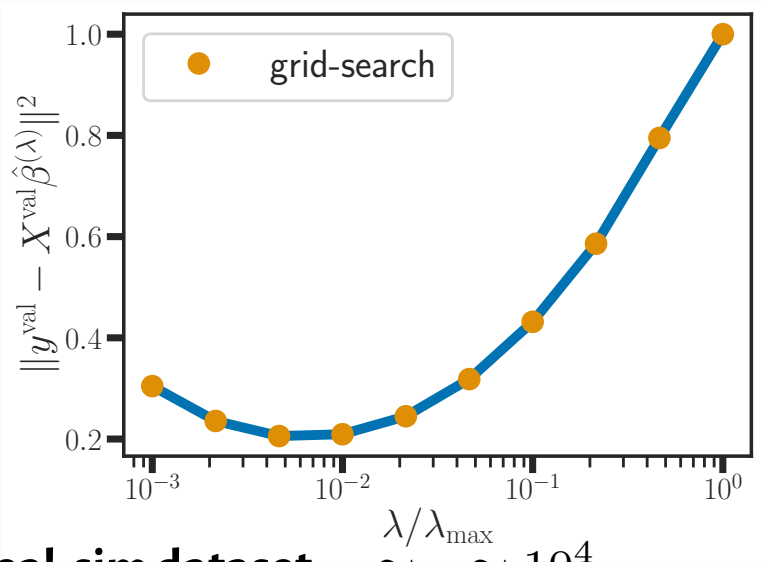
(3) W. Liu and Y. Yang (2011). "Parametric or nonparametric? A parametricness index for model selection". In: *Ann. Statist.* 39.4, pp. 2074–2102.

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

Hyperparameter optimization (HO)

Possible selection criterion:

- Good generalization[1,2] of $\hat{\beta}(\lambda)$
- AIC/BIC,[3] SURE[4] that controls model complexity



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

Simplified example ($T = 1$):

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

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

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

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

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

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

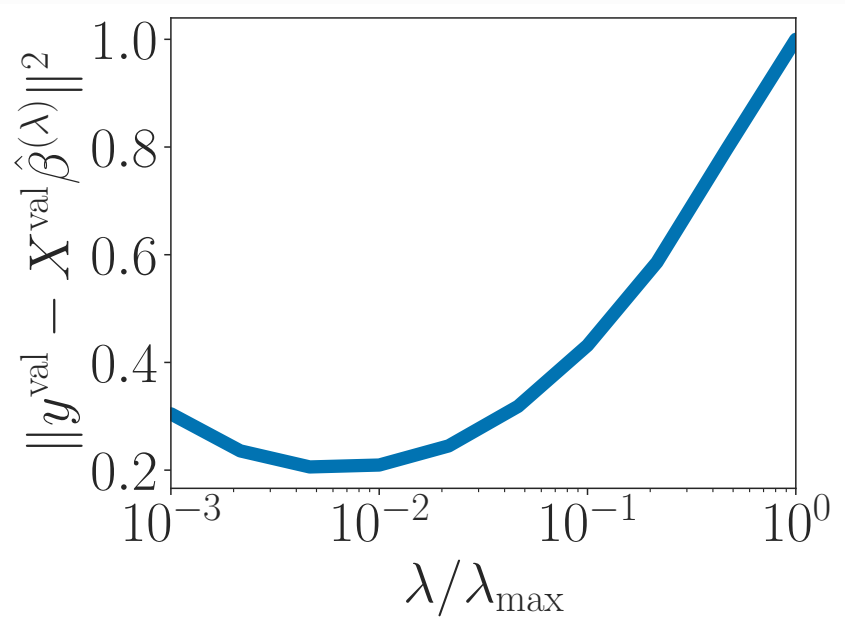
HO as a **bilevel optimization** problem[1], [2]

outer optimization problem

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

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

inner optimization problem



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

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

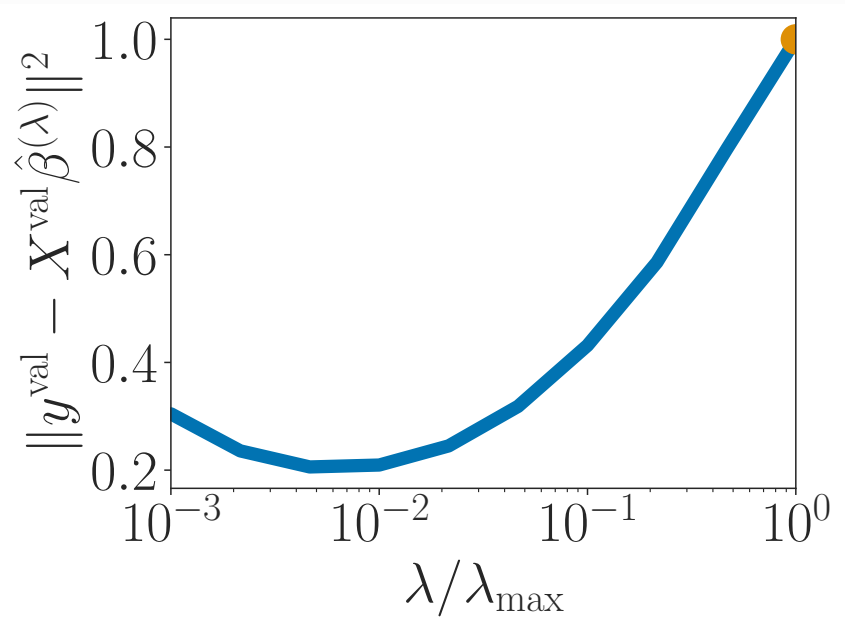
HO as a **bilevel optimization** problem[1], [2]

outer optimization problem

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

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

inner optimization problem



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

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

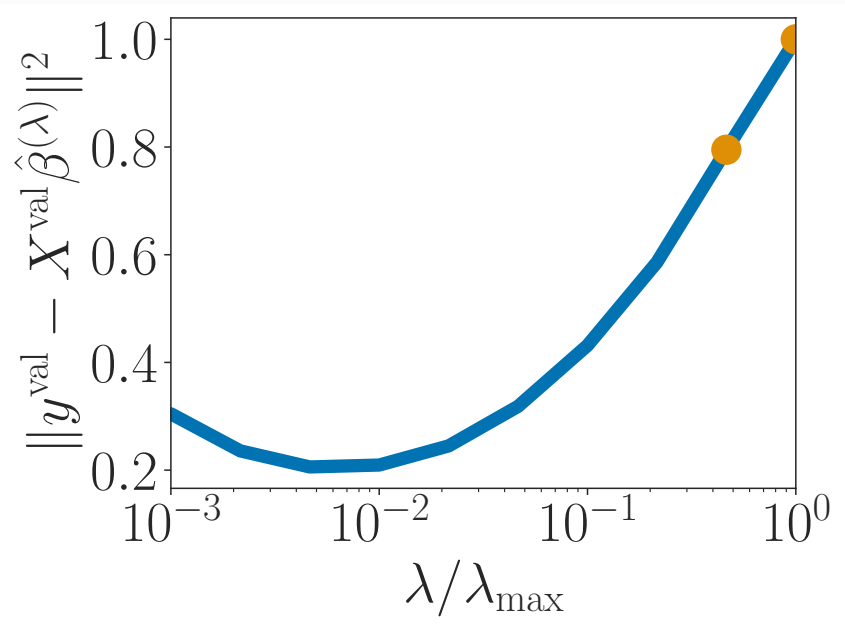
HO as a **bilevel optimization** problem [1], [2]

outer optimization problem

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

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

inner optimization problem



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

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

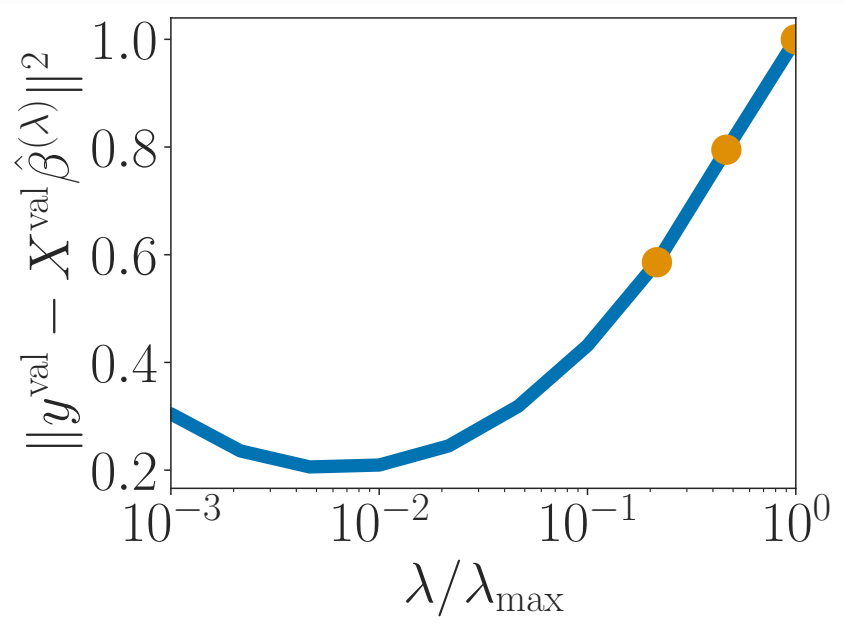
HO as a **bilevel optimization** problem[1], [2]

outer optimization problem

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

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

inner optimization problem



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

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

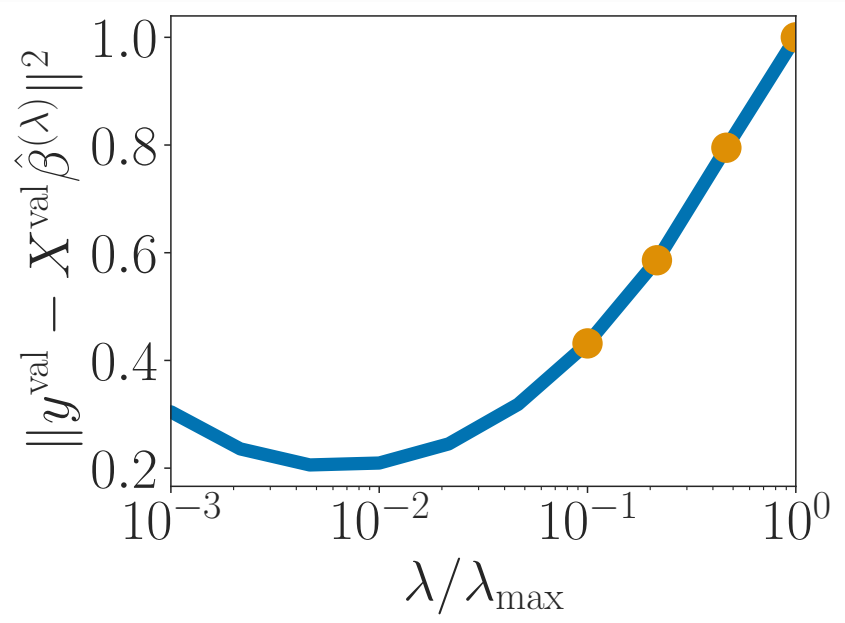
HO AS A **BILEVEL OPTIMIZATION** PROBLEM[1], [2]

outer optimization problem

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

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

inner optimization problem



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

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

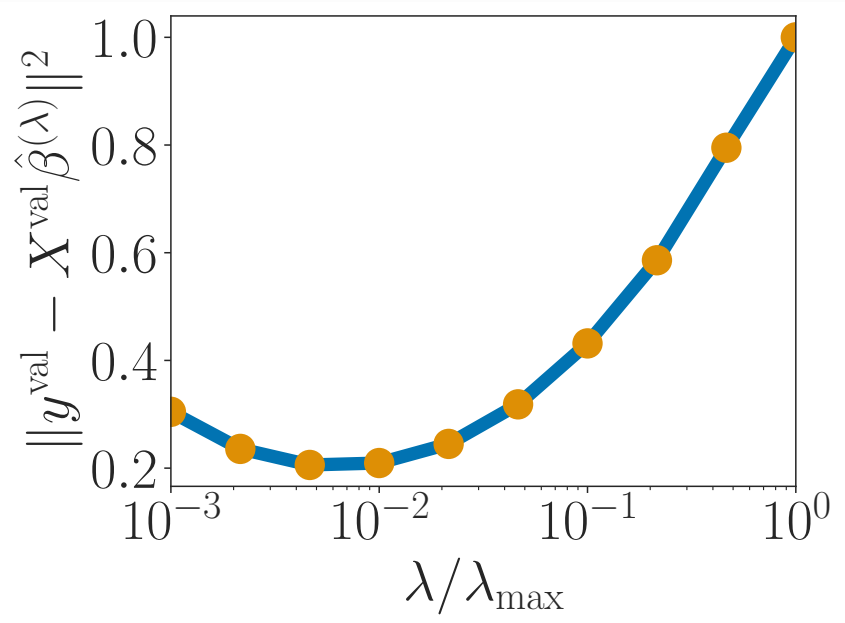
HO AS A **BILEVEL OPTIMIZATION** PROBLEM [1], [2]

outer optimization problem

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

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

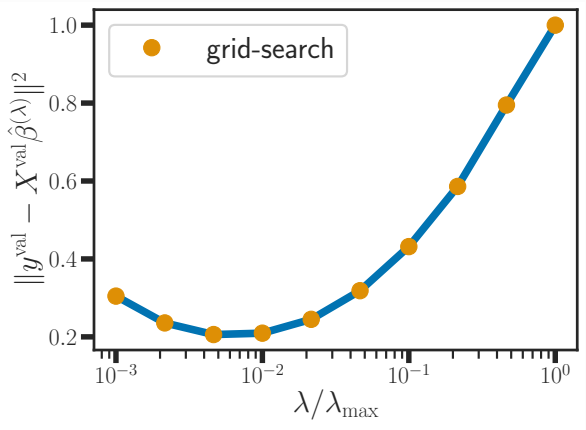
inner optimization problem



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

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

Grid-search
A 0-order optimization method



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

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

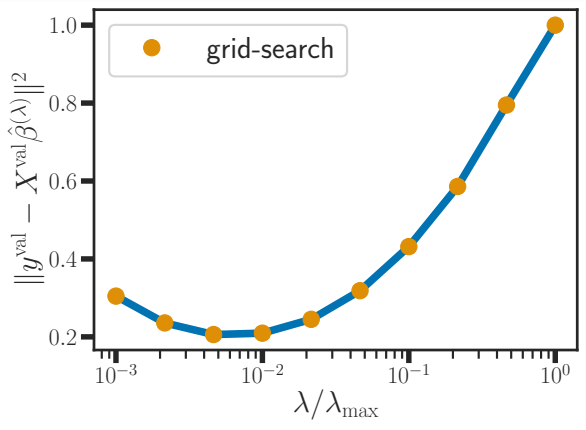
- Grid-search, random-search,[1] SMBO[2]:
  0-order methods to solve bilevel optimization problem

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

(2) E. Brochu, V. M. Cora, and N. D. Freitas (2010). "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*.

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

Grid-search
A 0-order optimization method



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

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

- Grid-search, random-search,[1] SMBO[2]:
  0-order methods to solve bilevel optimization problem
- **Idea:** if $\mathcal{L}$ is differentiable, use 1^{st}-order optimization
  - Compute gradient: $\nabla_\lambda \mathcal{L}$
  - Perform gradient descent step[3]:

$$\lambda^{(t+1)} = \lambda^{(t)} - \rho \nabla_\lambda \mathcal{L}(\lambda^{(t)}) \quad \text{with } \rho > 0$$

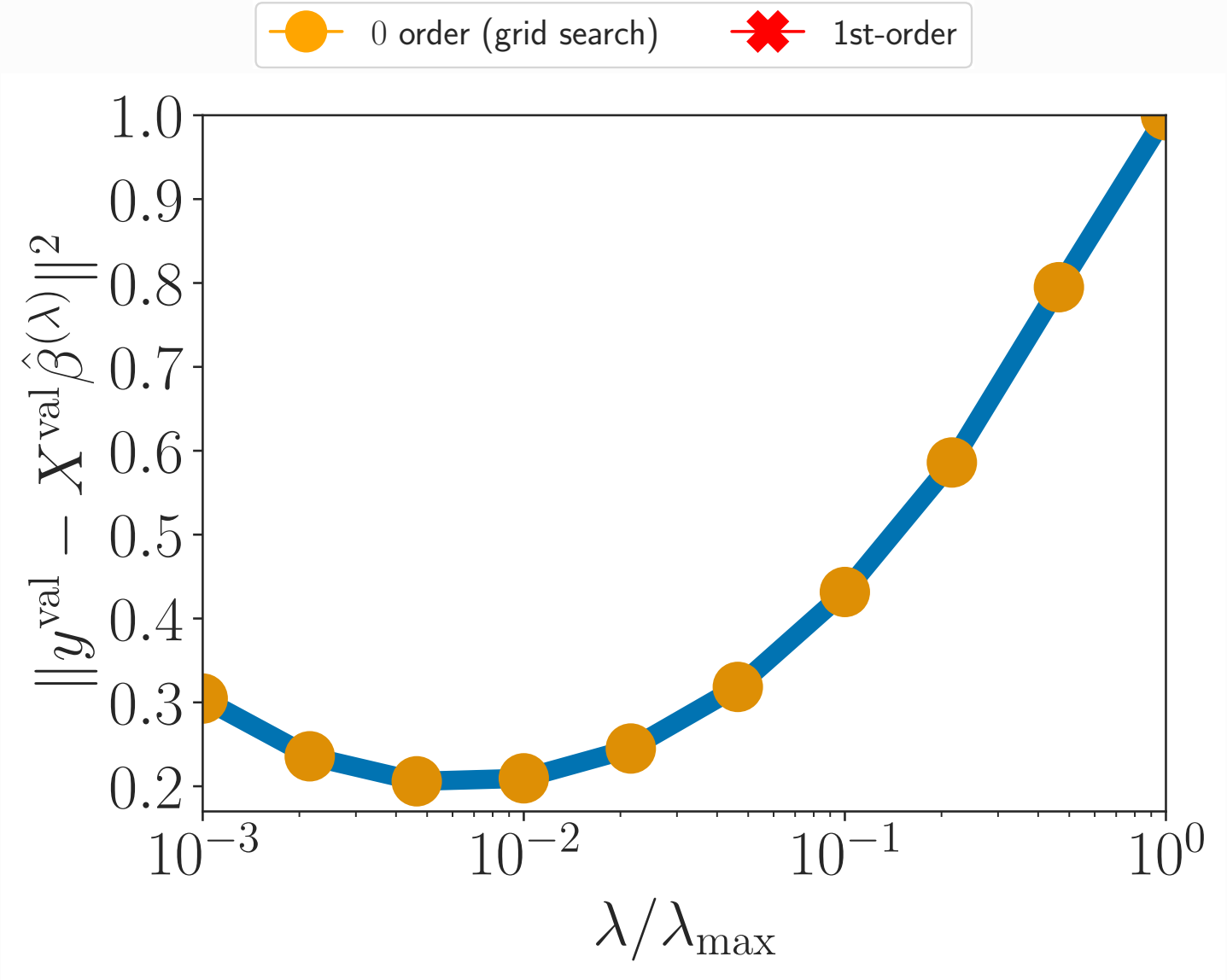
---

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

(2) E. Brochu, V. M. Cora, and N. D. Freitas (2010). "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*.

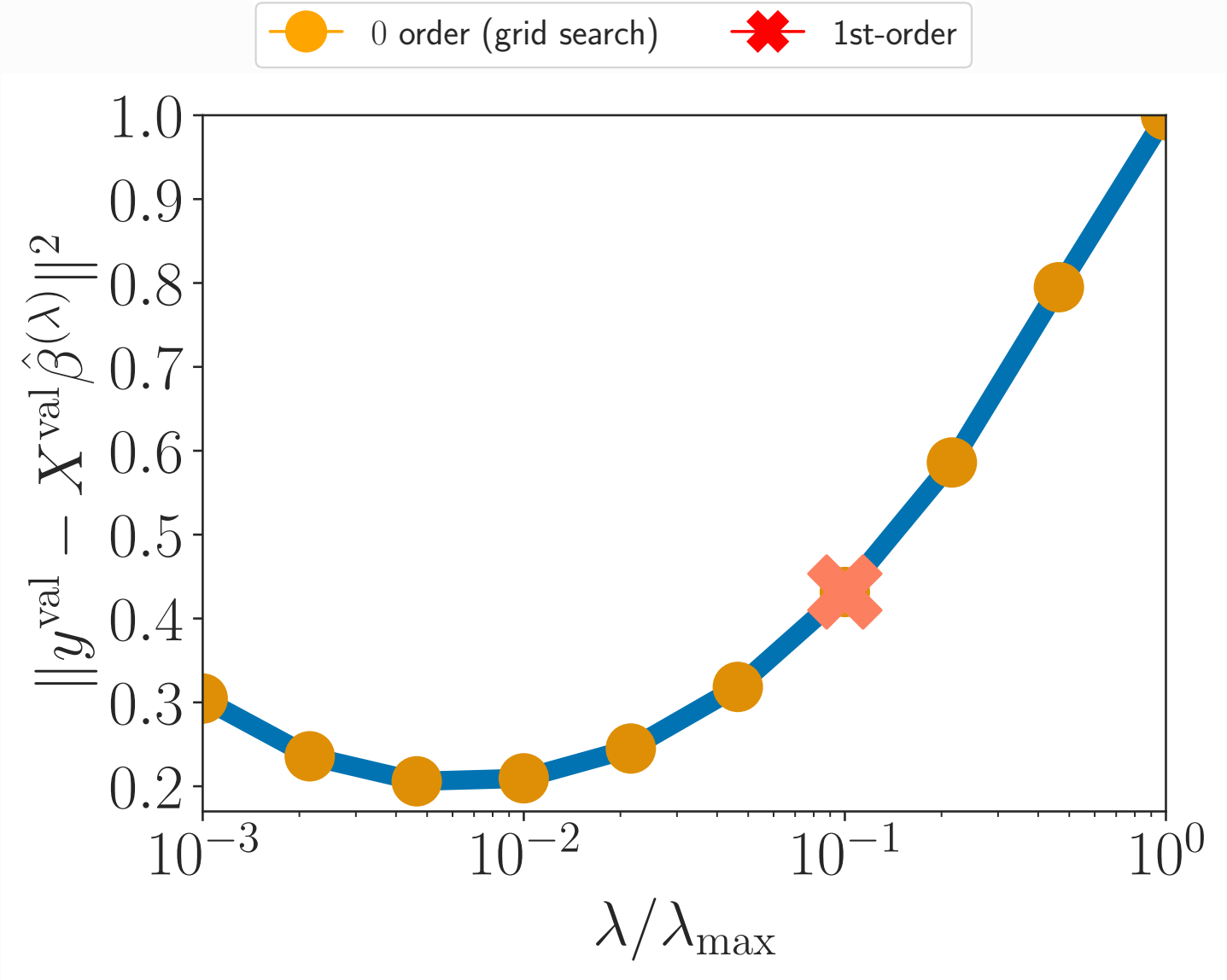
(3) F. Pedregosa (2016). "Hyperparameter optimization with approximate gradient". In: *ICML*.

First-order optimization in λ, Lasso



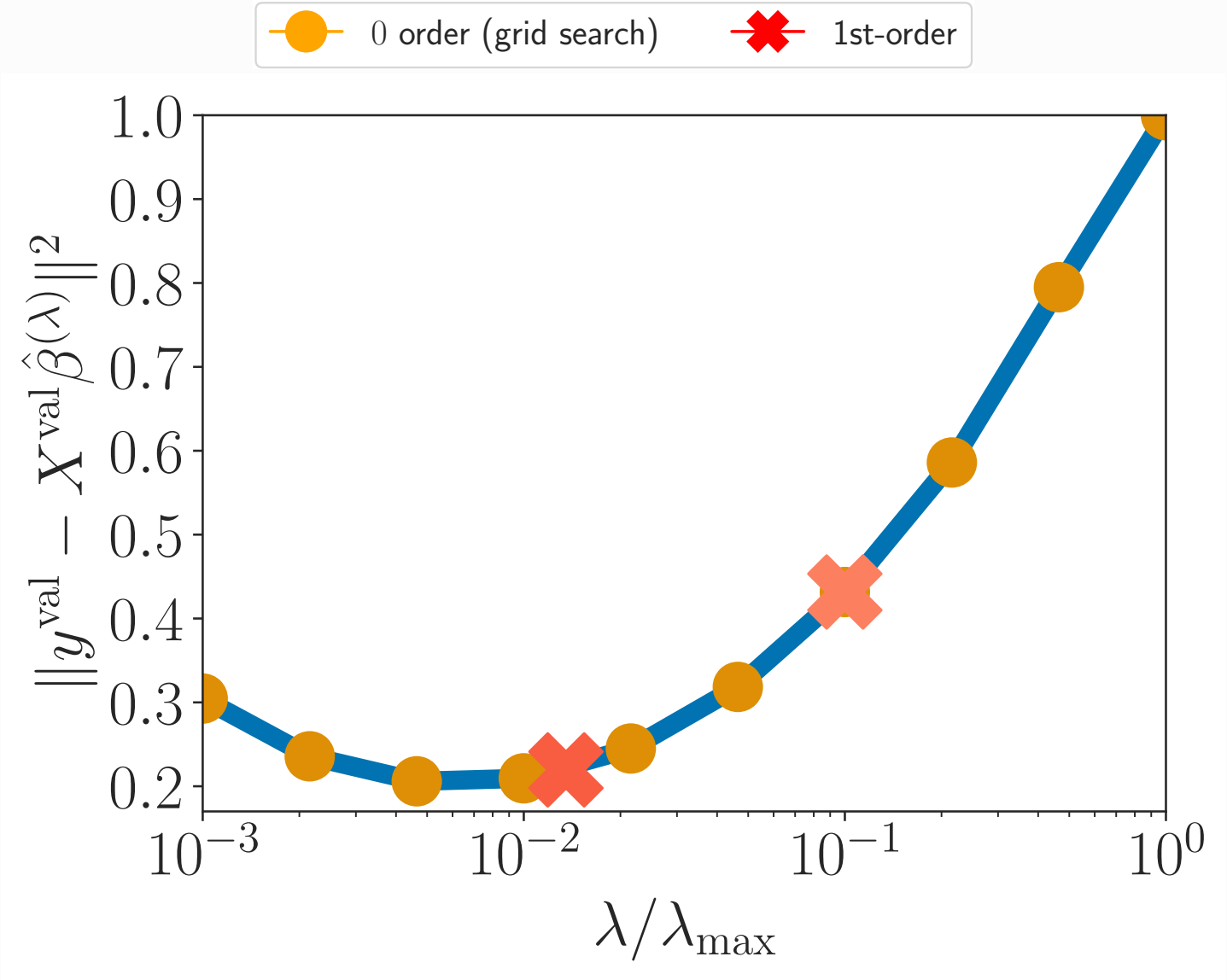
Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

First-order optimization in λ, Lasso



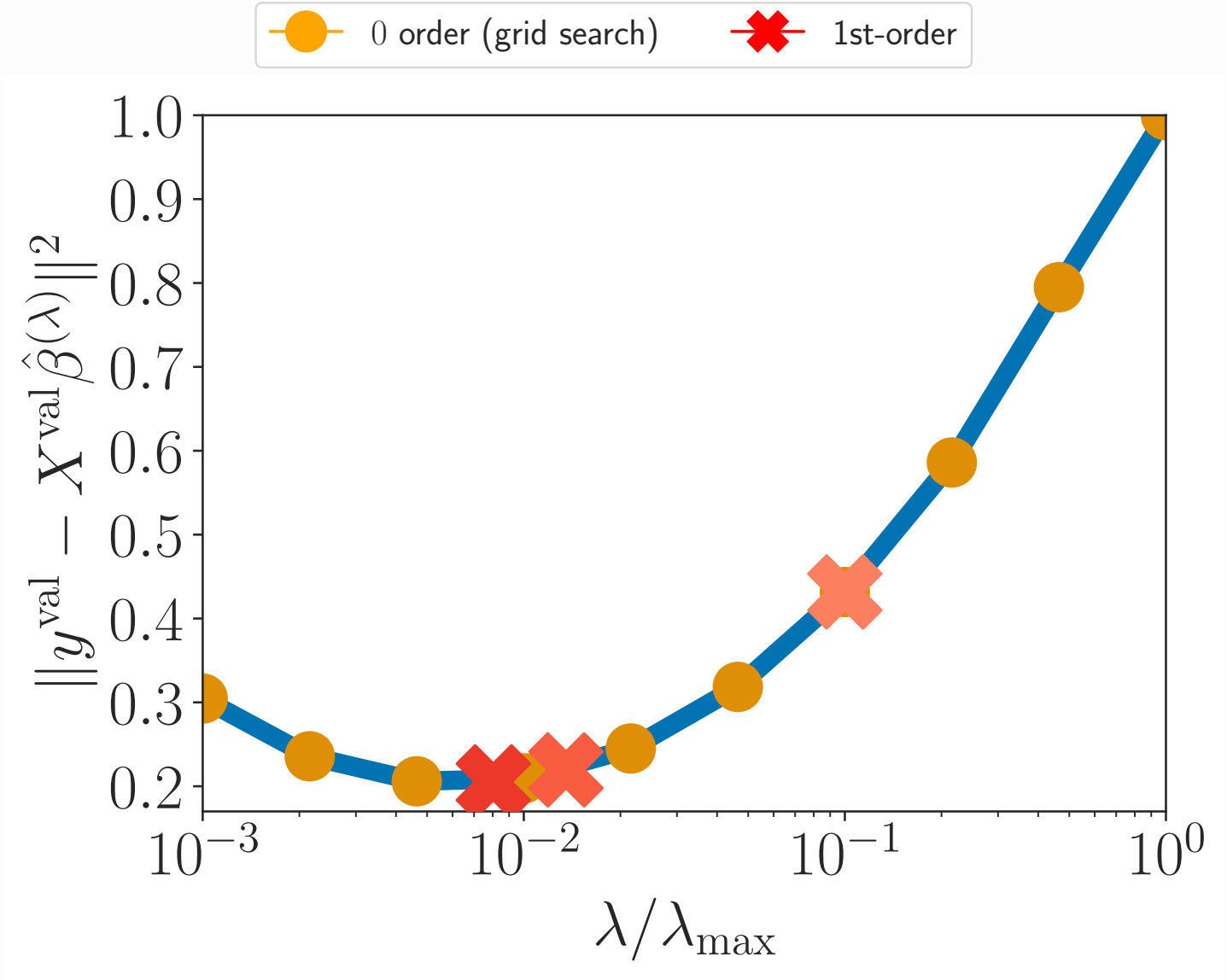
Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

First-order optimization in λ, Lasso



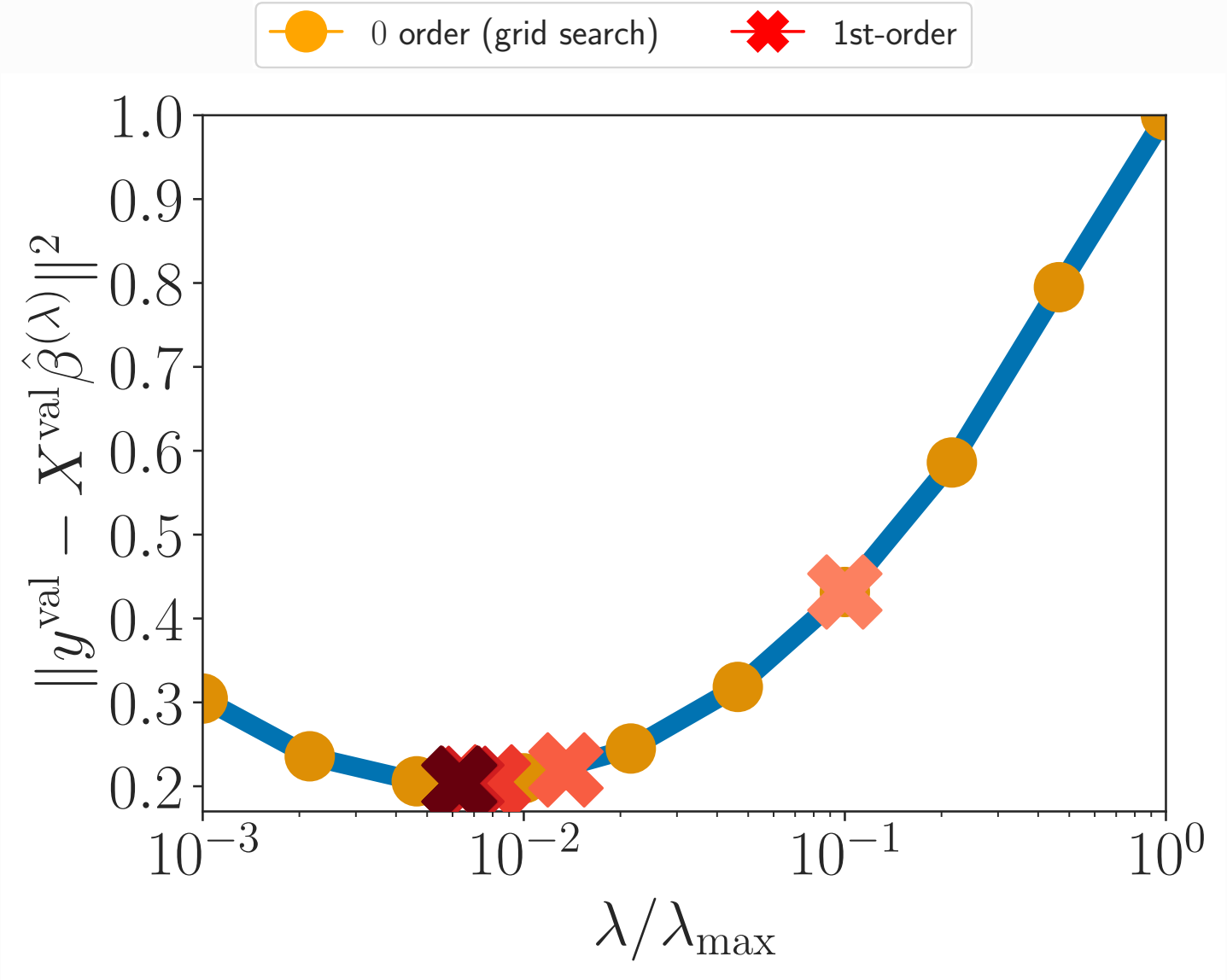
Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

First-order optimization in λ, Lasso



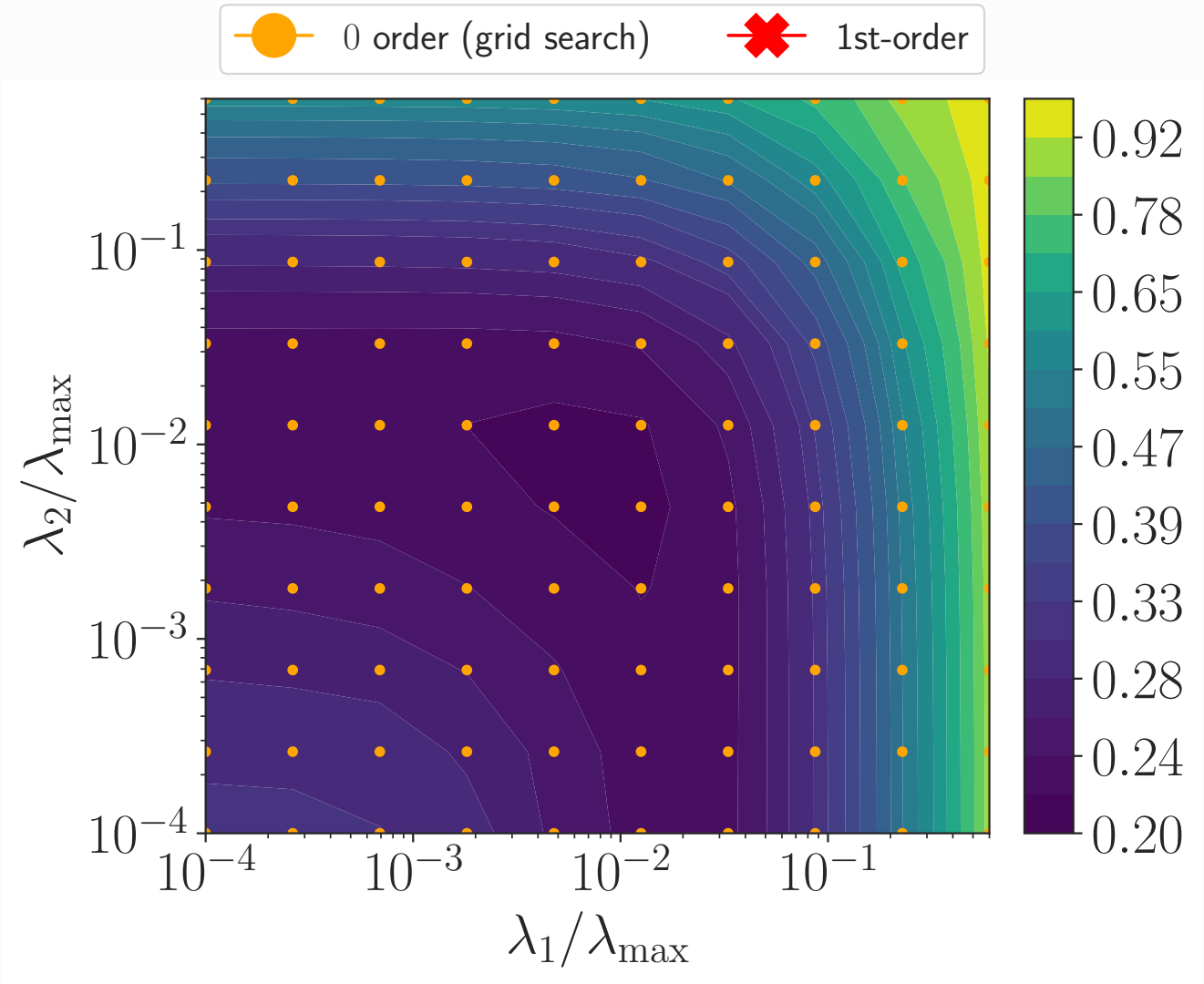
Real-sim dataset, $n \approx p \approx 10^4$**.** Validation loss as a function of λ.

First-order optimization in λ, Lasso



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

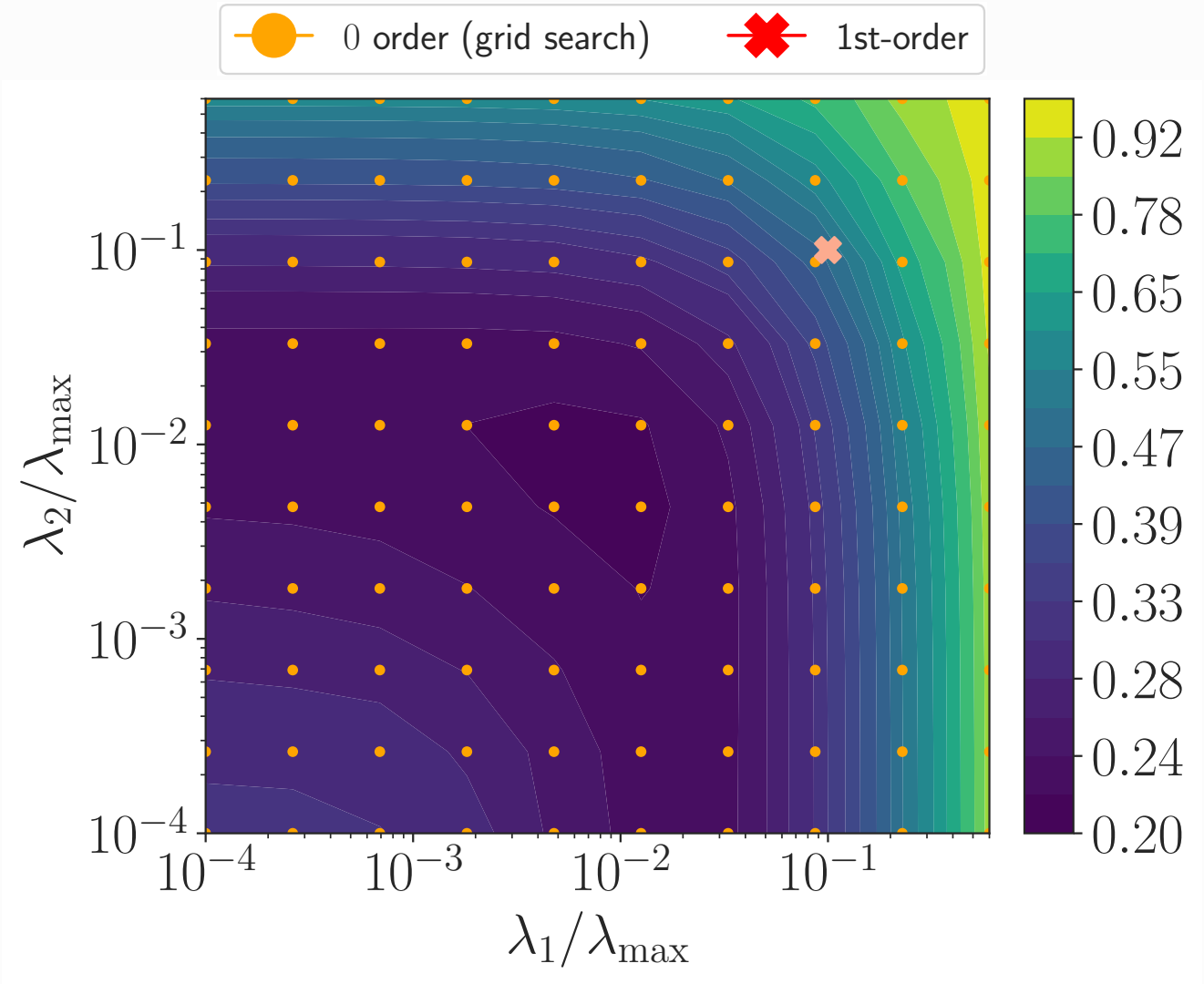
First-order optimization in λ, Enet



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

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

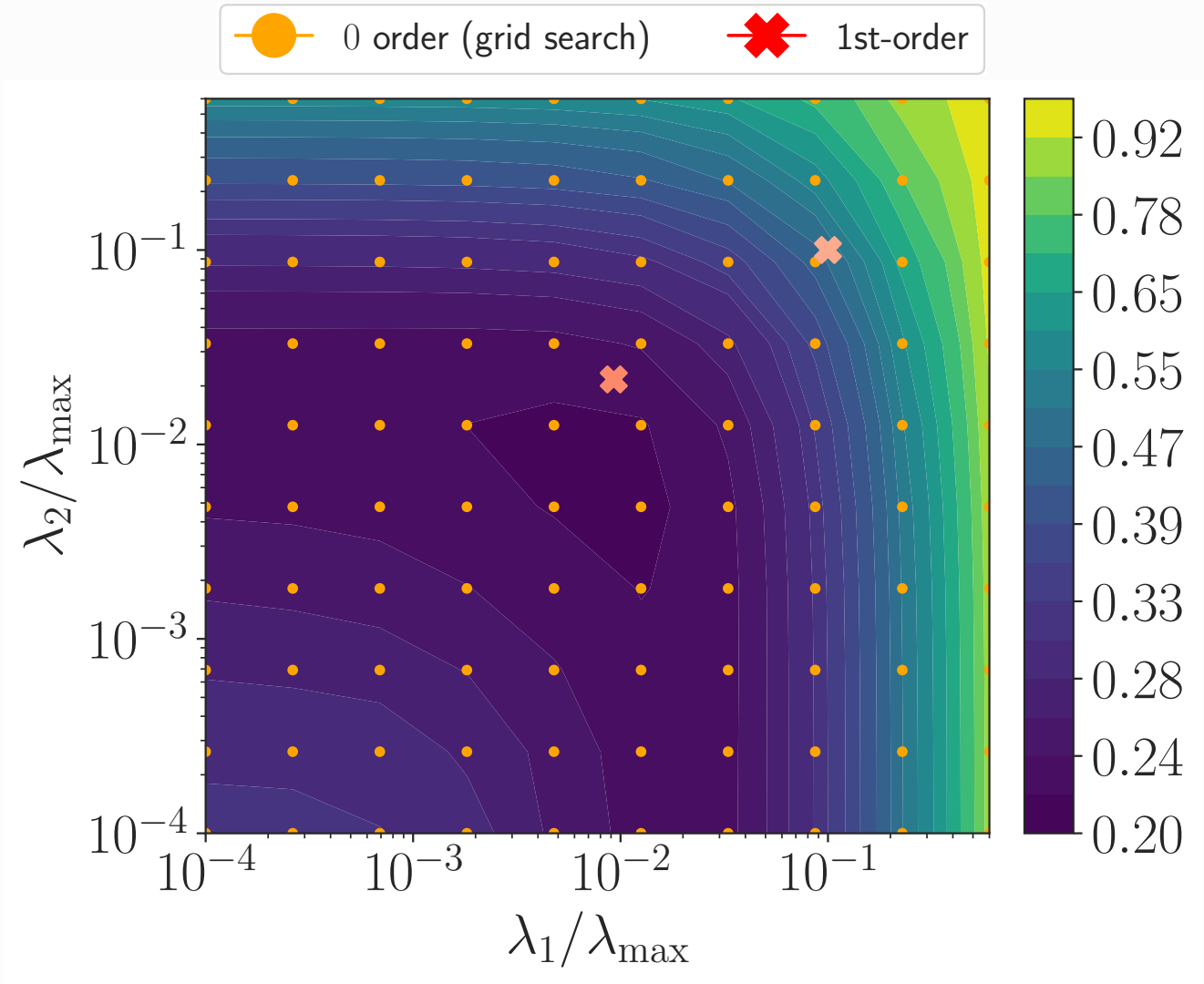
First-order optimization in λ, Enet



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

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

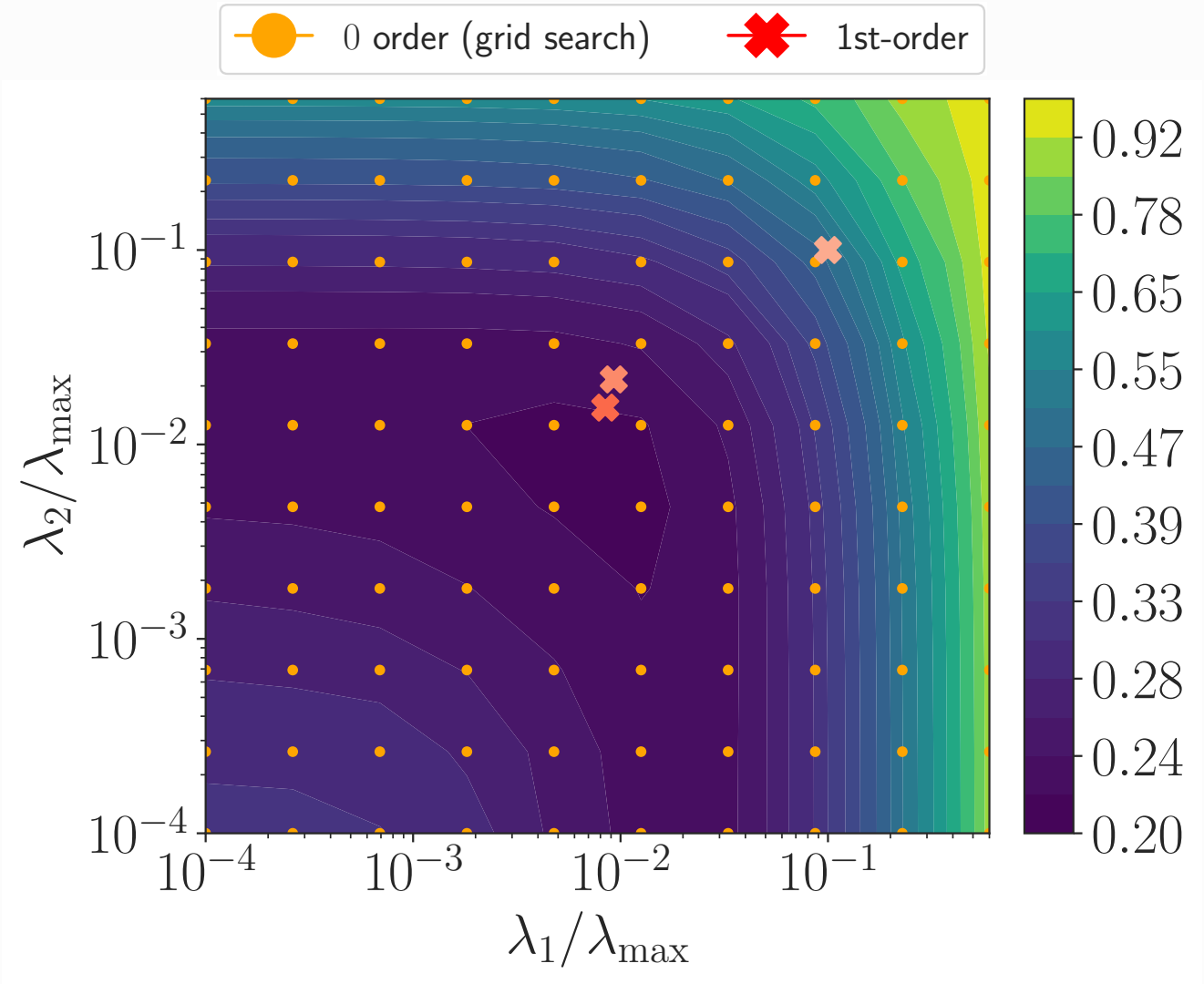
First-order optimization in λ, Enet



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

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

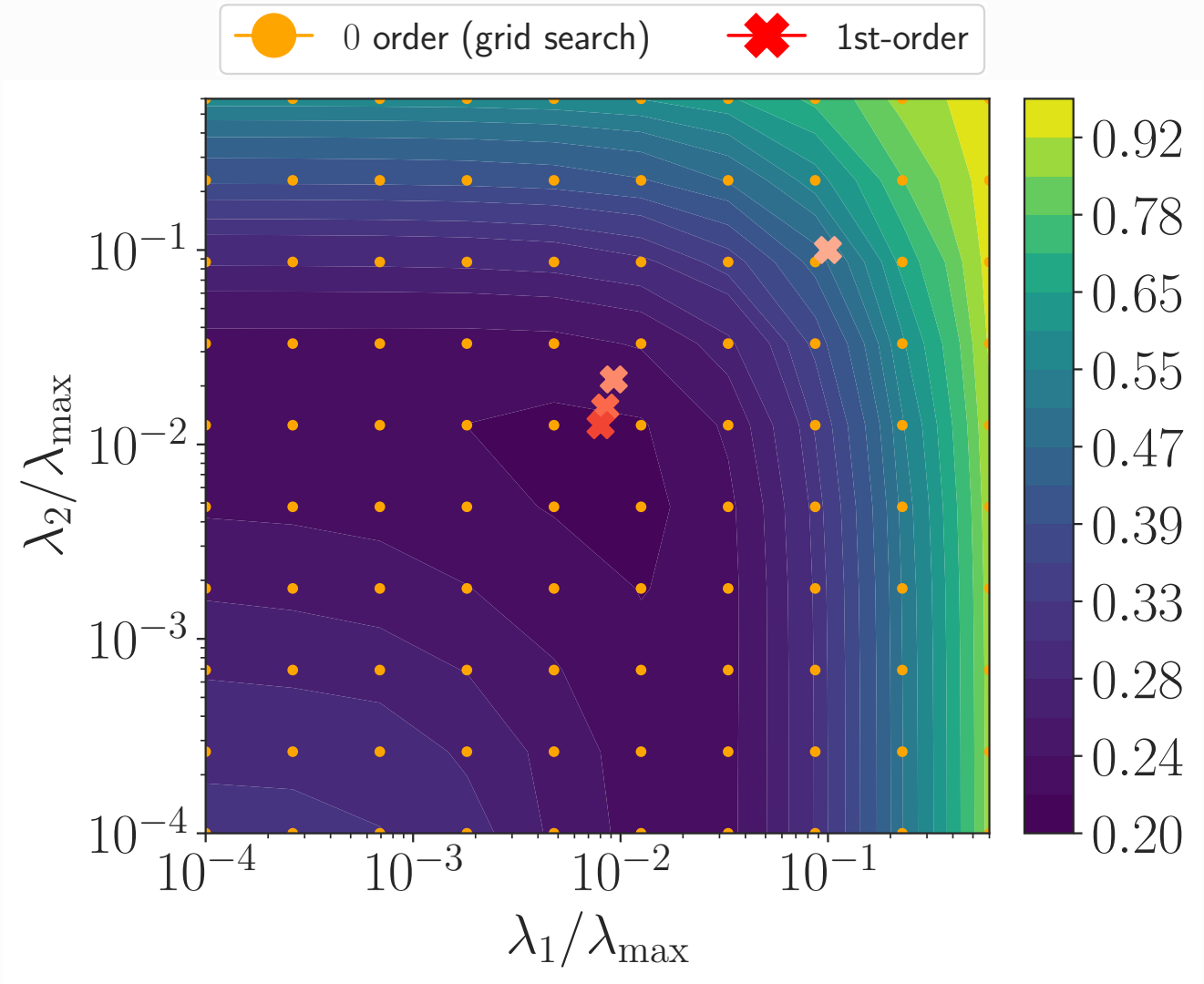
First-order optimization in λ, Enet



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

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

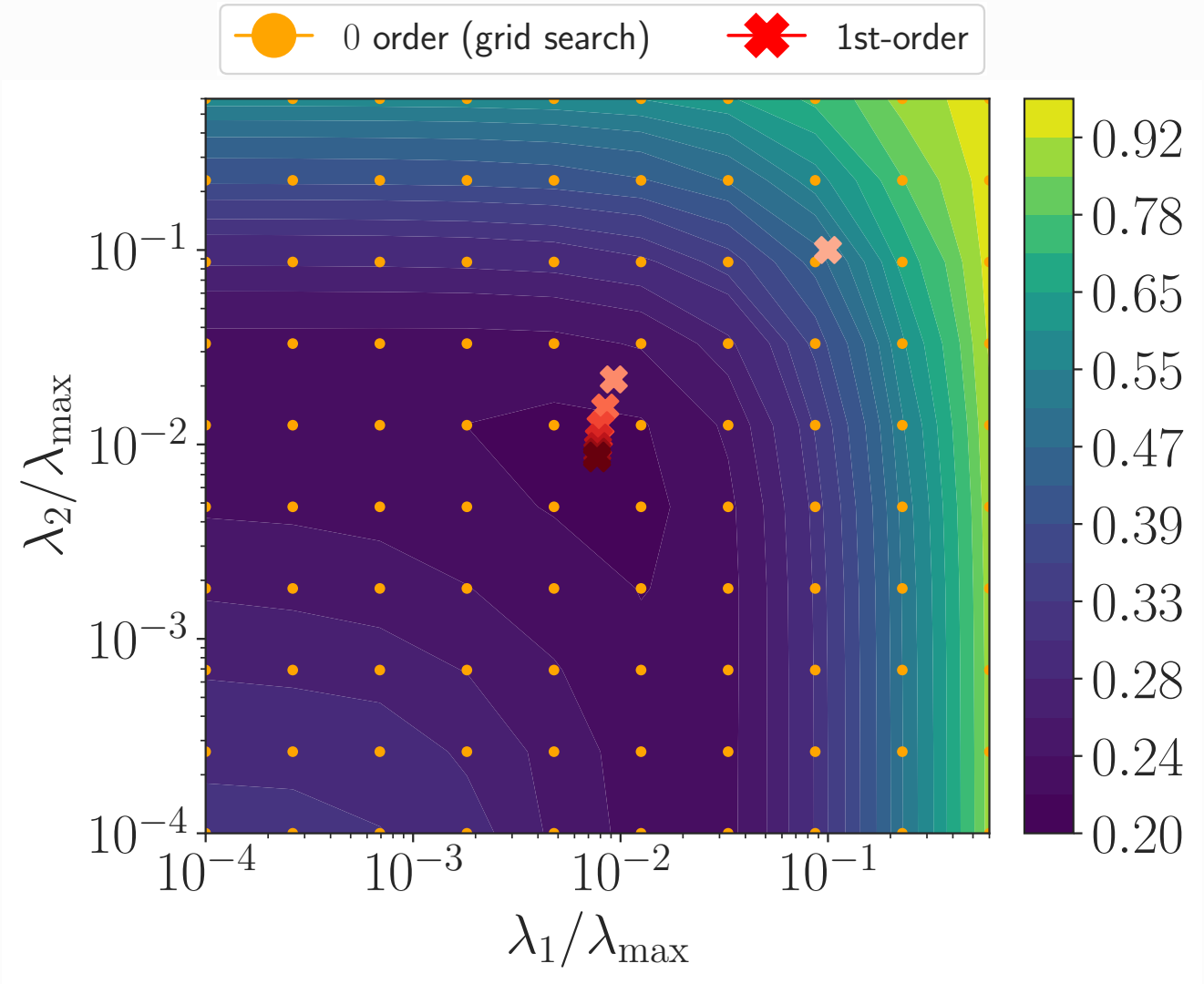
First-order optimization in λ, Enet



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)

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

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

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

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

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

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

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

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

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

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

- Line-search [1]
- L-BFGS [2]
- Gradient descent

---

[1] J. Nocedal and S. J. Wright (2006). *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. Springer, Chap. 3.

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

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

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

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

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

- Line-search [1]
- L-BFGS [2]
- Gradient descent

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

---

[1] J. Nocedal and S. J. Wright (2006). *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. Springer, Chap. 3.

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

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

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

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

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:

$$\nabla_\lambda \mathcal{L}(\lambda) = \underbrace{\hat{\mathcal{J}}_{(\lambda)}^\top}_{:=(\nabla_\lambda \hat{\beta}_1^{(\lambda)}, \ldots, \nabla_\lambda \hat{\beta}_p^{(\lambda)})} \nabla_\beta C(\hat{\beta}^{(\lambda)})$$

$\rightarrow$main challenge

- Boils down to:

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

Forward-mode differentiation [1], [2] of PGD

Proximal Gradient Descent [3]

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

Algorithm: Proximal gradient descent PGD

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

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

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

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

return β

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

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

[3] B. Martinet (1970). "Brève communication. Régularisation d'inéquations variationnelles par approximations successives". In: *Revue française d'informatique et de recherche opérationnelle. Série rouge* 4.R3, pp. 154–158

Forward-mode differentiation [1], [2] of PGD

Proximal Gradient Descent [3]

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

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p$, $\mathcal{J} = 0_p$, L (Lipschitz-constant for f)

for iter $= 1, \ldots,$ **do**

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

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

$\beta \leftarrow \mathrm{prox}_{\lambda g / L}(z)$ // proximal step: thresholding for us

return β

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

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

[3] B. Martinet (1970). "Brève communication. Régularisation d'inéquations variationnelles par approximations successives". In: *Revue française d'informatique et de recherche opérationnelle. Série rouge* 4.R3, pp. 154–158

Forward-mode differentiation [1], [2] of PGD

Proximal Gradient Descent [3]

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

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p$, $\mathcal{J} = 0_p$, L (Lipschitz-constant for f)

for $\text{iter} = 1, \ldots,$ **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: thresholding for us

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

return $\beta, \mathcal{J}$

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

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

(3) B. Martinet (1970). "Brève communication. Régularisation d'inéquations variationnelles par approximations successives". In: *Revue française d'informatique et de recherche opérationnelle. Série rouge* 4.R3, pp. 154–158

Forward-mode differentiation [1], [2] of PGD

Proximal Gradient Descent [3]

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

Algorithm: Forward-mode differentiation of PGD

init : $\beta = 0_p, \mathcal{J} = 0_p, L$ (Lipschitz-constant for f)

for $\text{iter} = 1, \ldots,$ **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: thresholding for us

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

$\quad + \partial_\lambda \text{prox}_{\lambda g/L}(z)$ // do not forget this term!

return $\beta, \mathcal{J}$

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

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

[3] B. Martinet (1970). "Brève communication. Régularisation d'inéquations variationnelles par approximations successives". In: *Revue française d'informatique et de recherche opérationnelle. Série rouge* 4.R3, pp. 154–158

A brief détour
Proximal operators [2]

Motivation for proximal operators:

smooth non-smooth functions (sic),
e.g., to perform gradient descent

$$\operatorname{prox}_{\frac{\lambda}{L}g}(z) = \underset{z' \in \mathbb{R}^p}{\arg\min}\ \frac{\lambda}{L} g(z') + \frac{1}{2}\left\|z' - z\right\|^2$$

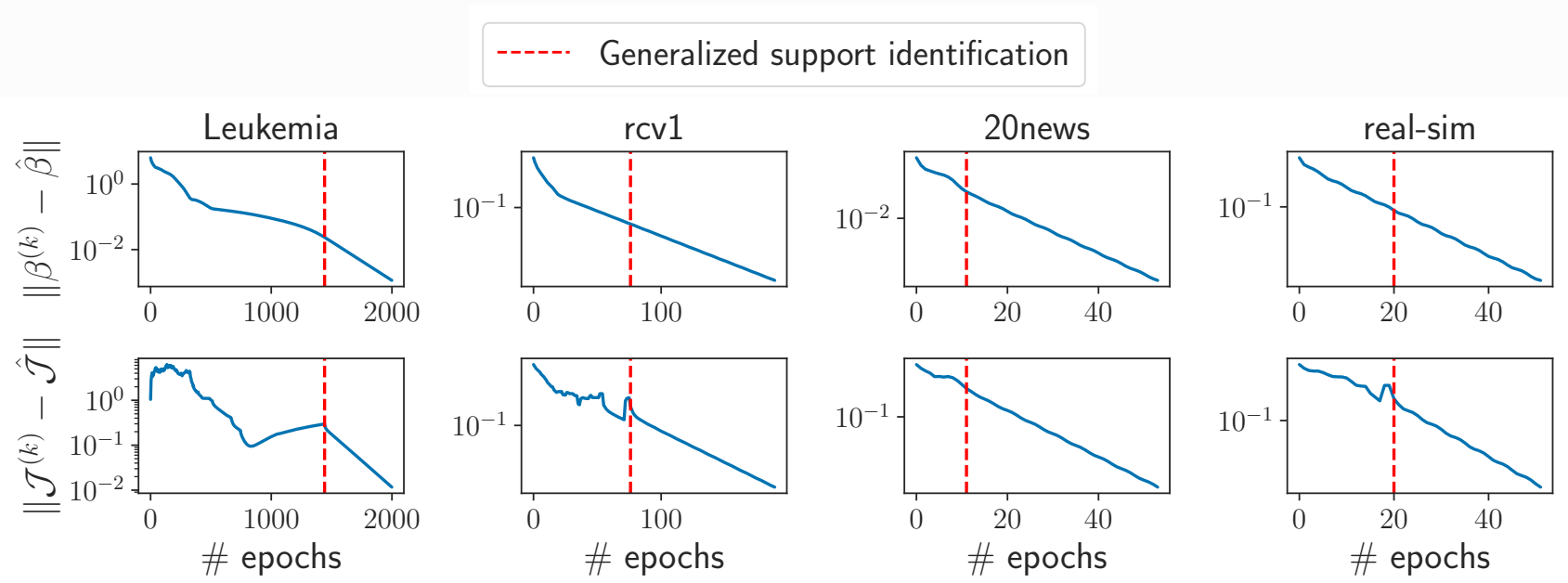
Jean-Jacques Moreau

[3] J.-J. Moreau (1962). "Fonctions convexes duales et points proximaux dans un espace hilbertien". In: *C. R. Acad. Sci. Paris* 255, pp. 2897–2899

Local linear convergence of the Jacobian

Forward diff. PCD convergence, Lasso

Provided that X (the design matrix) is not pathological, the sequence generated by PCD is converging to $\hat{\beta}$, and
the Jacobian sequence based on forward differentiation converges to the true Jacobian. Moreover, once the support (the non-zeros coefs.) has been identified, convergence is linear.[1]



[1] Q. Bertrand, Q. Klopfenstein, M. Blondel, et al. (2020). "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML*

Implicit differentiation (smooth ψ) [1]

$$\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \psi\left(\beta, \lambda\right)$$

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

Implicit differentiation (smooth ψ) [1]

$$\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \psi\left(\beta, \lambda\right)$$

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

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

Implicit differentiation (smooth ψ) [1]

$$\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \psi\left(\beta, \lambda\right)$$

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

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

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

Implicit differentiation (smooth ψ) [1]

$$\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \psi\left(\beta, \lambda\right)$$

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

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

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

- Need to solve a linear **system of size** p
- Prohibitive for large p

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

Implicit differentiation:
Blessing of sparsity

General formulation:

- Solve a linear **system of size** p
- Prohibitive for large p

With a sparsity inducing penalty:

- Solve a linear **system of size** $\mathcal{S}$ (sparsity degree of the estimator)
- $\mathcal{S} \ll p$ very often

Implicit diff.[1] $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$

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

[1] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2021). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*

Implicit diff.[1] $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$

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

$$\hat{\mathcal{J}} = \partial_\beta \,\mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right) + \left(\mathrm{Id} - \frac{\nabla^2 f}{L}\right)\hat{\mathcal{J}} + \partial_\lambda \,\mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

[1] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2021). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*

Implicit diff.[1] $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$

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

$$\hat{\mathcal{J}} = \partial_\beta \,\mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right) + \left(\mathrm{Id} - \frac{\nabla^2 f}{L}\right)\hat{\mathcal{J}} + \partial_\lambda \,\mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

Key observation, if $\hat{\beta}_j^{(\lambda)} = 0$:

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

[1] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2021). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*

Implicit diff.[1] $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda \sum_j |\beta_j|$

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

$$\hat{\mathcal{J}} = \partial_\beta \,\mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$
$$+ \left(\mathrm{Id} - \frac{\nabla^2 f}{L}\right)\hat{\mathcal{J}} + \partial_\lambda \,\mathrm{ST}\left(\hat{\beta}^{(\lambda)} - \frac{1}{L}\nabla f(\hat{\beta}^{(\lambda)}), \frac{\lambda}{L}\right)$$

Key observation, if $\hat{\beta}_j^{(\lambda)} = 0$:

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

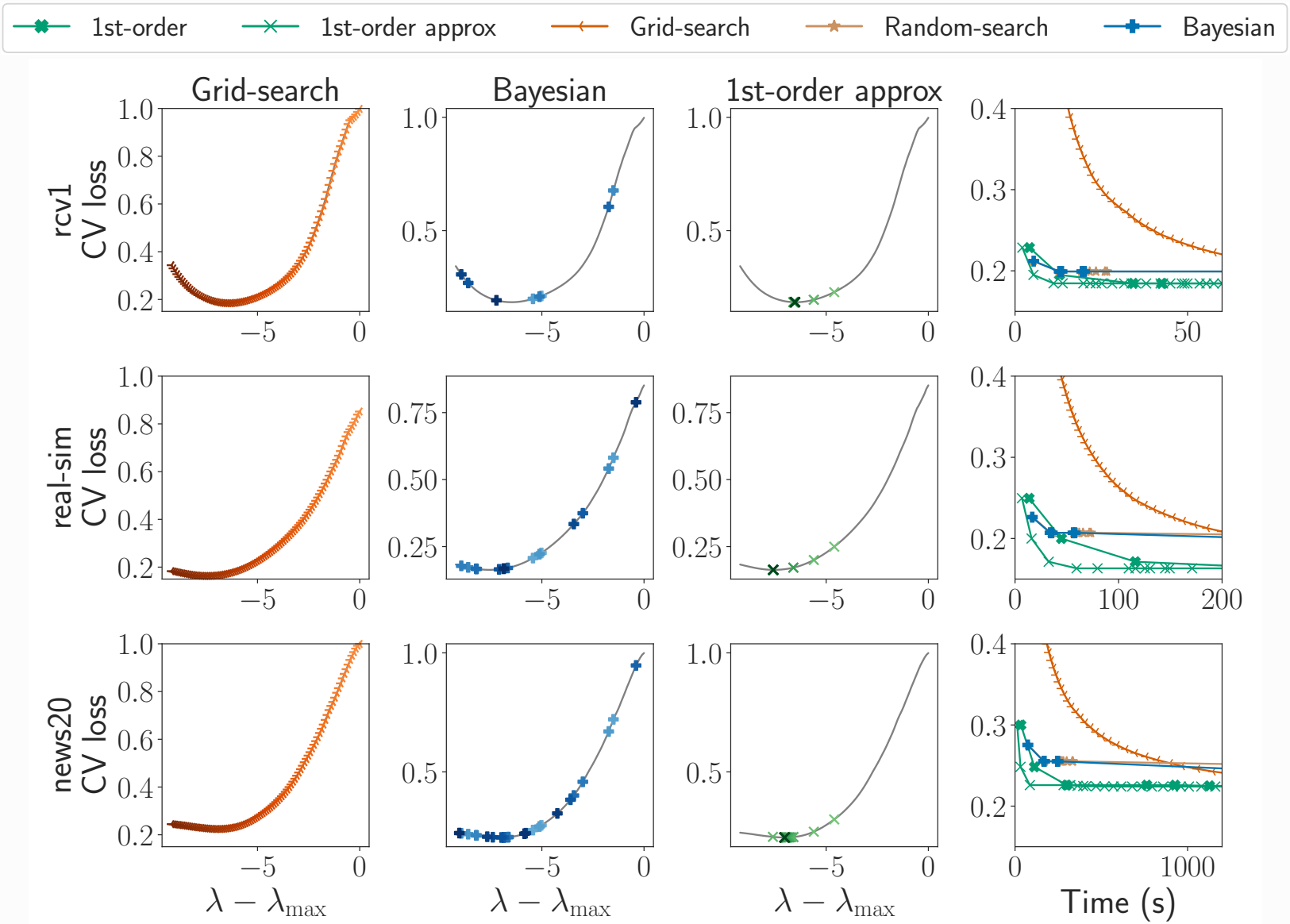
With $\mathcal{S} = \left\{j \in [p] \,:\, \hat{\beta}_j^{(\lambda)} = 0\right\}$ we have $\hat{\mathcal{J}}_{\mathcal{S}^c} = 0$

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

---

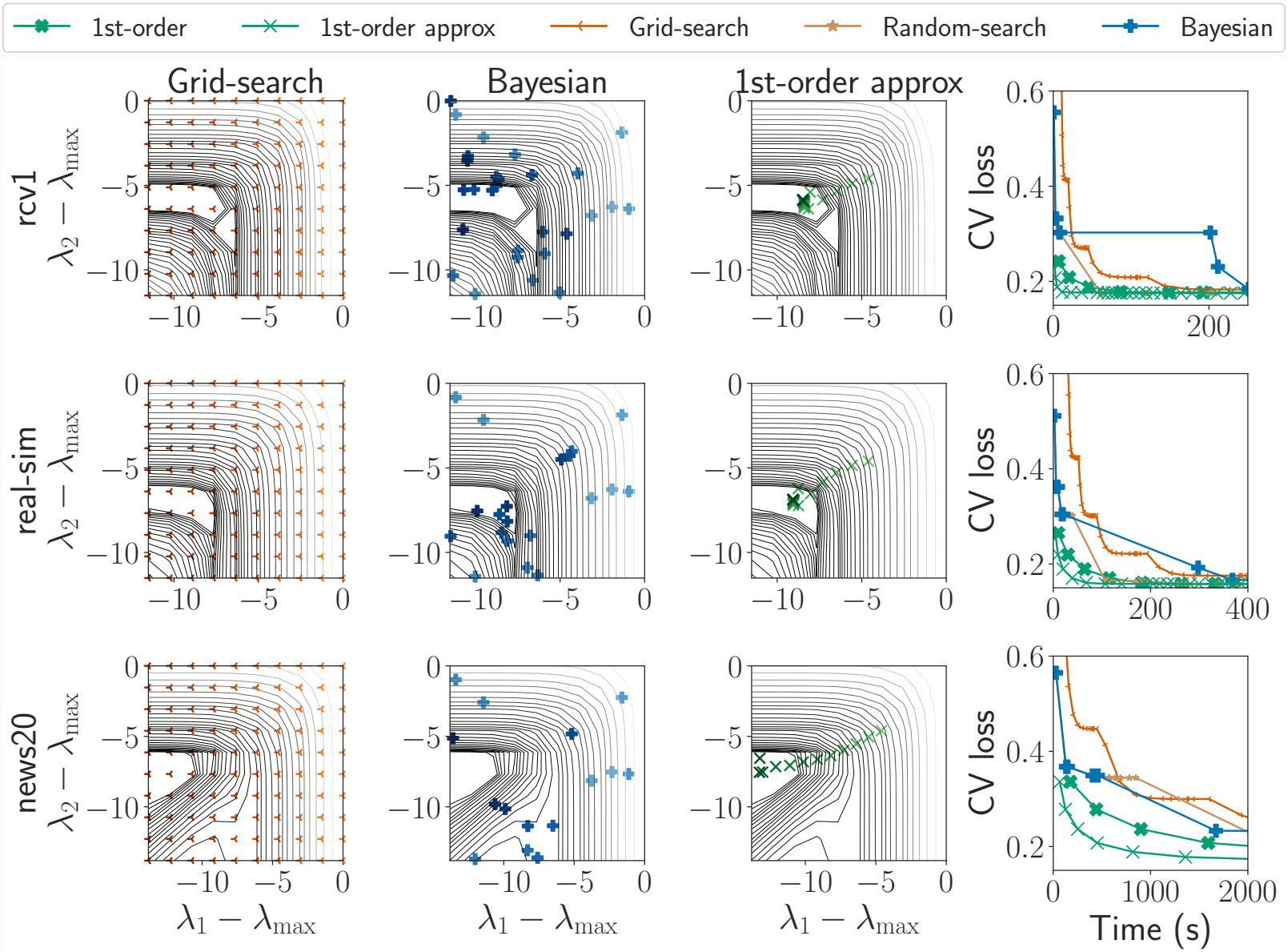
[1] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2021). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*

Experiments I - Lasso cross-validation



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

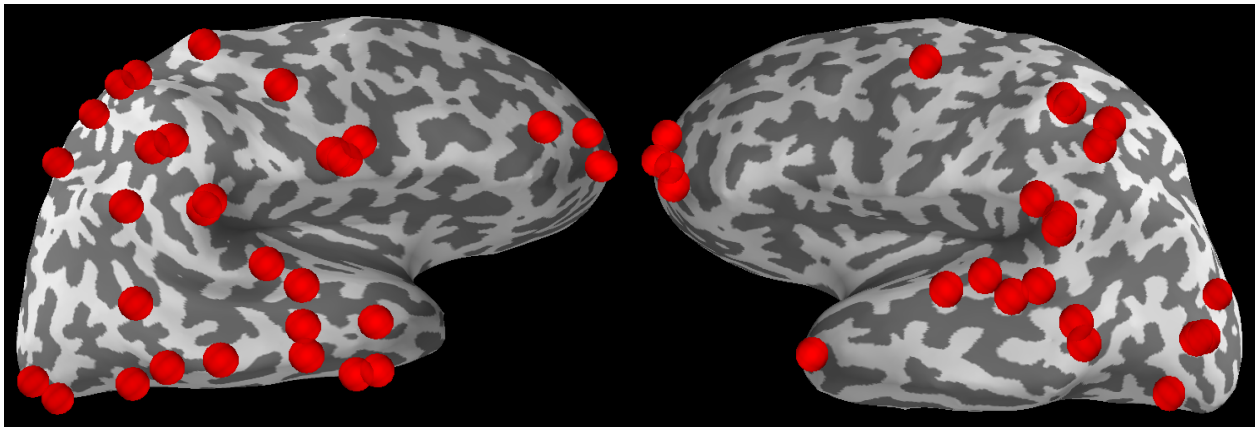
Experiments II - Enet cross-validation



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

Experiments III - Real MEEG data

- **Outer criterion:** FDMC SURE[1]
- **Inner problems:** vanilla Lasso



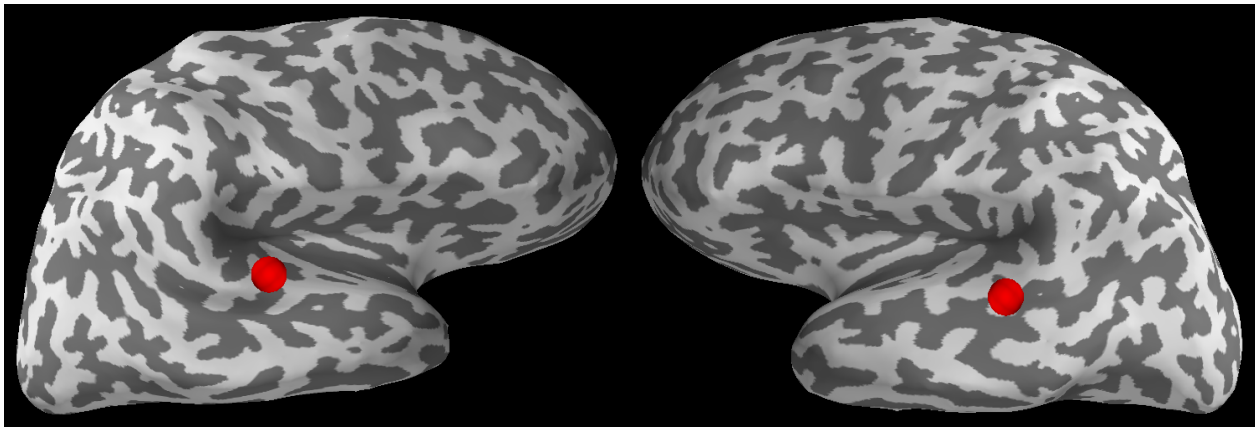
Real M/EEG data, vanilla Lasso (1 hyperparameter λ)

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

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

Experiments III - Real MEEG data

- **Outer criterion:** FDMC SURE[1]
- **Inner problems:** weighted Lasso



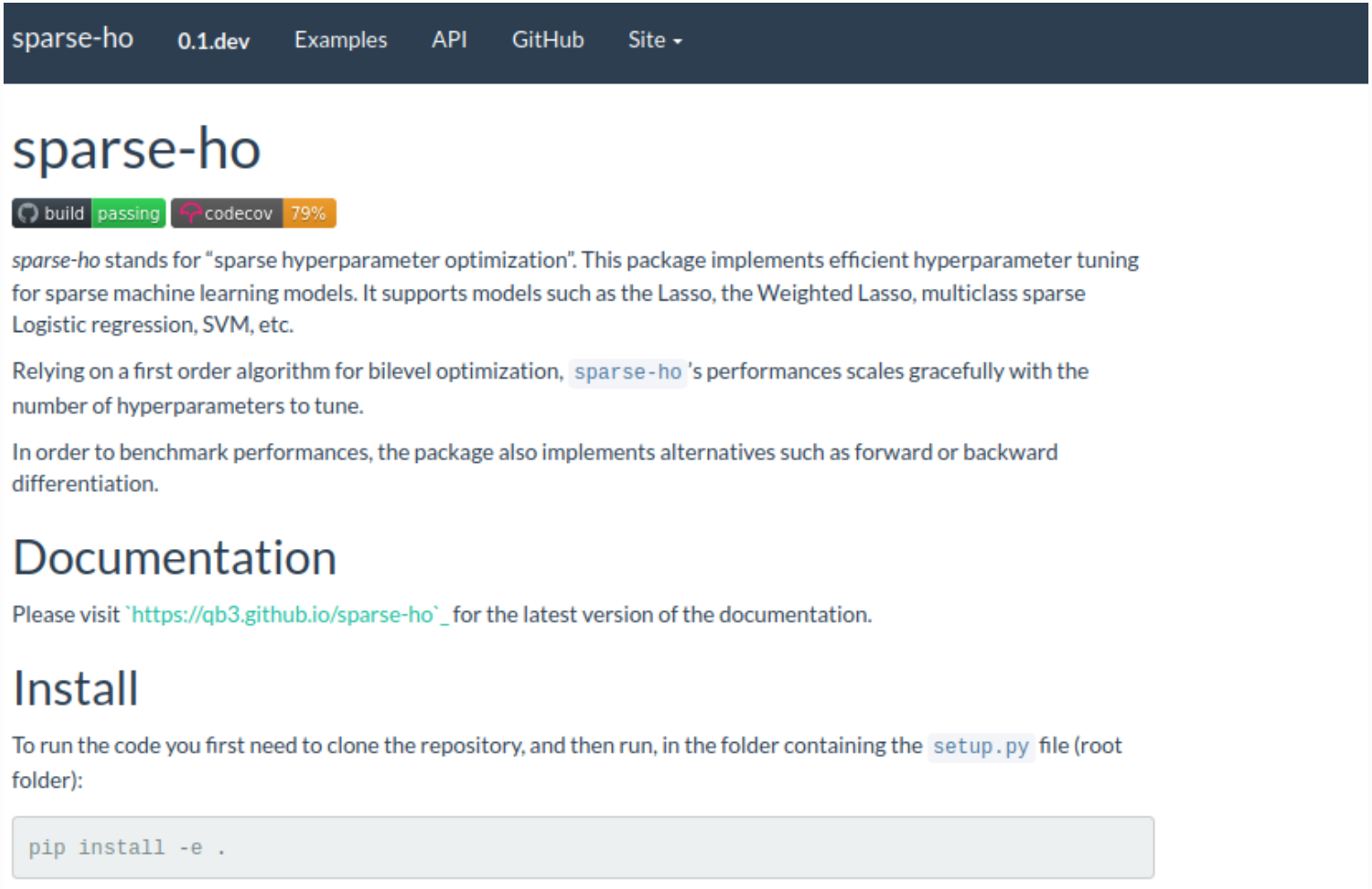
Real M/EEG data, weighted Lasso (p hyperparameters)

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

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

Contributions and dissemination

- **Local linear convergence** of the Jacobian
- **Leverage sparsity** to speed up hypergradient computation
- Open source package
  `https://github.com/QB3/sparse-ho`



sparse-ho 0.1.dev Examples API GitHub Site

sparse-ho

build passing codecov 79%

sparse-ho stands for "sparse hyperparameter optimization". This package implements efficient hyperparameter tuning for sparse machine learning models. It supports models such as the Lasso, the Weighted Lasso, multiclass sparse Logistic regression, SVM, etc.

Relying on a first order algorithm for bilevel optimization, `sparse-ho`'s performances scales gracefully with the number of hyperparameters to tune.

In order to benchmark performances, the package also implements alternatives such as forward or backward differentiation.

Documentation

Please visit `https://qb3.github.io/sparse-ho`_ for the latest version of the documentation.

Install

To run the code you first need to clone the repository, and then run, in the folder containing the `setup.py` file (root folder):

```
pip install -e .
```

Scientific publication

*"An article about computational science in a scientific publication is **not** the scholarship itself, it is merely **advertising** of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures."*

J. B. Buckheit and D. L. Donoho[1]

"All models are wrong, but some come with good open source implementation and good documentation: so use these."

A. Gramfort (circa 2015)

[1] J. B. Buckheit and D. L. Donoho (1995). "Wavelab and reproducible research". In: *Wavelets and statistics*. Springer, pp. 55–81.

How to reach me

Contact:

joseph.salmon@umontpellier.fr

http://josephsalmon.eu

Github: @josephsalmon

Twitter: @salmonjsph

References I

- Argyriou, A., T. Evgeniou, and M. Pontil (2008). "Convex multi-task feature learning". In: *Machine Learning*.
- Bengio, Y. (2000). "Gradient-based optimization of hyperparameters". In: *Neural computation*.
- Berger, H. (1929). "Über das elektroenkephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten*.
- Bergstra, J. and Y. Bengio (2012). "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research*.
- Bertrand, Q., Q. Klopfenstein, M. Blondel, et al. (2020). "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML*.
- Bertrand, Q., Q. Klopfenstein, M. Massias, et al. (2021). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*.
- Bertrand, Q. and M. Massias (2021). "Anderson acceleration of coordinate descent". In: *AISTATS*.

References II

- Bertrand, Q., M. Massias, et al. (2019). “Handling correlated and repeated measurements with the smoothed Multivariate square-root Lasso”. In: *NeurIPS*.
- Brochu, E., V. M. Cora, and N. D. Freitas (2010). “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*.
- Buckheit, J. B. and D. L. Donoho (1995). “Wavelab and reproducible research”. In: *Wavelets and statistics*. Springer, pp. 55–81.
- Cohen, D. (1968). “Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents”. In: *Science*.
- Deledalle, C.-A. et al. (2014). “Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection”. In: *SIAM J. Imaging Sci.*
- Figueiredo, M. (2001). “Adaptive Sparseness Using Jeffreys Prior.”. In: *Advances in Neural Information Processing Systems*.
- Gramfort, A., M. Kowalski, and M. Hämäläinen (2012). “Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods”. In: *Phys. Med. Biol.*

References III

- Hutter, F., J. Lücke, and L. Schmidt-Thieme (2015). "Beyond manual tuning of hyperparameters". In: *KI-Künstliche Intelligenz*.
- Kohavi, R. and G. H. John (1995). "Automatic parameter selection by minimizing estimated error". In: *Machine Learning Proceedings*.
- Liu, D. C. and J. Nocedal (1989). "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming*.
- Liu, W. and Y. Yang (2011). "Parametric or nonparametric? A parametricness index for model selection". In: *Ann. Statist.* 39.4, pp. 2074–2102.
- Lounici, K. (2008). "Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators". In: *Electron. J. Stat.*
- Lounici, K., K. Meziani, and B. Riu (2021). "Muddling Labels for Regularization, a novel approach to generalization". In: *arXiv preprint arXiv:2102.08769*.
- Lounici, K., M. Pontil, et al. (2009). "Taking Advantage of Sparsity in Multi-Task Learning". In: *arXiv preprint arXiv:0903.1468*.

References IV

- Martinet, B. (1970). “Brève communication. Régularisation d’inéquations variationnelles par approximations successives”. In: *Revue française d’informatique et de recherche opérationnelle. Série rouge* 4.R3, pp. 154–158.
- Moreau, J.-J. (1962). “Fonctions convexes duales et points proximaux dans un espace hilbertien”. In: *C. R. Acad. Sci. Paris* 255, pp. 2897–2899.
- Nocedal, J. and S. J. Wright (2006). *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. New York: Springer.
- Ochs, P. et al. (2015). “Bilevel optimization with nonsmooth lower level problems”. In: *SSVM*.
- Pedregosa, F. (2016). “Hyperparameter optimization with approximate gradient”. In: *ICML*.
- Stein, C. M. (1981). “Estimation of the mean of a multivariate normal distribution”. In: *Ann. Statist.* 9.6, pp. 1135–1151.
- Stone, L. R. A. and J. Ramer (1965). “Estimating WAIS IQ from Shipley Scale scores: Another cross-validation”. In: *Journal of clinical psychology* 21.3, pp. 297–297.

References V


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


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