Safe Grid Search with Optimal Complexity

Joseph Salmon
http://josephsalmon.eu
IMAG, Univ Montpellier, CNRS
Montpellier, France

Joint work with:
E. Ndiaye (RIKEN, Nagoya)
T. Le (RIKEN, Tokyo)
O. Fercoq (Institut Polytechnique de Paris)
I. Takeuchi (Nagoya Institute of Technology)

Simplest model: standard sparse regression

$y \in \mathbb{R}^n$: a signal

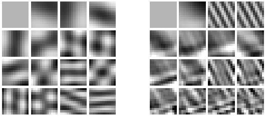
$X = [\mathbf{x}_1, \dots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$: **dictionary** of atoms/features

<u>Assumption</u> : signal well approximated by a **sparse** combination $\beta^* \in \mathbb{R}^p$: $y \approx X\beta^*$

<u>Objective(s)</u>: find $\hat{\beta}$

- Estimation: $\hat{\beta} \approx \hat{\beta}^*$
- Prediction: $X\hat{\beta} \approx X\hat{\beta}^*$
- Support recovery: $\operatorname{supp}(\hat{\beta}) \approx \operatorname{supp}(\beta^*)$

<u>Constraints</u>: large p, sparse β^*



$$\underbrace{\begin{bmatrix} y \end{bmatrix}}_{y \in \mathbb{R}^n} \approx \underbrace{\begin{bmatrix} \mathbf{x}_1 & \Big| & \dots & \Big| & \mathbf{x}_p \end{bmatrix}}_{X \in \mathbb{R}^{n \times p}} \cdot \underbrace{\begin{bmatrix} \beta_1^* \\ \vdots \\ \beta_p^* \end{bmatrix}}_{\beta \in \mathbb{R}^p}$$

$$y \approx \sum_{j=1}^{p} \beta_j^* \mathbf{x}_j$$

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least squares" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\text{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\text{sparsity-inducing penalty}} \right)$$

- Solutions are **sparse** (sparsity level controlled by λ)

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least squares" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\hat{\beta}^{(\lambda)} \in \mathop{\arg\min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\textbf{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\textbf{sparsity-inducing penalty}} \right)$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least squares" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\hat{\beta}^{(\lambda)} \in \mathop{\arg\min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\text{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\text{sparsity-inducing penalty}} \right)$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)
- Theoretical guaranties Bickel *et al.* (2009)

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least squares" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\hat{\beta}^{(\lambda)} \in \mathop{\arg\min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\textbf{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\textbf{sparsity-inducing penalty}} \right)$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)
- Theoretical guaranties Bickel *et al.* (2009)
- Refinements: non-convex approaches Adaptive Lasso Zou (2006), scaled invariance Sun and Zhang (2012), etc.

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least squares" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\hat{\beta}^{(\lambda)} \in \mathop{\arg\min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\textbf{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\textbf{sparsity-inducing penalty}} \right)$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)
- Theoretical guaranties Bickel *et al.* (2009)
- Refinements: non-convex approaches Adaptive Lasso Zou (2006), scaled invariance Sun and Zhang (2012), etc.

Well... many Lassos are needed

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

In practice:

- Step 1 compute T solutions on a grid, *i.e.*, compute $\beta^{(\lambda_0)}, \dots, \beta^{(\lambda_{T-1})}$ approximating $\hat{\beta}^{(\lambda_0)}, \dots, \hat{\beta}^{(\lambda_{T-1})}$, for some $\lambda_0 > \cdots > \lambda_{T-1}$
- Step 2 pick the "best" parameter

Questions:

- performance criterion: how to pick a "best" λ?
  - cross-validation (and variant)
  - SURE (Stein Unbiased Risk Estimation)
  - etc.
- grid choice: how to design the grid itself?

In practice: who does what?

Standard grid: (R-glmnet / Python-sklearn): **geometric** grid

- $\lambda_0 = \lambda_{\max} := \|X^\top y\|_\infty = \max_{j=1}^{p} \langle \mathbf{x}_j, y \rangle$ (critical value)
- $\lambda_t = \lambda_{\max} \times 10^{-\delta t/(T-1)}$, $T = 100$ and $\delta = 3$
- $\lambda_{T-1} = \lambda_{\max}/10^3 := \lambda_{\min}$

Parameter's choice:

`Python-sklearn` : vanilla 5-fold Cross-Validation, get smallest mean squared error (averaged over folds)

`R-glmnet` : vanilla 10-fold Cross-Validation, get largest λ such that the error is smaller than the mean squared error (averaged over folds) $+$ 1 standard deviation

Hold-out cross-validation

From now on : **hold-out cross-validation** (one single split)

Standard choice: 80 % train (n_{train}), 20 % test (n_{test})

- $X = X_{\text{train}} \cup X_{\text{test}}$
- $y = y_{\text{train}} \cup y_{\text{test}}$
- Change the error on test (validation):

$$E_{\text{test}}(\hat{\beta}^{(\lambda)}) = \mathcal{L}(y_{\text{test}},\ X_{\text{test}}\hat{\beta}^{(\lambda)}) := \left\| y_{\text{test}} - X_{\text{test}}\hat{\beta}^{(\lambda)} \right\|$$
$$\left(\text{or } \left\| y_{\text{test}} - X_{\text{test}}\hat{\beta}^{(\lambda)} \right\|^2\right)$$

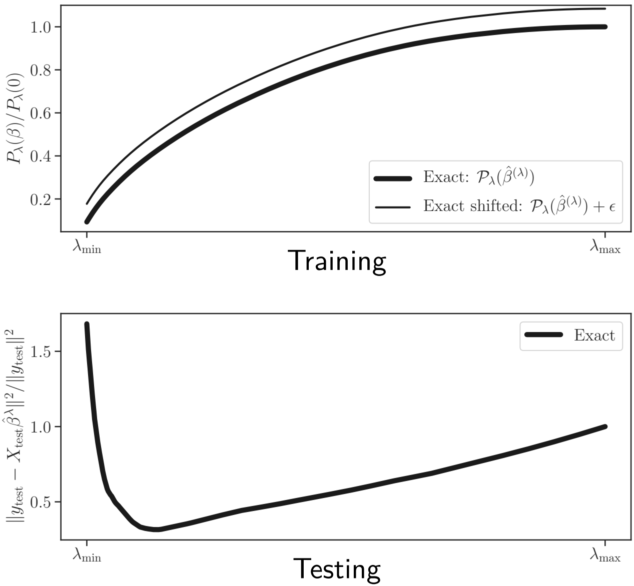
Some practical examples

- `leukemia`[1]: $n = 72, p = 7129$ (genes expression) y (binary) measure of disease
- `diabetes`[2]: $n = 442, p = 10$ (Age, Sex, Body mass index, Average blood pressure, S1, S2, S3, S4, S5, S6) y a quantitative measure of disease progression one year after baseline

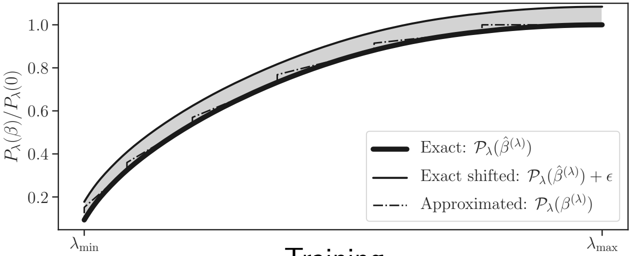
(1) https://sklearn.org/modules/generated/sklearn.datasets.fetch_mldata.html

(2) https://scikit-learn.org/stable/datasets/index.html#diabetes-dataset

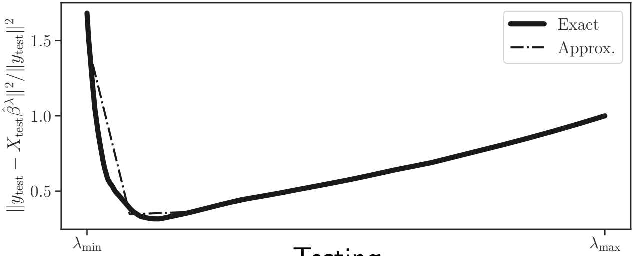
Example: Training / Testing (`leukemia`)



Example: Training / Testing (`leukemia`)

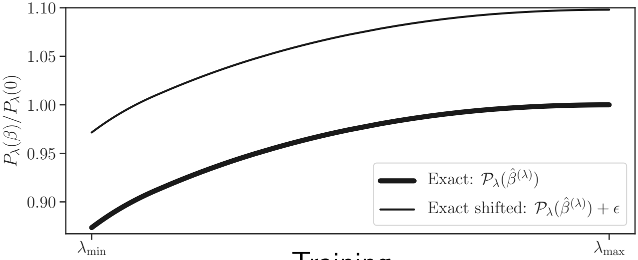


Training

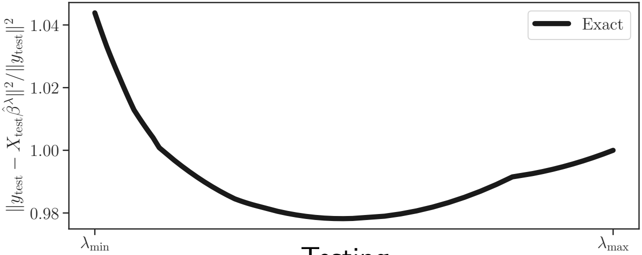


Testing

Example: Training / Testing (diabetes)

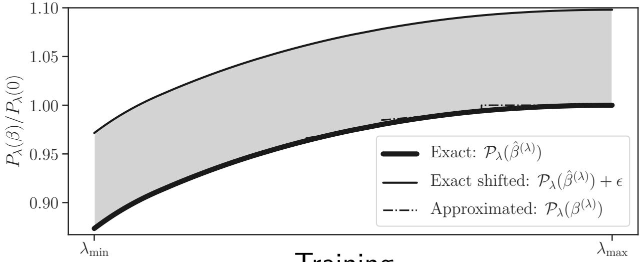


Training

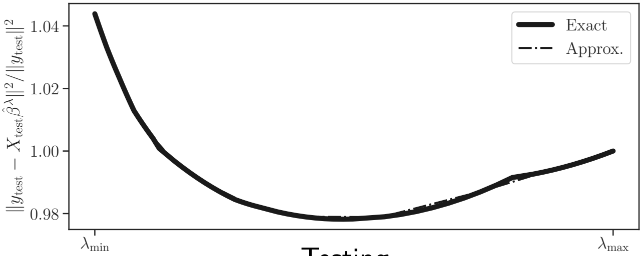


Testing

Example: Training / Testing (diabetes)



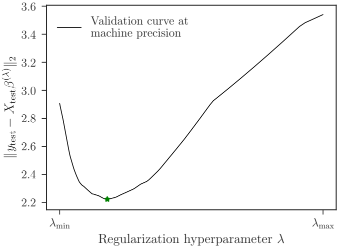
Training

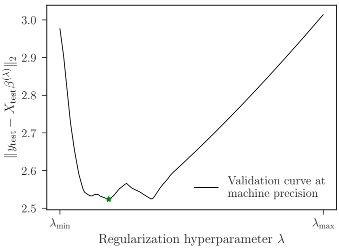


Testing

Hyperparameter tuning

- Learning Task: $\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \underbrace{f(X_{\text{train}}\beta)}_{\frac{1}{2}\|X_{\text{train}}\beta - y_{\text{train}}\|^2} + \lambda \underbrace{\Omega(\beta)}_{\|\beta\|_1}$
- Evaluation: $E_{\text{test}}(\hat{\beta}^{(\lambda)}) = \mathcal{L}(y_{\text{test}},\ X_{\text{test}}\hat{\beta}^{(\lambda)})$

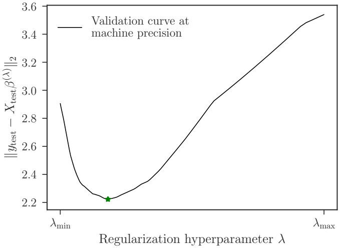


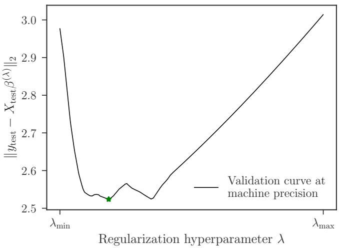


How to choose the grid of hyperparameter?

Hyperparameter tuning

- Learning Task: $$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \underbrace{f(X_{\text{train}}\beta)}_{\frac{1}{2}\|X_{\text{train}}\beta - y_{\text{train}}\|^2} + \lambda \underbrace{\Omega(\beta)}_{\|\beta\|_1}$$
- Evaluation: $$E_{\text{test}}(\hat{\beta}^{(\lambda)}) = \mathcal{L}(y_{\text{test}},\ X_{\text{test}}\hat{\beta}^{(\lambda)})$$





How to choose the grid of hyperparameter?

Hyperparameter tuning as bilevel optimization

The "optimal" hyperparameter is given by

$$\hat{\lambda} \in \underset{\lambda \in [\lambda_{\min}, \lambda_{\max}]}{\arg\min} \; E_{\text{test}}(\hat{\beta}^{(\lambda)}) = \mathcal{L}(y_{\text{test}}, X_{\text{test}} \hat{\beta}^{(\lambda)})$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \, f(X_{\text{train}} \beta) + \lambda \Omega(\beta)$$

Challenges:

- **non-smooth** and **non-convex** objective function
- **costly** to evaluate $E_{\text{test}}(\hat{\beta}^{(\lambda)})$ (*e.g.*, dense/continuous grid)

Hyperparameter tuning as bilevel optimization

The "optimal" hyperparameter is given by

$$\hat{\lambda} \in \underset{\lambda \in [\lambda_{\min}, \lambda_{\max}]}{\arg\min} \; E_{\text{test}}(\hat{\beta}^{(\lambda)}) = \mathcal{L}(y_{\text{test}}, X_{\text{test}} \hat{\beta}^{(\lambda)})$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \, f(X_{\text{train}} \beta) + \lambda \Omega(\beta)$$

Challenges:

- **non-smooth** and **non-convex** objective function
- **costly** to evaluate $E_{\text{test}}(\hat{\beta}^{(\lambda)})$ (*e.g.*, dense/continuous grid)

Tracking the curve of solutions

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} f(X\beta) + \lambda\Omega(\beta) := \mathcal{P}_\lambda(\beta)$$

Exact Path: For $(f, \Omega) =$ (Piecewise Quadratic, Piecewise Linear) the function $\lambda \longmapsto \hat{\beta}^{(\lambda)}$ is piecewise linear (`Lars`[(3)]).

Drawbacks:

- Exponential[(4)] complexity for Lasso $O((3^p + 1)/2)$
- Numerical instabilities[(5)]
- Hard to generalize to other losses / regularizations
- Cannot benefited of early stopping rule[(6)]

---

(3) B. Efron et al. "Least angle regression". In: *Ann. Statist.* 32.2 (2004). With discussion, and a rejoinder by the authors, pp. 407–499.

(4) J. Mairal and B. Yu. "Complexity analysis of the Lasso regularization path". In: *ICML*. 2012, pp. 353–360.

(5) Y. Li and Y. Singer. "The Well Tempered Lasso". In: *ICML* (2018), pp. 3030–3038.

(6) L. Bottou and O. Bousquet. "The tradeoffs of large scale learning". In: *NIPS*. 2008, pp. 161–168.

Tracking the curve of solutions

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} f(X\beta) + \lambda\Omega(\beta) := \mathcal{P}_\lambda(\beta)$$

Exact Path: For $(f, \Omega) =$ (Piecewise Quadratic, Piecewise Linear) the function $\lambda \longmapsto \hat{\beta}^{(\lambda)}$ is piecewise linear (`Lars`[3]).

Drawbacks:

- Exponential[4] complexity for Lasso $O((3^p + 1)/2)$
- Numerical instabilities[5]
- Hard to generalize to other losses / regularizations
- Cannot benefited of early stopping rule[6]

---

[3] B. Efron et al. "Least angle regression". In: *Ann. Statist.* 32.2 (2004). With discussion, and a rejoinder by the authors, pp. 407–499.

[4] J. Mairal and B. Yu. "Complexity analysis of the Lasso regularization path". In: *ICML*. 2012, pp. 353–360.

[5] Y. Li and Y. Singer. "The Well Tempered Lasso". In: *ICML* (2018), pp. 3030–3038.

[6] L. Bottou and O. Bousquet. "The tradeoffs of large scale learning". In: *NIPS*. 2008, pp. 161–168.

Aparté: Duality for the Lasso

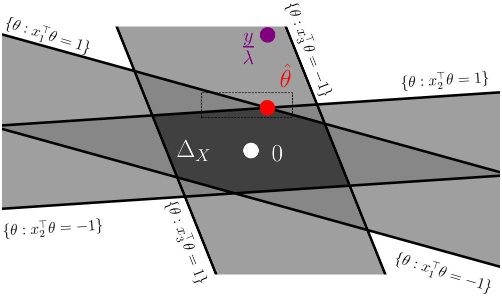
$$\hat{\theta}^{(\lambda)} = \underset{\theta \in \Delta_X}{\arg\max} \underbrace{\frac{1}{2} \|y\|^2 - \frac{\lambda^2}{2} \|y/\lambda - \theta\|^2}_{\mathcal{D}_\lambda(\theta)}$$

$\Delta_X = \{\theta \in \mathbb{R}^n : \forall j \in [p],\ |\mathbf{x}_j^\top \theta| \leqslant 1\}$: **dual feasible set**

Aparté: Duality for the Lasso

$$\hat{\theta}^{(\lambda)} = \underset{\theta \in \Delta_X}{\arg\max} \underbrace{\frac{1}{2}\|y\|^2 - \frac{\lambda^2}{2}\|y/\lambda - \theta\|^2}_{\mathcal{D}_\lambda(\theta)}$$

$\Delta_X = \{\theta \in \mathbb{R}^n : \forall j \in [p],\ |\mathbf{x}_j^\top \theta| \leqslant 1\}$: **dual feasible set**

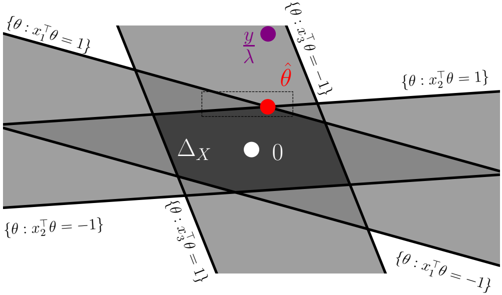


Toy visualization example: $n = 2, p = 3$

Aparté: Duality for the Lasso

$$\hat{\theta}^{(\lambda)} = \underset{\theta \in \Delta_X}{\arg\max} \underbrace{\frac{1}{2}\|y\|^2 - \frac{\lambda^2}{2}\|y/\lambda - \theta\|^2}_{\mathcal{D}_\lambda(\theta)}$$

$\Delta_X = \{\theta \in \mathbb{R}^n : \forall j \in [p],\ |\mathbf{x}_j^\top \theta| \leqslant 1\}$: **dual feasible set**

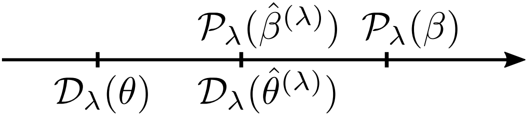


Projection problem: $\hat{\theta}^{(\lambda)} = \Pi_{\Delta_X}(y/\lambda)$

Duality gap as a stopping criterion

For any primal-dual pair $(\beta, \theta) \in \mathbb{R}^p \times \Delta_X$:

$$\textbf{(Dual)} \quad \mathcal{D}_\lambda(\theta) \leqslant \mathcal{D}_\lambda(\hat{\theta}^{(\lambda)}) = \mathcal{P}_\lambda(\hat{\beta}) \leqslant \mathcal{P}_\lambda(\beta^{(\lambda)}) \quad \textbf{(Primal)}$$



Duality gap : $\quad \mathrm{gap}_\lambda(\beta, \theta) := \mathcal{P}_\lambda(\beta) - \mathcal{D}_\lambda(\theta)$

upper bound on **suboptimality gap** : $\quad \mathcal{P}_\lambda(\beta) - \mathcal{P}_\lambda(\hat{\beta}^{(\lambda)})$

$$\forall \beta, (\exists \theta \in \Delta_X, \mathrm{gap}_\lambda(\beta, \theta) \leqslant \epsilon) \Rightarrow \mathcal{P}_\lambda(\beta) - \mathcal{P}_\lambda(\hat{\beta}^{(\lambda)}) \leqslant \epsilon$$

i.e., β is an ϵ-solution whenever $\mathrm{gap}_\lambda(\beta, \theta) \leqslant \epsilon$

Approximate path: adaptive grid[7]

Start : fix grid upper ($\lambda_{\max}$) lower ($\lambda_{\min}$) bound

Quadratic bound: helps get ϵ-accurate grid on $[\lambda_{\min}, \lambda_{\max}]$

$$\mathcal{P}_\lambda(\beta^{(\lambda_t)}) - \mathcal{P}_\lambda(\hat{\beta}^{(\lambda)}) \leqslant \mathsf{gap}_\lambda(\beta^{(\lambda_t)}, \theta^{(\lambda_t)}) \quad \leqslant Q_{\lambda_t}\left(1 - \frac{\lambda}{\lambda_t}\right)$$

Rem: holds whenever f is strongly convex

[7] J. Giesen et al. "Approximating concavely parameterized optimization problems". In: *NIPS*. 2012, pp. 2105–2113.

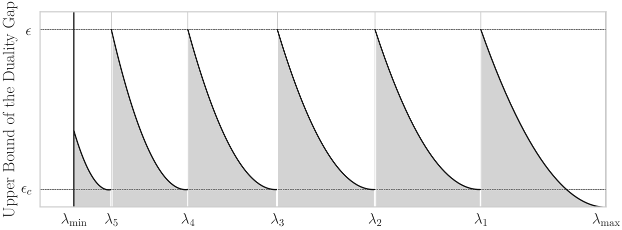
Approximate path: adaptive grid[7]

Start : fix grid upper ($\lambda_{\max}$) lower ($\lambda_{\min}$) bound

Quadratic bound: helps get ϵ-accurate grid on $[\lambda_{\min}, \lambda_{\max}]$

$$\mathcal{P}_\lambda(\beta^{(\lambda_t)}) - \mathcal{P}_\lambda(\hat{\beta}^{(\lambda)}) \leqslant \mathsf{gap}_\lambda(\beta^{(\lambda_t)}, \theta^{(\lambda_t)}) \quad \leqslant Q_{\lambda_t}\left(1 - \frac{\lambda}{\lambda_t}\right)$$

Rem: holds whenever f is strongly convex



[7] J. Giesen et al. "Approximating concavely parameterized optimization problems". In: *NIPS*. 2012, pp. 2105–2113.

Approximation of the validation path

$$\operatorname*{arg\,min}_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} E_{\text{test}}(\hat\beta^{(\lambda)}) = \mathcal{L}(y_{\text{test}}, X_{\text{test}}\hat\beta^{(\lambda)})$$

$$\text{s.t. } \hat\beta^{(\lambda)} \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} f(X_{\text{train}}\beta) + \lambda\Omega(\beta)$$

Bound the validation Gap[8],[9]

$$\left| E_{\text{test}}(\hat\beta^{(\lambda)}) - E_{\text{test}}(\beta^{(\lambda_t)}) \right| \leqslant \max_{\beta \in \mathcal{B}_\lambda} \mathcal{L}(X_{\text{test}}\beta, X_{\text{test}}\beta^{(\lambda_t)}) \enspace ,$$

[8] A. Shibagaki et al. "Regularization Path of Cross-Validation Error Lower Bounds". In: *NIPS*. 2015, pp. 1666–1674.

[9] E. Ndiaye et al. "Gap Safe screening rules for sparsity enforcing penalties". In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.

Approximation of the validation path

$$\arg\min_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} E_{\text{test}}(\hat{\beta}^{(\lambda)}) = \mathcal{L}(y_{\text{test}}, X_{\text{test}} \hat{\beta}^{(\lambda)})$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} f(X_{\text{train}} \beta) + \lambda \Omega(\beta)$$

Bound the validation Gap[8],[9]

$$\left| E_{\text{test}}(\hat{\beta}^{(\lambda)}) - E_{\text{test}}(\beta^{(\lambda_t)}) \right| \leqslant \max_{\beta \in \mathcal{B}_\lambda} \mathcal{L}(X_{\text{test}} \beta, X_{\text{test}} \beta^{(\lambda_t)}) \enspace ,$$

$$\text{where} \quad \mathcal{B}_\lambda = \text{Ball}\left(\beta^{(\lambda_t)}, r_t\right) \ni \hat{\beta}^{(\lambda)}$$

Rem: $r_t = \sqrt{\frac{\mu}{2} \text{gap}(\beta^{(\lambda_t)}, \theta^{(\lambda_t)})}$ for μ-strongly convex $\mathcal{P}_\lambda$ (Enet)

---

(8) A. Shibagaki et al. “Regularization Path of Cross-Validation Error Lower Bounds”. In: *NIPS*. 2015, pp. 1666–1674.

(9) E. Ndiaye et al. “Gap Safe screening rules for sparsity enforcing penalties”. In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.

Approximation of the validation path

$$\arg\min_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} E_{\text{test}}(\hat{\beta}^{(\lambda)}) = \mathcal{L}(y_{\text{test}}, X_{\text{test}} \hat{\beta}^{(\lambda)})$$

$$\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} f(X_{\text{train}} \beta) + \lambda \Omega(\beta)$$

Bound the validation Gap[(8),(9)]

$$\left| E_{\text{test}}(\hat{\beta}^{(\lambda)}) - E_{\text{test}}(\beta^{(\lambda_t)}) \right| \leqslant \max_{\beta \in \mathcal{B}_\lambda} \mathcal{L}(X_{\text{test}} \beta, X_{\text{test}} \beta^{(\lambda_t)}) \;,$$

$$\text{where} \quad \mathcal{B}_\lambda = \text{Ball}\left(\beta^{(\lambda_t)}, r_t\right) \ni \hat{\beta}^{(\lambda)}$$

<u>Rem</u>: $r_t = \sqrt{\frac{\mu}{2} \text{gap}(\beta^{(\lambda_t)}, \theta^{(\lambda_t)})}$ for μ-strongly convex $\mathcal{P}_\lambda$ (Enet)

(8) A. Shibagaki et al. "Regularization Path of Cross-Validation Error Lower Bounds". In: *NIPS*. 2015, pp. 1666–1674.

(9) E. Ndiaye et al. "Gap Safe screening rules for sparsity enforcing penalties". In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.

Testing (Validation) control

Motivation: fix a precision level ϵ_v on the testing (or validation) set; then calibrate the optimization accuracy needed ϵ to target this precision.

Theorem

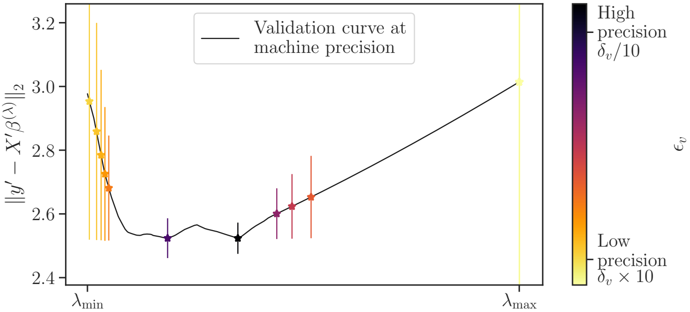
When $\mathcal{P}_\mu$ is a μ-strongly convex function, with the grid construction provided before

$$\forall \lambda \in [\lambda_{\min}, \lambda_{\max}], \exists \lambda_t \in \text{grid}, \quad \left| E_{\text{test}}(\hat{\beta}^{(\lambda)}) - E_{\text{test}}(\beta^{(\lambda_t)}) \right| \leqslant \epsilon_v$$

provided the algorithm is run up to precision ϵ at training, with

$$\epsilon = \frac{\mu}{2} \left(\frac{\epsilon_v}{\|X_{\text{test}}\|} \right)^2$$

Approximation of the optimal hyperparameter



Conclusion

- Extension to GLM (more technical, but done)
- Take home message: more connexions needed between optimization / statistics / learning
- Future works: What about several parameters? How to handle vanilla CV & variants?

Code: https://github.com/EugeneNdiaye/safe_grid_search
ICML paper: https://arxiv.org/abs/1810.05471

Powered with **MooseTeX**

One last word

"*All models are wrong but some come with good open source implementation and good documentation so use those.*"

A. Gramfort

References I

- Bickel, P. J., Y. Ritov, and A. B. Tsybakov. "Simultaneous analysis of Lasso and Dantzig selector". In: *Ann. Statist.* 37.4 (2009), pp. 1705–1732.
- Bottou, L. and O. Bousquet. "The tradeoffs of large scale learning". In: *NIPS*. 2008, pp. 161–168.
- Candès, E. J., M. B. Wakin, and S. P. Boyd. "Enhancing Sparsity by Reweighted l_1 Minimization". In: *J. Fourier Anal. Applicat.* 14.5-6 (2008), pp. 877–905.
- Chen, S. S., D. L. Donoho, and M. A. Saunders. "Atomic decomposition by basis pursuit". In: *SIAM J. Sci. Comput.* 20.1 (1998), pp. 33–61.
- Efron, B. et al. "Least angle regression". In: *Ann. Statist.* 32.2 (2004). With discussion, and a rejoinder by the authors, pp. 407–499.
- Giesen, J. et al. "Approximating concavely parameterized optimization problems". In: *NIPS*. 2012, pp. 2105–2113.

References II

- Li, Y. and Y. Singer. “The Well Tempered Lasso”. In: *ICML* (2018), pp. 3030–3038.
- Mairal, J. and B. Yu. “Complexity analysis of the Lasso regularization path”. In: *ICML*. 2012, pp. 353–360.
- Ndiaye, E. et al. “Gap Safe screening rules for sparsity enforcing penalties”. In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.
- Shibagaki, A. et al. “Regularization Path of Cross-Validation Error Lower Bounds”. In: *NIPS*. 2015, pp. 1666–1674.
- Sun, T. and Cun-Hui Zhang. “Scaled sparse linear regression”. In: *Biometrika* 99.4 (2012), pp. 879–898.
- Tibshirani, R. “Regression Shrinkage and Selection via the Lasso”. In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58.1 (1996), pp. 267–288.
- Zou, H. “The adaptive lasso and its oracle properties”. In: *J. Amer. Statist. Assoc.* 101.476 (2006), pp. 1418–1429.