Hyperparameter selection for high dimensional sparse learning

with neuroimaging motivations in mind

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

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

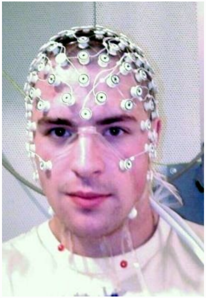
Pierre-Antoine

Mathieu

Samuel

Alexandre

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



(a) EEG

(b) MEG=Mag.+Grad.

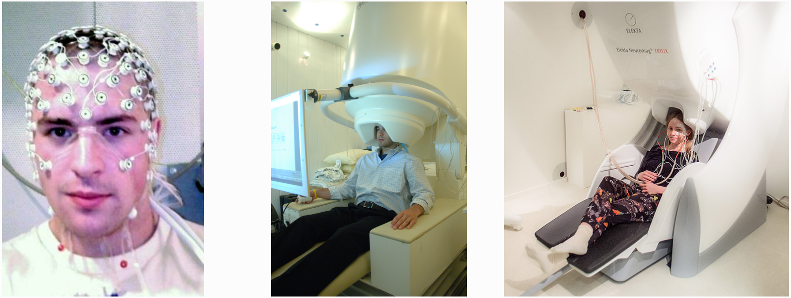
(c) M/EEG

<u>Photo credit</u>: 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

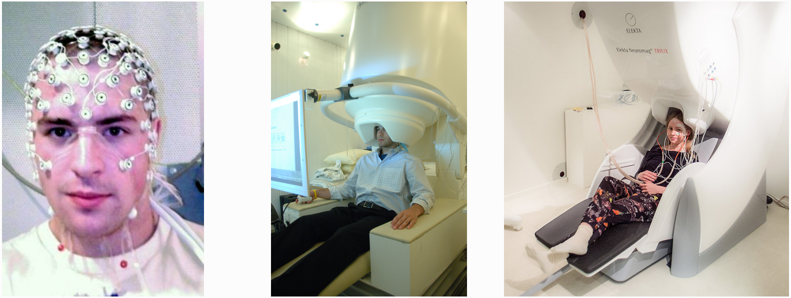
- **Data** Y: electric and magnetic fields at the head surface

---

[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

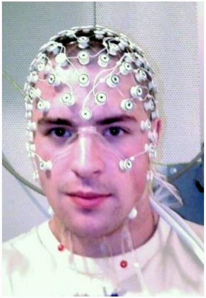
Photo credit: S. Whitmarsh

- **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

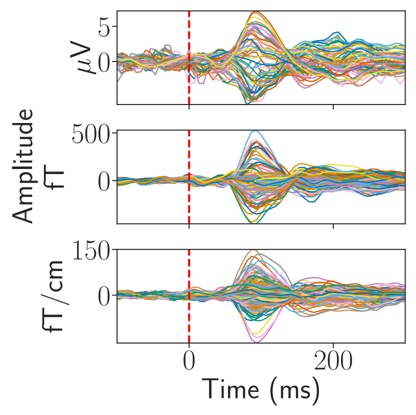
- **Data** Y: electric and magnetic fields at the head surface
- **Goal**: which parts of the brain are responsible for the signals?
- **Applications**: clinical and cognitive experiments

---

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

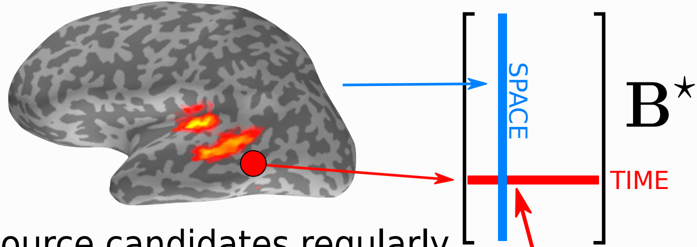
[2] D. Cohen (1968). "Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents". In: *Science*

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



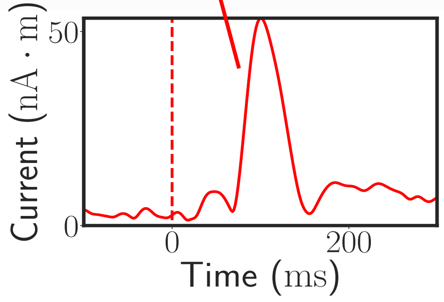
3 modalities:

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

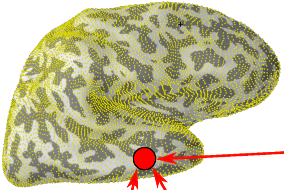


Source candidates regularly spaced in the brain (e.g., every 5mm)

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

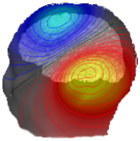


Design matrix - Forward operator

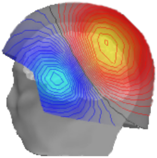


$$X = \begin{bmatrix} X_{\text{EEG}} \\ \hdashline X_{\text{MEG}} \end{bmatrix} \in \mathbb{R}^{n \times p}$$

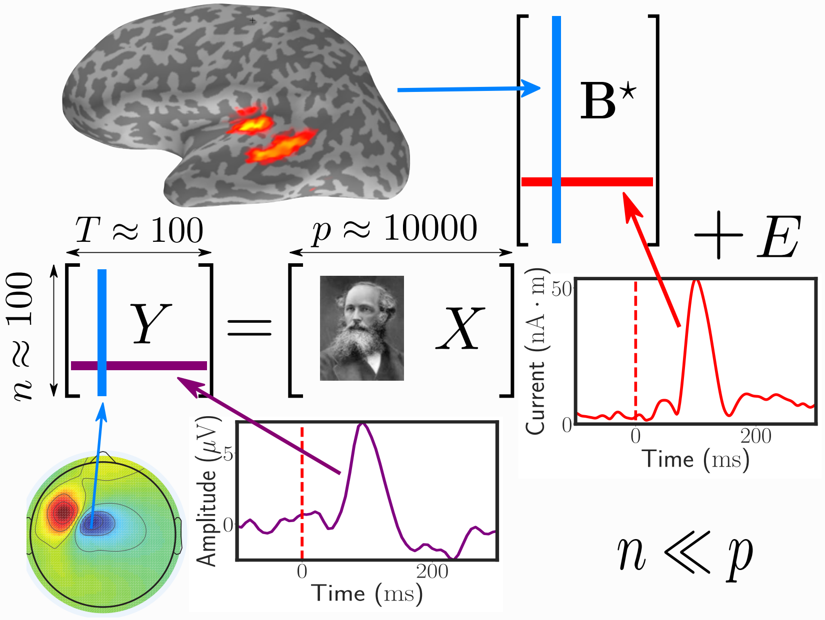
EEG:
Forward field of the electrodes



MEG:
Forward field of sensor



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

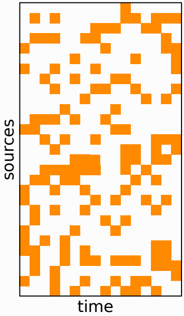


$\mathbf{B}^\star$
$+ E$
$T \approx 100$
$p \approx 10000$
$n \approx 100$
Y
$=$
X
Current $(\mathrm{nA \cdot m})$
50
0
0
200
Time (ms)
Amplitude (μV)
5
0
0
200
Time (ms)
$n \ll p$

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**

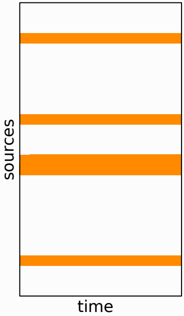
$$\boxed{\Omega(\mathrm{B}) = \|\mathrm{B}\|_1 = \sum_{j=1}^{p} \sum_{k=1}^{T} |\mathrm{B}_{j,k}|}$$

(1) A. Argyriou, T. Evgeniou, and M. Pontil (2008). “Convex multi-task feature learning”. In: *Machine Learning* 73.3, pp. 243–272

(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.* 57.7, pp. 1937–1961

Popular convex penalties: multitask Lasso (MTL)

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



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

Sparse support: group structure ✓

Penalty: **Group-Lasso**

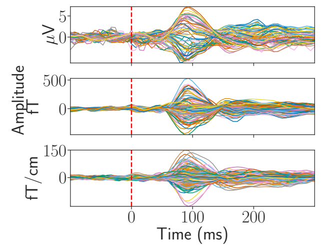
$$\boxed{\Omega(\mathrm{B}) = \|\mathrm{B}\|_{2,1} = \sum_{j=1}^{p} \|\mathrm{B}_{j,:}\|_2}$$

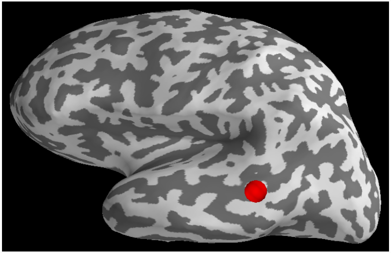
where $\mathrm{B}_{j,:}$ the j-th row of B

(1) A. Argyriou, T. Evgeniou, and M. Pontil (2008). “Convex multi-task feature learning”. In: *Machine Learning* 73.3, pp. 243–272

(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.* 57.7, pp. 1937–1961

Summary of the problem setting

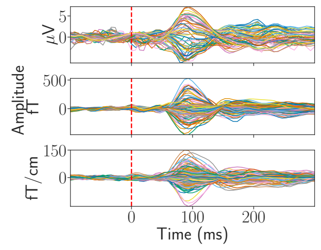


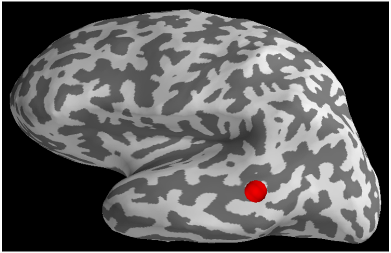


What you have: $Y \in \mathbb{R}^{n \times T}$

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

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

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

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

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*.

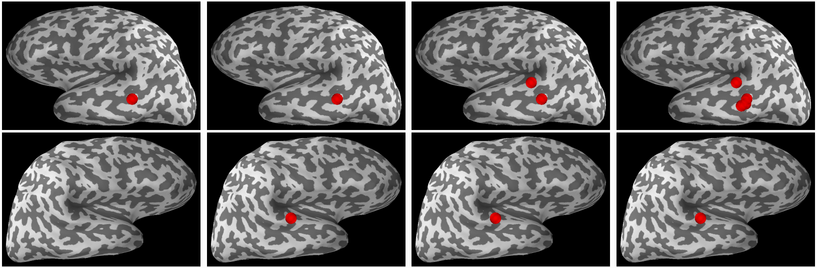
[(2)] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2022). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*.

[(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?

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



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

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

- Statistical route[(1),(2)]:
  use assumptions on X, provide guarantees but often conservative
- Bayesian statistics[(3),(4)]: prior on λ
- Bayesian optimization,[(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, pp. 90–102.

(2) K. Lounici, M. Pontil, et al. (2011). “Oracle inequalities and optimal inference under group sparsity”. In: *Ann. Statist.* 39.4, pp. 2164–2204.

(3) M. E. Tipping (2001). “Sparse Bayesian learning and the relevance vector machine”. In: *J. Mach. Learn. Res.* 1, pp. 211–244.

(4) M. Figueiredo (2001). “Adaptive Sparseness Using Jeffreys Prior”. In: *NIPS*, pp. 697–704.

(5) F. Hutter, J. Lücke, and L. Schmidt-Thieme (2015). “Beyond Manual Tuning of Hyperparameters”. In: *Künstliche Intell.* 29.4, pp. 329–337.

(6) R. Kohavi and G. H. John (1995). “Automatic parameter selection by minimizing estimated error”. In: *ICML*, pp. 304–312.

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

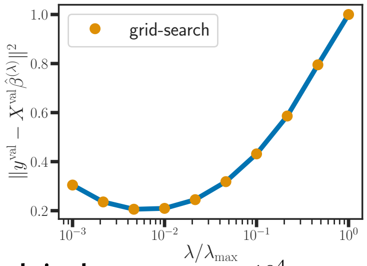
(2) K. Lounici, K. Meziani, and B. Riu (2021). "Muddling Labels for Regularization, a novel approach to generalization". In: *arXiv preprint arXiv:2102.08769*.

(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.

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

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

(1) L. R. A. Stone and J.C. 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*.

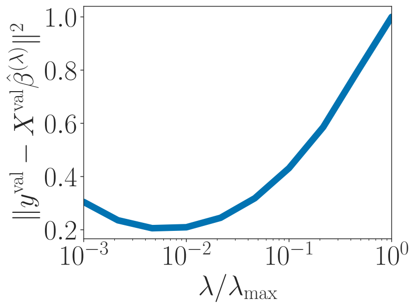
(3) W. Liu and Y. Yang (2011). “Parametric or nonparametric? A parametricness index for model selection”. In: *Ann. Statist.* 39.4, pp. 2074–2102.

(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)]

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

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



Grid search evaluation (sequential)

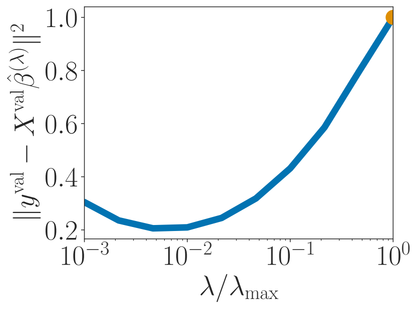
(1) P. Ochs et al. (2015). "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. vol. 9087, pp. 654–665

(2) F. Pedregosa (2016). "Hyperparameter optimization with approximate gradient". In: *ICML*. vol. 48, pp. 737–746

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

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

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



Grid search evaluation (sequential)

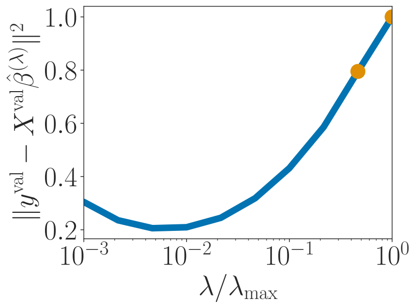
(1) P. Ochs et al. (2015). "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. vol. 9087, pp. 654–665

(2) F. Pedregosa (2016). "Hyperparameter optimization with approximate gradient". In: *ICML*. vol. 48, pp. 737–746

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

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

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



Grid search evaluation (sequential)

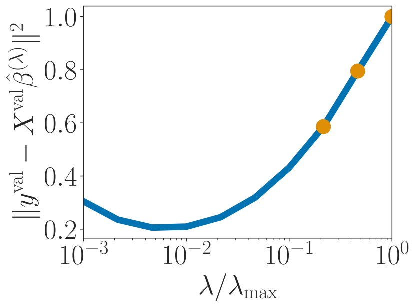
[1] P. Ochs et al. (2015). "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. vol. 9087, pp. 654–665

[2] F. Pedregosa (2016). "Hyperparameter optimization with approximate gradient". In: *ICML*. vol. 48, pp. 737–746

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

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

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



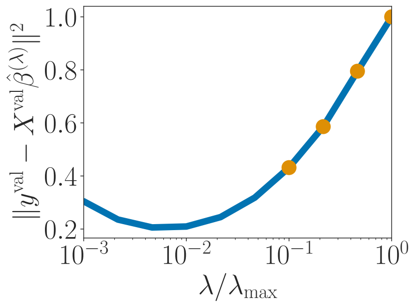
Grid search evaluation (sequential)

(1) P. Ochs et al. (2015). “Bilevel optimization with nonsmooth lower level problems”. In: *SSVM*. vol. 9087, pp. 654–665

(2) F. Pedregosa (2016). “Hyperparameter optimization with approximate gradient”. In: *ICML*. vol. 48, pp. 737–746

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

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



Grid search evaluation (sequential)

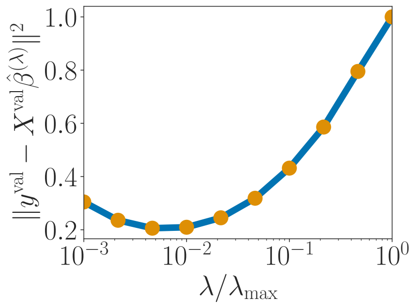
(1) P. Ochs et al. (2015). "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. vol. 9087, pp. 654–665

(2) F. Pedregosa (2016). "Hyperparameter optimization with approximate gradient". In: *ICML*. vol. 48, pp. 737–746

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

$$\underbrace{\arg\min_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := \| y^{\text{test}} - X^{\text{test}} \hat{\beta}^{(\lambda)} \|^2 \right\}}_{\text{outer optimization problem}}$$

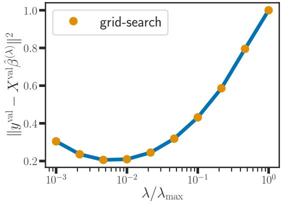
$$\text{s.t. } \underbrace{\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \| y^{\text{train}} - X^{\text{train}} \beta \|^2 + \lambda \|\beta\|_1}_{\text{inner optimization problem}}$$



Grid search evaluation (sequential)

(1) P. Ochs et al. (2015). "Bilevel optimization with nonsmooth lower level problems". In: *SSVM*. vol. 9087, pp. 654–665

(2) F. Pedregosa (2016). "Hyperparameter optimization with approximate gradient". In: *ICML*. vol. 48, pp. 737–746



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

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

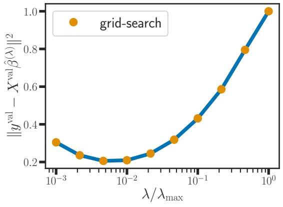
- Grid-search, random-search,[1] SMBO[2]:
  0-order methods to solve bilevel optimization problem

---

[1] J. Bergstra and Y. Bengio (2012). "Random search for hyper-parameter optimization". In: *J. Mach. Learn. Res.* 13.2.

[2] E. Brochu, V. M. Cora, and N. De Freitas (2010). *A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*. Tech. rep.

[3] F. Pedregosa (2016). "Hyperparameter optimization with approximate gradient". In: *ICML*. vol. 48, pp. 737–746.



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

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

- 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]:

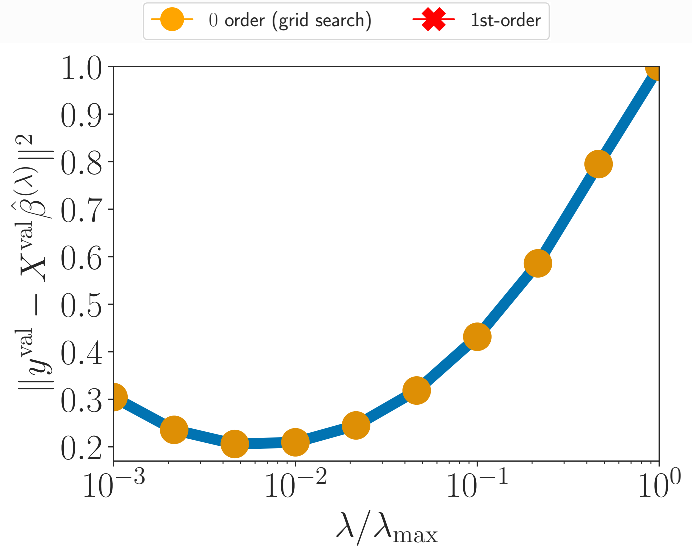
$$\lambda^{(t+1)} = \lambda^{(t)} - \rho\nabla_\lambda\mathcal{L}(\lambda^{(t)}) \quad \text{with } \rho > 0$$

(1) J. Bergstra and Y. Bengio (2012). “Random search for hyper-parameter optimization”. In: *J. Mach. Learn. Res.* 13.2.

(2) E. Brochu, V. M. Cora, and N. De Freitas (2010). *A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*. Tech. rep.

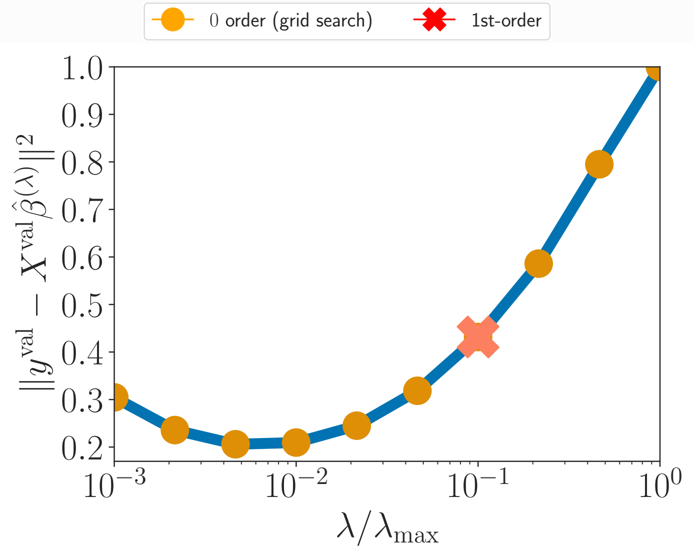
(3) F. Pedregosa (2016). “Hyperparameter optimization with approximate gradient”. In: *ICML*. vol. 48, pp. 737–746.

First-order optimization in λ, Lasso



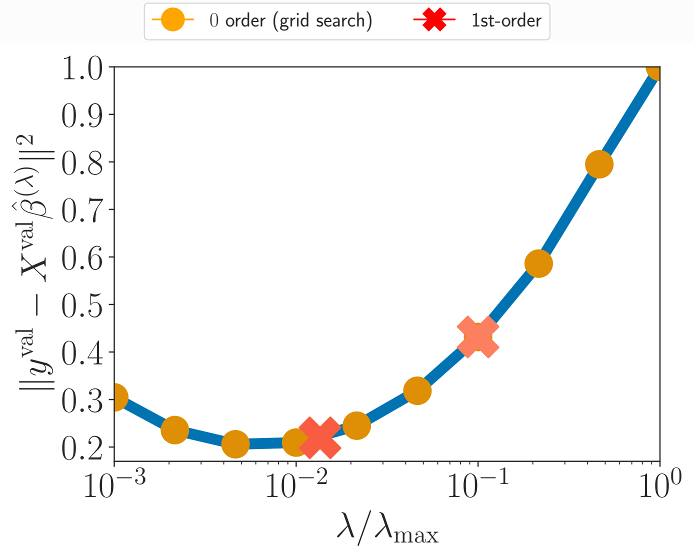
Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ.

First-order optimization in λ, Lasso



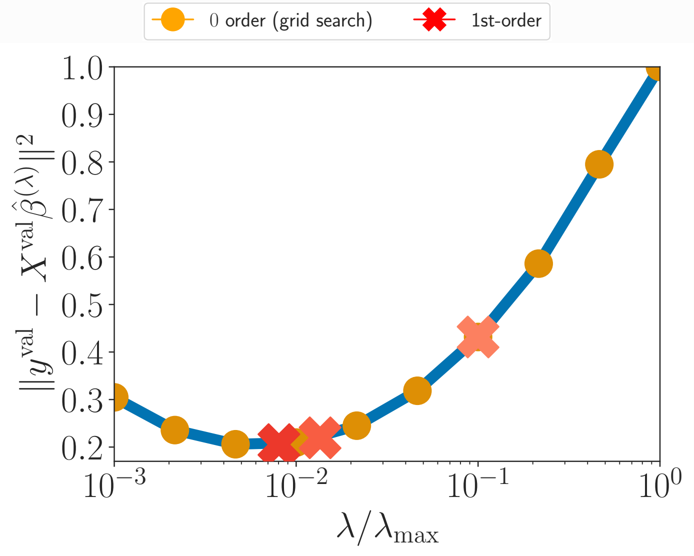
Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ.

First-order optimization in λ, Lasso



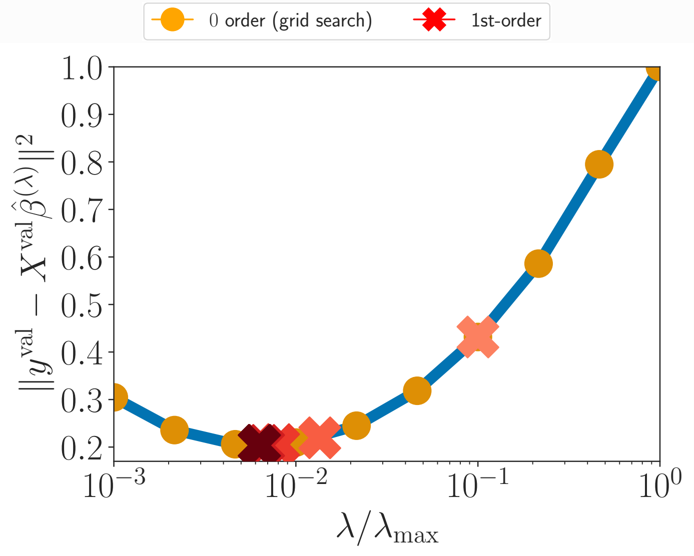
Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ.

First-order optimization in λ, Lasso



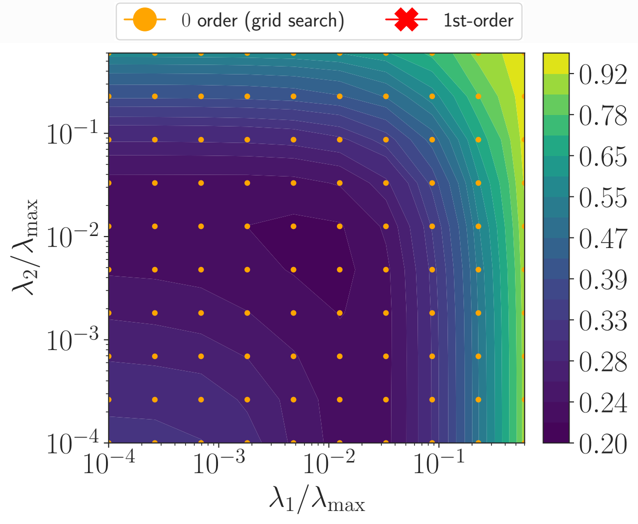
Real-sim dataset, $n \approx p \approx 10^4$. Validation loss as a function of λ.

First-order optimization in λ, Lasso



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

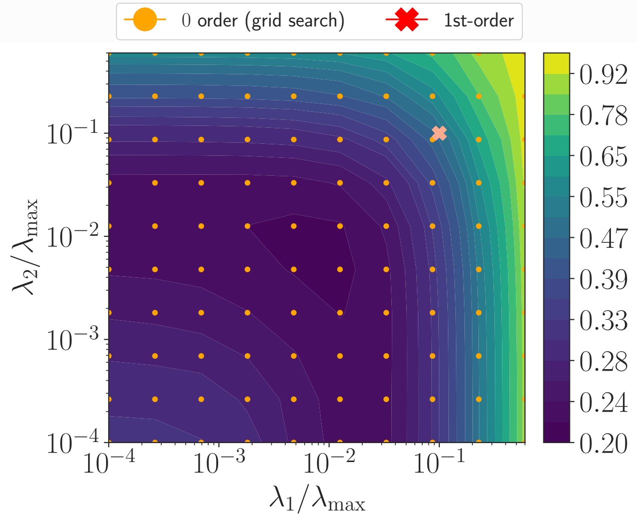
First-order optimization in λ, Enet



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

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

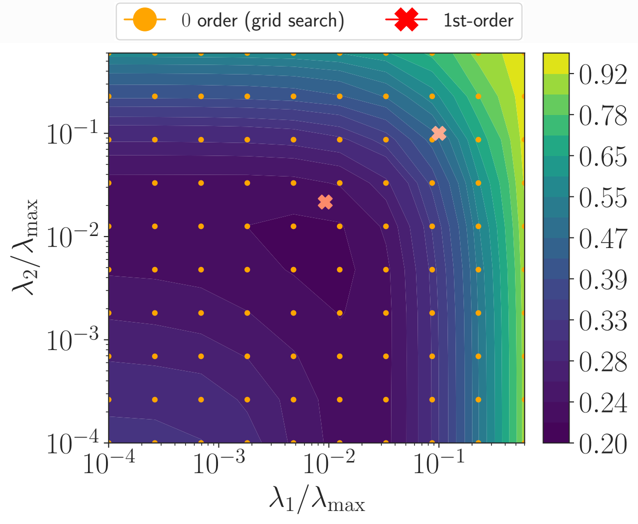
First-order optimization in λ, Enet



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

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

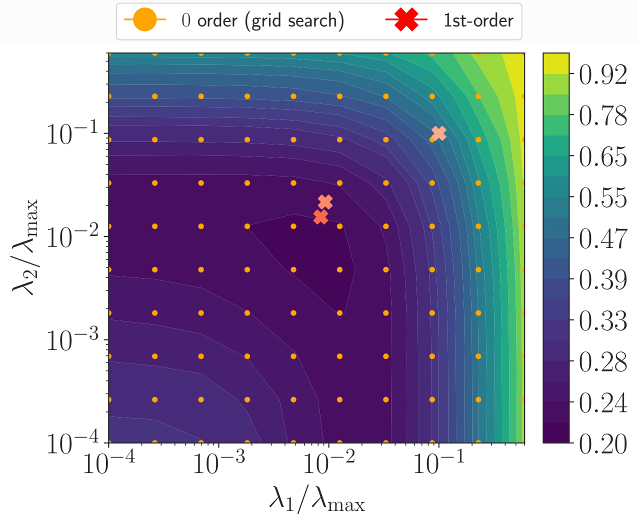
First-order optimization in λ, Enet



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

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

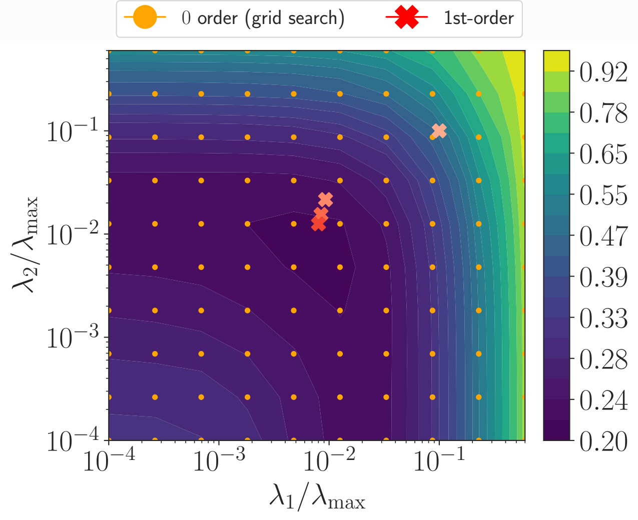
First-order optimization in λ, Enet



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

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

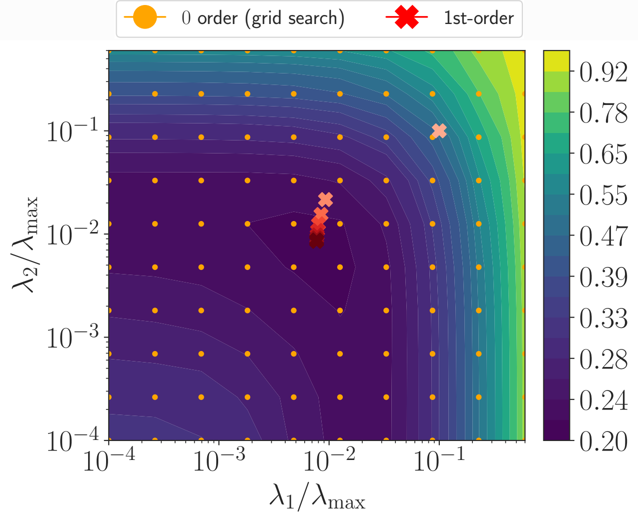
First-order optimization in λ, Enet



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 $\nabla_\lambda \mathcal{L}(\lambda)$

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

(1) J. Nocedal and S. J. Wright (2006). *Numerical optimization*. Second. Springer Series in Operations Research. Springer.

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

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

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

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

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. Springer.

(2) D. C. Liu and J. Nocedal (1989). “On the limited memory BFGS method for large scale optimization”. In: *Math. Program.*

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

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

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 $\lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix} \in \mathbb{R}^r$

(1) J. Nocedal and S. J. Wright (2006). *Numerical optimization*. Second. Springer Series in Operations Research. Springer.

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

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

$$\operatorname*{arg\,min}_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) := C(\hat{\beta}^{(\lambda)}) := \|y^{\text{test}} - X^{\text{test}} \hat{\beta}^{(\lambda)}\|^2 \right\}$$

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

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

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

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

Chain rule:

$$\nabla_\lambda \mathcal{L}(\lambda) = \underbrace{\hat{\mathcal{J}}^\top_{(\lambda)}}_{:= (\nabla_\lambda \hat\beta_1^{(\lambda)}, \dots, \nabla_\lambda \hat\beta_p^{(\lambda)})} \nabla_\beta C(\hat\beta^{(\lambda)})$$

$\rightarrow$ main technical challenge

- Boils down to:

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

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

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

- *Implicit differentiation* (**closed-form** formula)[1],[2]: need to solve a $p \times p$ linear system ($p = \#$features)
- *Automatic differentiation*, *forward*[3] or *reverse*[4] mode

Example:

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

(1) J. Larsen et al. (1996). "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*.

(2) Y. Bengio (2000). "Gradient-based optimization of hyperparameters". In: *Neural comput.* 12.8, pp. 1889–1900.

(3) L. Franceschi et al. (2017). "Forward and reverse gradient-based hyperparameter optimization". In: *ICML*, pp. 1165–1173.

(4) J. Domke (2012). "Generic methods for optimization-based modeling". In: *AISTATS*. vol. 22, pp. 318–326.

Optimization problem: $\boxed{\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda g(\beta)}$

Properties:

- f convex, gradient L-Lipschitz
- g convex but non necessarily smooth (can have kinks)

Example: $f(\beta) = \frac{1}{2n} \|X\beta - y\|^2, g(\beta) = \lambda \|\beta\|_1$

Optimization problem: $\boxed{\min_{\beta \in \mathbb{R}^p} f(\beta) + \lambda g(\beta)}$

Properties:

- f convex, gradient L-Lipschitz
- g convex but non necessarily smooth (can have kinks)

Example: $f(\beta) = \frac{1}{2n} \|X\beta - y\|^2, g(\beta) = \lambda \|\beta\|_1$

Rem: fix step size (sub-)gradient descent does not converge: take $f = 0$, $g = |\cdot|$ and use $\beta_0 = 1/2, \alpha = 1$ (ping-pong!)

J.-J. Moreau

Properties:

- g (convex)
- its **proximal** operator[1] has a closed-form:

$$\boxed{\operatorname{prox}_{\lambda g}\left(\beta^0\right) := \underset{\beta\in\mathbb{R}^p}{\arg\min}\ \frac{1}{2}\|\beta-\beta^0\|^2+\lambda g(\beta)}$$

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

(2) N. Parikh and S. Boyd (2014). "Proximal Algorithms". In: *Foundations and Trends in Machine Learning* 1.3, pp. 127–239.

(3) H. H. Bauschke and P. L. Combettes (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, pp. xvi+468.

J.-J. Moreau

Properties:

- g (convex)
- its **proximal** operator[1] has a closed-form:

$$\boxed{\operatorname{prox}_{\lambda g}\left(\beta^0\right) := \underset{\beta \in \mathbb{R}^p}{\arg\min}\ \frac{1}{2}\|\beta - \beta^0\|^2 + \lambda g(\beta)}$$

Majorizer (at step t): $\beta \mapsto f(\beta^t) + \langle \nabla f(\beta^t), \beta - \beta^t \rangle + \frac{L\|\beta^t - \beta\|^2}{2} + \lambda g(\beta)$

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

(2) N. Parikh and S. Boyd (2014). "Proximal Algorithms". In: *Foundations and Trends in Machine Learning* 1.3, pp. 127–239.

(3) H. H. Bauschke and P. L. Combettes (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, pp. xvi+468.

Properties:

- g (convex)
- its **proximal** operator[1] has a closed-form:

$$\boxed{\operatorname{prox}_{\lambda g}\left(\beta^0\right) := \underset{\beta \in \mathbb{R}^p}{\arg\min}\ \frac{1}{2}\|\beta - \beta^0\|^2 + \lambda g(\beta)}$$

J.-J. Moreau

Majorizer (at step t): $\beta \mapsto f(\beta^t) + \langle \nabla f(\beta^t), \beta - \beta^t \rangle + \frac{L\|\beta^t - \beta\|^2}{2} + \lambda g(\beta)$

Update rule (minimize the majorizer) : $\boxed{\beta^{t+1} = \operatorname{prox}_{\frac{\lambda g}{L}}\left(\beta^t - \frac{1}{L}\nabla f(\beta^t)\right)}$

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

(2) N. Parikh and S. Boyd (2014). "Proximal Algorithms". In: *Foundations and Trends in Machine Learning* 1.3, pp. 127–239.

(3) H. H. Bauschke and P. L. Combettes (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, pp. xvi+468.

J.-J. Moreau

Properties:

- g (convex)
- its **proximal** operator[1] has a closed-form:

$$\boxed{\mathrm{prox}_{\lambda g}\left(\beta^0\right) := \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|\beta - \beta^0\|^2 + \lambda g(\beta)}$$

Majorizer (at step t): $\beta \mapsto f(\beta^t) + \langle \nabla f(\beta^t), \beta - \beta^t \rangle + \frac{L\|\beta^t - \beta\|^2}{2} + \lambda g(\beta)$

Update rule (minimize the majorizer) : $\boxed{\beta^{t+1} = \mathrm{prox}_{\frac{\lambda g}{L}}\left(\beta^t - \frac{1}{L}\nabla f(\beta^t)\right)}$

- Proximal algorithms / recipes[2]
- Associated theory / analysis[3]

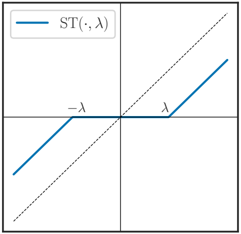
(1) Jean-Jacques Moreau (1962). “Fonctions convexes duales et points proximaux dans un espace hilbertien”. In: *C. R. Acad. Sci. Paris* 255, pp. 2897–2899.

(2) N. Parikh and S. Boyd (2014). “Proximal Algorithms”. In: *Foundations and Trends in Machine Learning* 1.3, pp. 127–239.

(3) H. H. Bauschke and P. L. Combettes (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, pp. xvi+468.

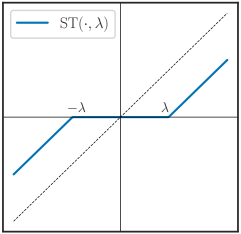
- **Soft-Thresholding**
  Closed form solution for the prox of $|\cdot|$:

$$
\begin{aligned}
\operatorname{prox}_{\lambda|\cdot|}(\beta^0) = \mathrm{ST}(\beta^0, \lambda) \\
&:= \arg\min_{\beta \in \mathbb{R}} \left(\frac{1}{2}(\beta^0 - \beta)^2 + \lambda|\beta| \right) \\
&= \operatorname{sign}(\beta^0) \cdot \max(0, |\beta^0| - \lambda)
\end{aligned}
$$



- **Soft-Thresholding**
  Closed form solution for the prox of $|\cdot|$:

$$\begin{aligned}
\operatorname{prox}_{\lambda|\cdot|}(\beta^0) &= \mathrm{ST}(\beta^0, \lambda) \\
&:= \operatorname*{arg\,min}_{\beta \in \mathbb{R}} \left(\frac{1}{2}(\beta^0 - \beta)^2 + \lambda|\beta| \right) \\
&= \operatorname{sign}(\beta^0) \cdot \max(0, |\beta^0| - \lambda)
\end{aligned}$$



- Componentwise **Soft-Thresholding**
  Closed form solution for the $\|\cdot\|_1$:

$$\left[\operatorname{prox}_{\lambda\|\cdot\|_1}(\beta^0)\right]_j = \left[\mathrm{ST}(\beta^0, \lambda)\right]_j , \qquad \text{for all } j \in [p]$$

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

Algorithm: Proximal gradient descent PGD

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

for $\text{iter} = 1, \dots,$ **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.* 7.4

[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

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

Algorithm: Forward-mode differentiation of PGD

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

for $\text{iter} = 1, \dots,$ **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.* 7.4

(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] of PGD[2]

Proximal Gradient Descent (PGD)[3]

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

Algorithm: Forward-mode differentiation of PGD

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

for iter $= 1, \dots,$ **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`

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

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

return β, $\mathcal{J}$

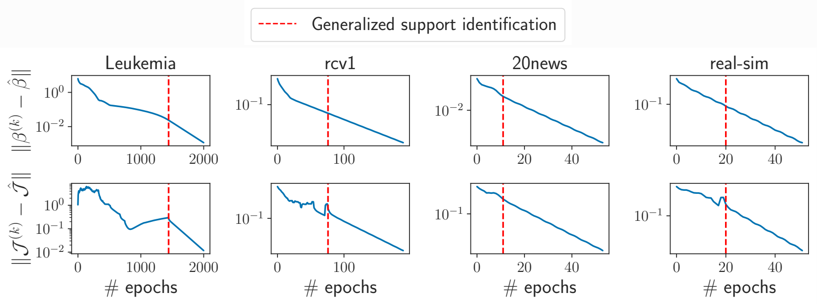
(1) R. E. Wengert (1964). “A simple automatic derivative evaluation program”. In: *Communications of the ACM* 7.8, pp. 463–464

(2) C.-A. Deledalle et al. (2014). “Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection”. In: *SIAM J. Imaging Sci.* 7.4

(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 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 g)[1]

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

[1] Y. Bengio (2000). "Gradient-based optimization of hyperparameters". In: *Neural comput.* 12.8, pp. 1889–1900

Implicit differentiation (smooth g) [1]

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

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

(1) Y. Bengio (2000). "Gradient-based optimization of hyperparameters". In: *Neural comput.* 12.8, pp. 1889–1900

Implicit differentiation (smooth g) [1]

$$\hat{\beta}^{(\lambda)} \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} f(\beta, \lambda)$$

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

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

(1) Y. Bengio (2000). "Gradient-based optimization of hyperparameters". In: *Neural comput.* 12.8, pp. 1889–1900

Implicit differentiation (smooth g)[1]

$$\hat{\beta}^{(\lambda)} \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} f(\beta, \lambda)$$

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

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

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

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

(1) Y. Bengio (2000). "Gradient-based optimization of hyperparameters". In: *Neural comput.* 12.8, pp. 1889–1900

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. (2022). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*

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

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

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

[1] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2022). “Implicit differentiation for fast hyperparameter selection in non-smooth convex learning”. In: *Submitted to JMLR*

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

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

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

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

[1] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2022). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*

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

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

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

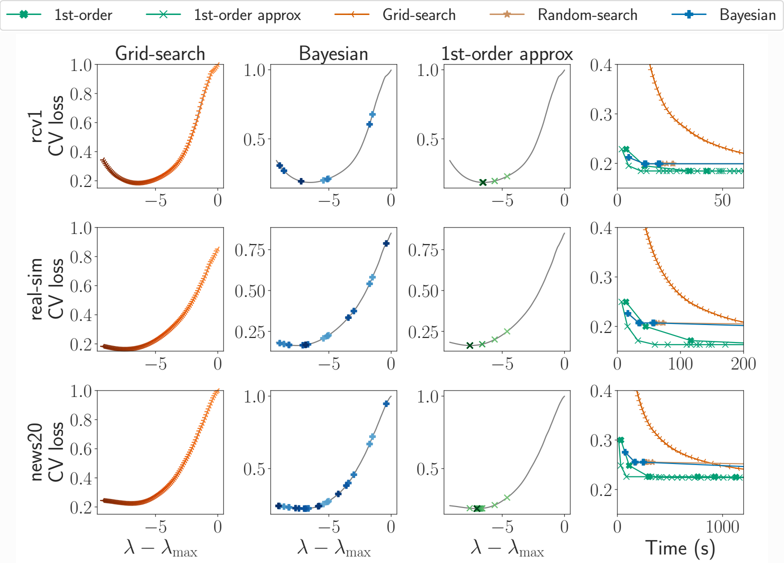
$$\partial_\beta \,\mathrm{ST}\left(\hat\beta_j^{(\lambda)} - \frac{1}{L}\nabla_j f(\hat\beta^{(\lambda)}), \frac{\lambda}{L}\right) = 0; 0 = \partial_\lambda \,\mathrm{ST}\left(\hat\beta_j^{(\lambda)} - \frac{1}{L}\nabla_j f(\hat\beta^{(\lambda)}), \frac{\lambda}{L}\right)$$

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

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

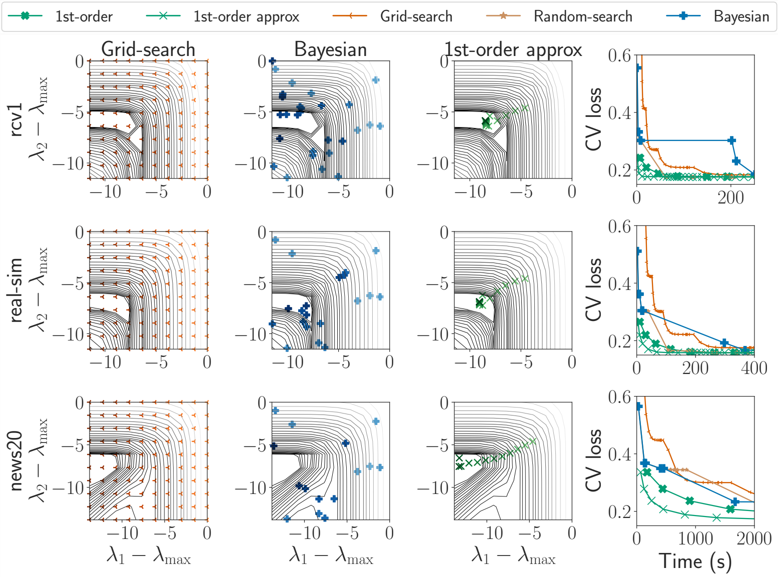
[1] Q. Bertrand, Q. Klopfenstein, M. Massias, et al. (2022). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*

Experiments I - Lasso cross-validation



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

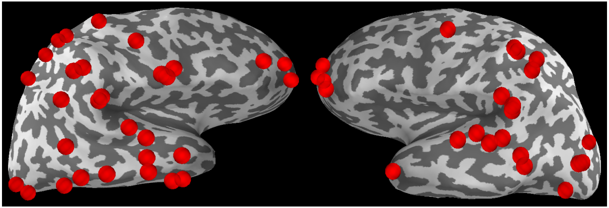
Experiments II - Enet cross-validation



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

Experiments III - Real MEEG data

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

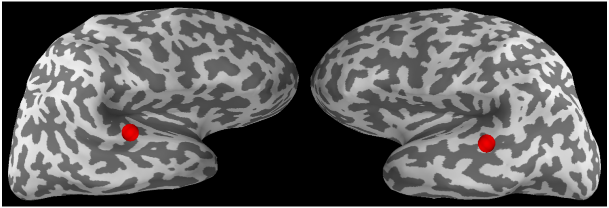


Real M/EEG data, vanilla Lasso (1 hyperparameter λ)

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

(1) C.-A. Deledalle et al. (2014). "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J. Imaging Sci.* 7.4.

- **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.* 7.4.

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 .
```

- Specific parametrization ($\lambda \leftarrow e^{\lambda}$ for simplicity)
- Need a **differentiable criterion**: cannot use 0/1-loss
- Need a **continuous estimator** *w.r.t.* data and hyperparameters: does not apply yet to **non-convex** penalties[1] nor reweighted Lasso[2]
- Optimized function often **non-convex**:
possibly multiple local minima, use multi-starts
- Potentially slow and handy **outer solver**

(1) P. Breheny and J. Huang (2011). "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection". In: *Ann. Appl. Stat.* 5.1, p. 232.

(2) E. J. Candès, M. B. Wakin, and S. P. Boyd (2008). "Enhancing Sparsity by Reweighted l_1 Minimization". In: *J. Fourier Anal. Applicat.* 14.5-6, pp. 877–905.

Conclusion

Contributions:

- 1st-order optimization with nonsmooth inner problem
- **Local linear convergence** of the Jacobian
- **Leverage sparsity** to speed up hypergradient computation

Conclusion

Contributions:

- 1st-order optimization with nonsmooth inner problem
- **Local linear convergence** of the Jacobian
- **Leverage sparsity** to speed up hypergradient computation

Future work:

- Convergence of the bilevel procedure
- Smarter outer solver

Conclusion

Contributions:

- 1st-order optimization with nonsmooth inner problem
- **Local linear convergence** of the Jacobian
- **Leverage sparsity** to speed up hypergradient computation

Future work:

- Convergence of the bilevel procedure
- Smarter outer solver

Contact information and citations

*"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)

Contact:

joseph.salmon@umontpellier.fr

http://josephsalmon.eu

Github: @josephsalmon

Twitter: @salmonjsph

[1] J. B. Buckheit and D. L. Donoho (1995). "WaveLab and Reproducible Research". In: *Wavelets and Statistics*. Lect. Notes Statist. 103. Springer-Verlag, pp. 55–81.

References I

- Argyriou, A., T. Evgeniou, and M. Pontil (2008). "Convex multi-task feature learning". In: *Machine Learning* 73.3, pp. 243–272.
- Bauschke, H. H. and P. L. Combettes (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. New York: Springer, pp. xvi+468.
- Bengio, Y. (2000). "Gradient-based optimization of hyperparameters". In: *Neural comput.* 12.8, pp. 1889–1900.
- 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: *J. Mach. Learn. Res.* 13.2.
- 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. (2022). "Implicit differentiation for fast hyperparameter selection in non-smooth convex learning". In: *Submitted to JMLR*.

References II

- Bertrand, Q. and M. Massias (2021). "Anderson acceleration of coordinate descent". In: *AISTATS*.
- Bertrand, Q., M. Massias, et al. (2019). "Handling correlated and repeated measurements with the smoothed multivariate square-root Lasso". In: *NeurIPS*.
- Breheny, P. and J. Huang (2011). "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection". In: *Ann. Appl. Stat.* 5.1, p. 232.
- Brochu, E., V. M. Cora, and N. De Freitas (2010). *A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*. Tech. rep.
- Buckheit, J. B. and D. L. Donoho (1995). "WaveLab and Reproducible Research". In: *Wavelets and Statistics*. Lect. Notes Statist. 103. New York: Springer-Verlag, pp. 55–81.
- Candès, E. J., M. B. Wakin, and S. P. Boyd (2008). "Enhancing Sparsity by Reweighted l_1 Minimization". In: *J. Fourier Anal. Applicat.* 14.5-6, pp. 877–905.

References III

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.* 7.4.

Domke, J. (2012). “Generic methods for optimization-based modeling”. In: *AISTATS*. Vol. 22, pp. 318–326.

Figueiredo, M. (2001). “Adaptive Sparseness Using Jeffreys Prior”. In: *NIPS*, pp. 697–704.

Franceschi, L. et al. (2017). “Forward and reverse gradient-based hyperparameter optimization”. In: *ICML*, pp. 1165–1173.

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.* 57.7, pp. 1937–1961.

Hutter, F., J. Lücke, and L. Schmidt-Thieme (2015). “Beyond Manual Tuning of Hyperparameters”. In: *Künstliche Intell.* 29.4, pp. 329–337.

Kohavi, R. and G. H. John (1995). “Automatic parameter selection by minimizing estimated error”. In: *ICML*, pp. 304–312.

- Larsen, J. et al. (1996). "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.*
- Liu, D. C. and J. Nocedal (1989). "On the limited memory BFGS method for large scale optimization". In: *Math. Program.*
- 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.* 2, pp. 90–102.
- 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. (2011). "Oracle inequalities and optimal inference under group sparsity". In: *Ann. Statist.* 39.4, pp. 2164–2204.

- 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, Jean-Jacques (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. New York: Springer.
- Ochs, P. et al. (2015). “Bilevel optimization with nonsmooth lower level problems”. In: *SSVM*. Vol. 9087, pp. 654–665.
- Parikh, N. and S. Boyd (2014). “Proximal Algorithms”. In: *Foundations and Trends in Machine Learning* 1.3, pp. 127–239.
- Pedregosa, F. (2016). “Hyperparameter optimization with approximate gradient”. In: *ICML*. Vol. 48, pp. 737–746.
- Stein, C. M. (1981). “Estimation of the mean of a multivariate normal distribution”. In: *Ann. Statist.* 9.6, pp. 1135–1151.

References VI

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

Tipping, M. E. (2001). “Sparse Bayesian learning and the relevance vector machine”. In: *J. Mach. Learn. Res.* 1, pp. 211–244.

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