Bilevel Hyperparameter Learning for Nonsmooth Regularized Imaging and ML Models

David Villacís

joint work with Pedro Péres-Aros and Emilio Vilches

AIP 2025 - Rio de Janeiro, Brasil

1. Motivation & Problem Statement

Context

Let us consider a variational formulation of an inverse problem parametrized by $x \in X$:

$$\min_{y \in Y} F(\mathcal{A}(y); d) + G_x(y)$$

In particular we are interested in the following weighted regularizer:

$$G_x(y) = \sum_{i=1}^{n} \psi(x_i)|y_i|$$

Context

- Weighted regularizers are valuable when the model needs to be **robust** and able to handle **small and structured disturbances** on the data.
- By assigning different weights to individual features, the model can **adapt** more flexibly to uncertainty.
- Maintain **stable** performance under noise or variation.

Xu et. al. (2008), Guo et. al. (2023)

Applications

Overparametrized regression

$$\min_{y \in \mathbb{R}^n} \frac{1}{2} \parallel A^{\text{train}} y - y^{\text{train}} \parallel^2 + \sum_{i=1}^{n} \exp(x_i)|y_i|$$

where $A^{\text{train}} \in \mathbb{R}^{n \times m}$ with $n \ll m$ and $y^{\text{train}} \in \mathbb{R}^n$ are the sample matrix and the corresponding target vector, respectively.

Applications

Wavelet-based image restoration

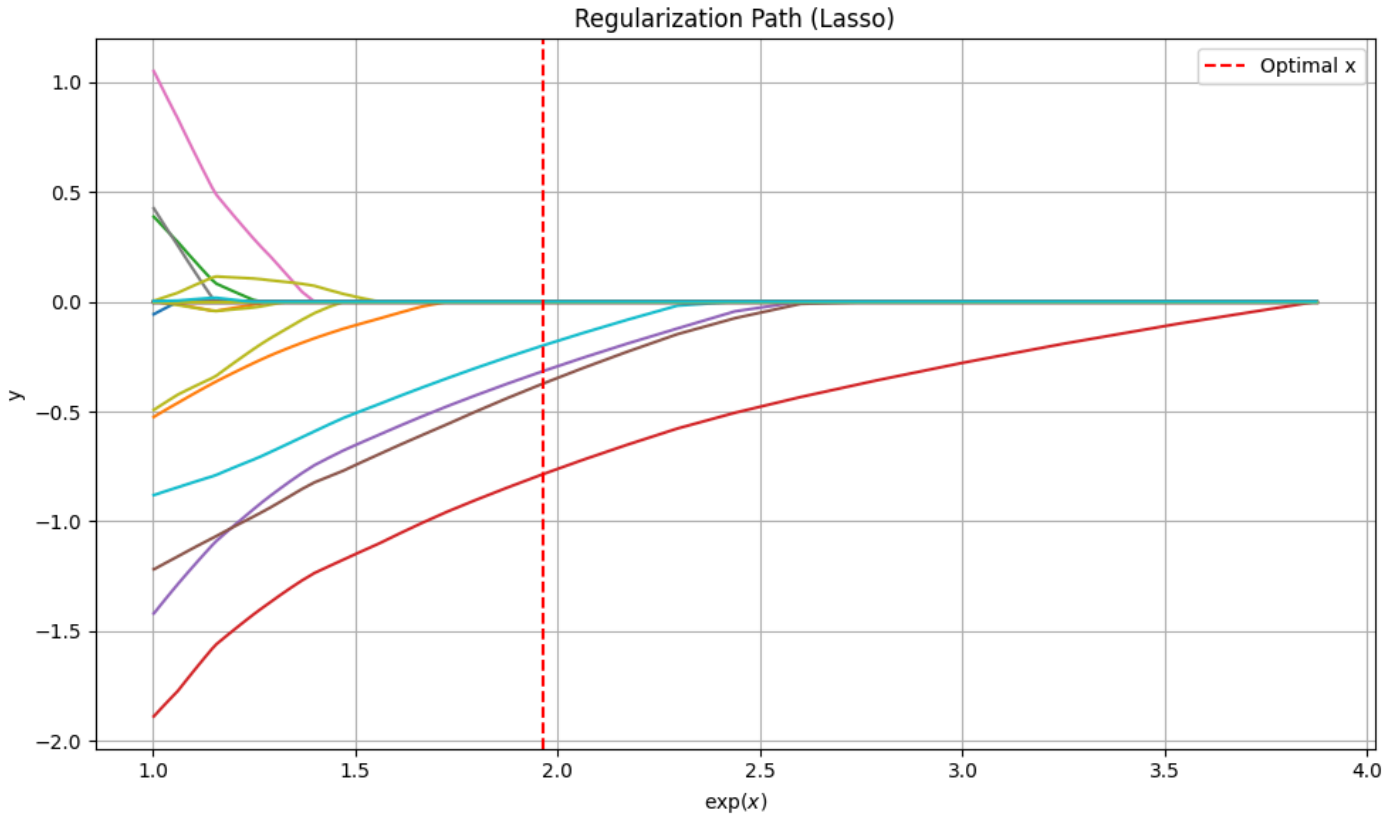
$$\min_{y \in \mathbb{R}^n} \frac{1}{2} \parallel RWy - y^{\text{damaged}} \parallel^2 + \sum_{i=1}^{n} \exp(x_i)|y_i|$$

where R is a blurring matrix and W contains a wavelet basis.

- Multiplying by W corresponds to performing the inverse wavelet transform.

Challenge

The hyperparameter x is **not known** and must be learned from data.



Challenge

The optimal selection of the hyperparameter x has been addressed by traditional families of techniques:

Grid search: evaluate the model for a set of hyperparameter values and select the one that minimizes the validation error. $\rightarrow$ Cross-validation.

Bayesian optimization: model the hyperparameter space with a probabilistic model and select the hyperparameter that maximizes the expected improvement. $\rightarrow$ low dimensionality.

2. Learning Model

Bilevel Hyperparameter Learning

We consider the optimal hyperparameter as the **solution of a bilevel optimization** problem:

$$\min_{x \in X} L(x, y^*(x)) \ \text{ s.t. } y^*(x) \in S(x) := \arg\min_{y \in Y} \{F(y) + G_x(y)\}$$

where L evaluates the performance of the learned hyperparameter on a **validation set**, and $S(x)$ denotes the solution set of the lower-level problem, obtained from the **training set**.

Bilevel Hyperparameter Learning (Smooth Case)

If the solution map S is **single-valued and continuously differentiable**, then the bilevel problem can be reformulated as a single-level optimization problem:

$$\min_{x \in X} \Phi(x) := L(x, S(x)).$$

We can then compute its gradient with respect to x using the IFT and the chain rule (hypergradient):

$$\nabla_x \Phi(x) = \nabla_x L(x, S(x)) + S'(x)^\top \nabla_y L(x, S(x))$$

Limitations & Gap

In the case of our regularizer, G_x is **not differentiable**

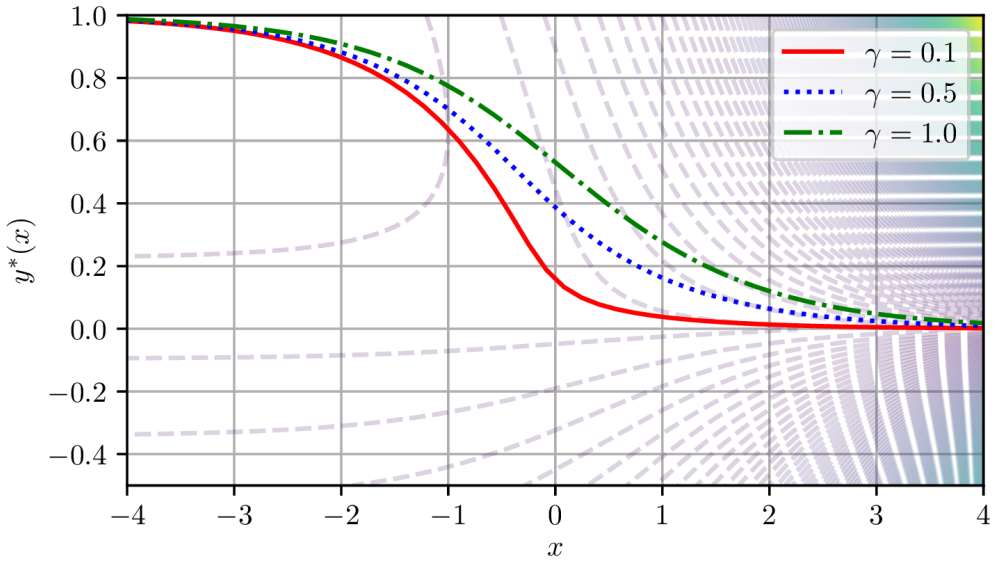


Figure 1: $|y_i|_\gamma = \sqrt{y_i^2 + \gamma^2}$

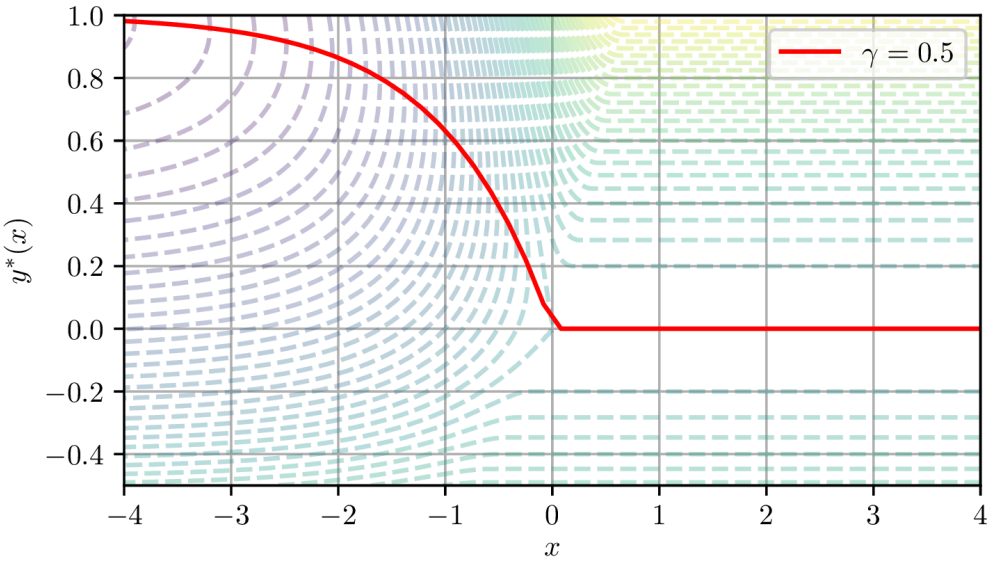


Figure 2: FB Reformulation

3. Forward-Backward Reformulation

Let us introduce the **forward-backward** operator and the **residual** operator as follows:

$$T^{\gamma}(x,y) := \operatorname{prox}_{\gamma G_x}(y - \gamma \nabla F(y)) = (\mathcal{T} \circ \mathcal{H}^{\gamma})(x,y),$$
$$R^{\gamma}(x,y) := \gamma^{-1}(y - T^{\gamma}(x,y)),$$

where $\mathcal{T}$ is the **component-wise soft-thresholding** operator and $\mathcal{H}^{\gamma}$ is defined as:

$$\mathcal{T}(u,v) := (\operatorname{sign}(v_i)\max(|v_i| - \gamma, 0))_{i=1}^{n},$$
$$\mathcal{H}^{\gamma}(x,y) := (\gamma\Psi(x), y - \gamma\nabla F(y)).$$

Bilevel FB Reformulation

Then, we can reformulate the bilevel problem as:

$$\min_{x \in X} L(x, y^*(x))$$
$$\text{s.t. } y^*(x) \in S(x) := \{y \in Y \mid R^\gamma(x, y) = 0\}$$

Bilevel FB Reformulation

Then, we can reformulate the bilevel problem as:

$$\min_{x \in X} L(x, y^*(x))$$
$$\text{s.t. } y^*(x) \in S(x) := \{y \in Y \mid R^\gamma(x, y) = 0\}$$

- Does the **solution set** of this problem **coincide** with the original one?
- What **properties** does the solution mapping $S : X \to Y$ have?
- Can we compute the **hyper(sub)gradient**?

Lower Level Solution Set

The solution set of the original lower level problem and their FB reformulation coincide.

Proposition: Let $\gamma > 0$, $x \in X$ and $y \in Y$. Then,

$$(i)\ R^{\gamma}(x, y) = 0,$$
$$(\text{ii})\ 0 \in \nabla F(y) + \partial G_x(y),$$
$$(\text{iii})\ y \in S(x)$$

Solution Set and Mapping

Theorem: For every $x \in X$, the set $S(x)$ is **single-valued**.

Moreover, the mapping $S : X \to Y$ is **locally Lipschitz continuous**.

$$\min_{x \in X} L(x, y^*(x))$$

$$\text{s.t. } y^*(x) = S(x) := \{y \in Y \mid y - (\mathcal{T} \circ \mathcal{H}^\gamma)(x, y) = 0\}$$

Limiting Subdifferential

Lemma: Let L be strictly differentiable at $(\bar{x}, \bar{y})$ with $\bar{y} := S(\bar{x})$, then we have the equality:

$$\partial\Phi(\bar{x}) = \nabla_x L(\bar{x}, \bar{y}) + D^* S(\bar{x})\big(\nabla_y L(\bar{x}, \bar{y})\big).$$

where $D^*S(\bar{x})$ is the **Mordukhovich coderivative** of S at $\bar{x}$.

Generalized Differentiation

Given a closed set C, the **tangent cone** to a set C at $\bar{x} \in C$ is

$$T(\bar{x}; C) := \{d \in X \mid \exists x_k \to \bar{x}, t_k \downarrow 0, \text{s.t. } \bar{x} + t_k d \in C\}.$$

The **Fréchet normal cone** to C at $\bar{x} \in C$ is given by $\widehat{N}(\bar{x}; C) := T(\bar{x}; C)^\circ$.

The **Mordukhovich (limiting) normal cone** to C at $\bar{x} \in C$ is defined as:

$$N(\bar{x}; C) := \{x^* \in X \mid \exists x_k^* \in \widehat{N}(x_k; C), \text{s.t. } (x_k, x_k^*) \to (\bar{x}, x^*)\}.$$

The **Mordukhovich coderivative** of a mapping $S : X \to Y$ at $\bar{x} \in X$ is:

$$D^*S(\bar{x})(y^*) := \{x^* \in X \mid (x^*, -y^*) \in N(\bar{x}; \text{gph } S), \text{for some } y^* \in Y\}$$

Coderivative Estimation

Lemma: Let $\gamma > 0$, $\bar{x} \in X$, $\bar{y} := S(\bar{x})$ and $R^\gamma : X \times Y \to Z$. Then, provided that $\ker\, D^* R^\gamma(\bar{x}, \bar{y}) = \{0\}$, the following relation holds:

$$D^* S(\bar{x})(y^*) \subset \{x^* \in X |\ (x^*, -y^*) \in D^* R^\gamma(\bar{x}, \bar{y})(z^*),\ \text{for some } z^* \in Z\}$$

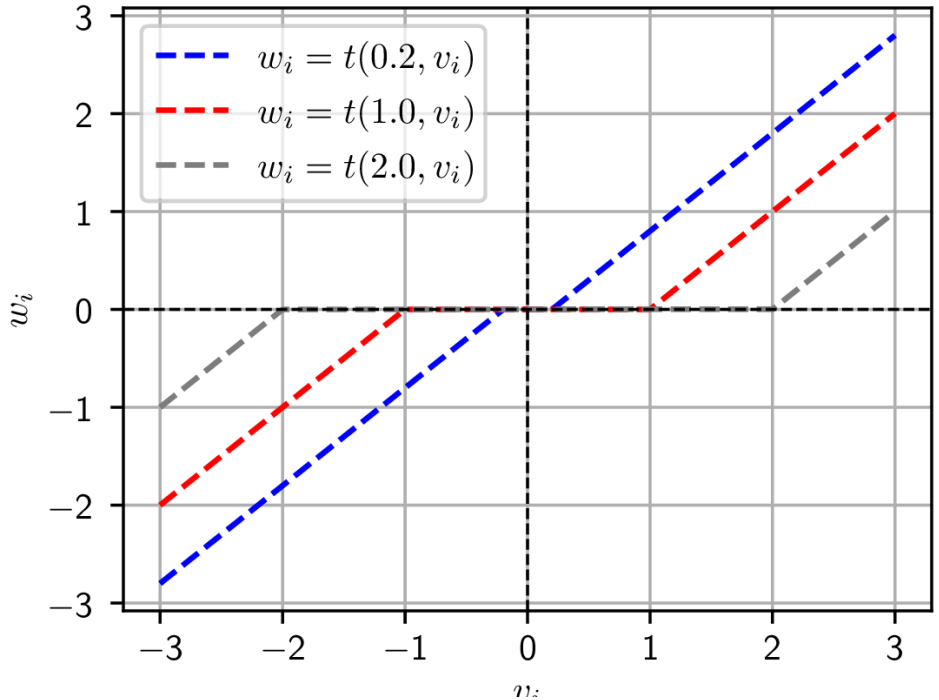
4. Coderivative Calculus for R^γ

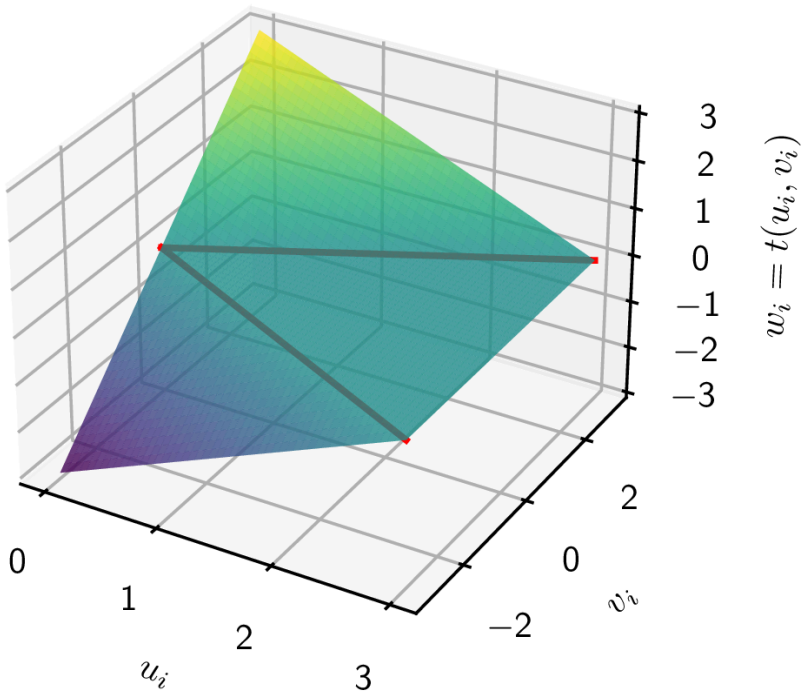
Once verified that the mappings $\mathcal{T}$ and $\mathcal{H}^\gamma$ **satisfy the composition and sum rules** for the coderivative, we may rewrite the coderivative of the residual operator R^γ as follows:

$$\begin{aligned} D^*R^\gamma(x,y)(z^*) &= (0,z^*) + D^*(\mathcal{T}\circ\mathcal{H}^\gamma)(x,y)(-z^*) \\ &= (0,z^*) + \nabla\mathcal{H}^\gamma(x,y)^\top \circ D^*\mathcal{T}(\mathcal{H}^\gamma(x,y))(z^*), \end{aligned}$$

where

$$D^*\mathcal{T}(u,v)(z^*) := \{(u,v) | (u^*,v^*,-z^*) \in N((u,v,w); \operatorname{gph}\mathcal{T})\}$$


$w_i = t(0.2, v_i)$
$w_i = t(1.0, v_i)$
$w_i = t(2.0, v_i)$
w_i
v_i


$w_i = t(u_i, v_i)$
u_i
v_i

$$D^*\mathcal{T}(u,v)(z^*) =$$

$$\left\{ (u^*, v^*) \in \mathbb{R}^n_+ \times \mathbb{R}^n \;\middle|\; \begin{cases} u_i^* = -z_i^*,\ v_i^* = z_i^*, & \text{if } i \in \mathcal{I}^+(u,v), \\ u_i^* = z_i^*,\ v_i^* = z_i^*, & \text{if } i \in \mathcal{I}^-(u,v), \\ u_i^* = v_i^* = 0, & \text{if } i \in \mathcal{A}(u,v), \\ \left.\begin{array}{l} u_i^* = v_i^* = 0 \vee \\ u_i^* = -v_i^*,\ v_i^* \in [0, z_i^*] \vee \\ u_i^* = -z_i^*,\ v_i^* = z_i^* \end{array}\right\} & \text{if } i \in \mathcal{B}^+(u,v), \\ \left.\begin{array}{l} u_i^* = v_i^* = 0 \vee \\ u_i^* = v_i^*,\ v_i^* \in [-z_i^*, 0] \vee \\ u_i^* = z_i^*,\ v_i^* = z_i^* \end{array}\right\} & \text{if } i \in \mathcal{B}^-(u,v). \end{cases} \right\}$$

Residual coderivative

Lemma: Let F be twice continuously differentiable and let $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ and take $\gamma \in (0, \Lambda_F^{-1})$. Then, the coderivative of the residual operator R^γ is:

$$D^* R^\gamma(x,y)(z^*) = \left\{ (x^*, -y^*) | \right.$$

$$\left. \begin{cases} x^* = \gamma \nabla \Psi(x) u^* \\ -y^* = z^* + (I - \gamma \nabla^2 F(y))^\top v^* \end{cases}, \exists (u^*, v^*) \in D^* \mathcal{T}(\mathcal{H}^\gamma(x,y))(-z^*) \right\}$$

Residual coderivative

Lemma: Let F be twice continuously differentiable and let $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ and take $\gamma \in \left(0, \Lambda_F^{-1}\right)$. Then, the kernel of coderivative of the residual operator R^γ satisfy:

$$\ker\, D^* R^\gamma(x, y) = \{0\}$$

5. Algorithm

Limiting subgradient selection

Lemma: Let $\gamma \in \left(0, \Lambda_F^{-1}\right)$ and fix $x \in \mathbb{R}^n$. If

$$\mathcal{B}^+(\mathcal{H}^\gamma(x, S(x))) \cup \mathcal{B}^-(\mathcal{H}^\gamma(x, S(x))) = \emptyset.$$

Then the coderivative of the solution operator S at x is a singleton defined by:

$$D^*S(x)(y^*) = \{-\Xi\Upsilon^{-1}y^*\}.$$

Remark: This result also holds for arbitrary partitions of the biactive sets.

Limiting subgradient selection

$\Xi \in \mathbb{R}^{n\times n}$ and $\Upsilon \in \mathbb{R}^{n\times n}$ are square matrices built as follows:

$$\Xi_i = \begin{cases} \gamma \nabla \Psi(x)_i & \text{if } i \in \mathcal{I}^+(\mathcal{H}^\gamma(x, S(x))) \\ -\gamma \nabla \Psi(x)_i & \text{if } i \in \mathcal{I}^-(\mathcal{H}^\gamma(x, S(x))) \\ 0_i & \text{otherwise} \end{cases}$$

$$\Upsilon_i = \begin{cases} \nabla^2 F(S(x))_i & \text{if } i \notin \mathcal{A}(\mathcal{H}^\gamma(x, S(x))) \\ 0_i & \text{otherwise} \end{cases}$$

Subgradient selection: $\eta = \nabla_x L(\bar{x}, \bar{y}) - \Xi \Upsilon^{-1} \nabla_y L(\bar{x}, \bar{y})$

Subgradient computation

Algorithm 1 Subgradient Computation

Require: x, y, γ

1: Set $w \leftarrow y - \gamma \nabla F(y)$

2: Given (x, w), compute $\nabla^2 F(y)$, Ξ and Υ, according to (29)

3: Solve the linear system $\Upsilon \xi = -\nabla_y L(x, y)$

4: **return** $\nabla_x L(x, y) + \Xi \xi$

Nonsmooth Bilevel Approximation (NBA-wL_1)

Algorithm 2 Nonsmooth Bilevel Approximation (NBA-$w\ell_1$)

Require: x_0, tol > 0, γ, K

for $k \in \{0, \ldots, K\}$ **do**
  Set $y_k \leftarrow S(x_k)$
  Call Algorithm 1 with (x_k, y_k) and parameter γ to obtain η_k
  Call the line-search procedure with (x_k, η_k) and obtain α_k
  Set $x_{k+1} \leftarrow x_k - \alpha_k \eta_k$
  if $\|\eta_k\| <$ tol **then**
    return x_k
  end if
  $k \leftarrow k + 1$
end for

6. Experiments

Weighted Elastic Net Linear Regression

$$\min_{x \in \mathbb{R}^m} L(x, y^*(x)) := \frac{1}{2} \parallel A^{\text{val}} y^*(x) - b^{\text{val}} \parallel^2$$

$$\text{s.t. } y^*(x) = \arg\min_{y \in \mathbb{R}^n} \left\{ \frac{1}{2} \parallel A^{\text{train}} y - b^{\text{train}} \parallel^2 + \frac{\alpha}{2} \|y\|^2 + \sum_{i=1}^{m} \exp(x_i)|y_i| \right\}$$

Weighted Elastic Net Linear Regression

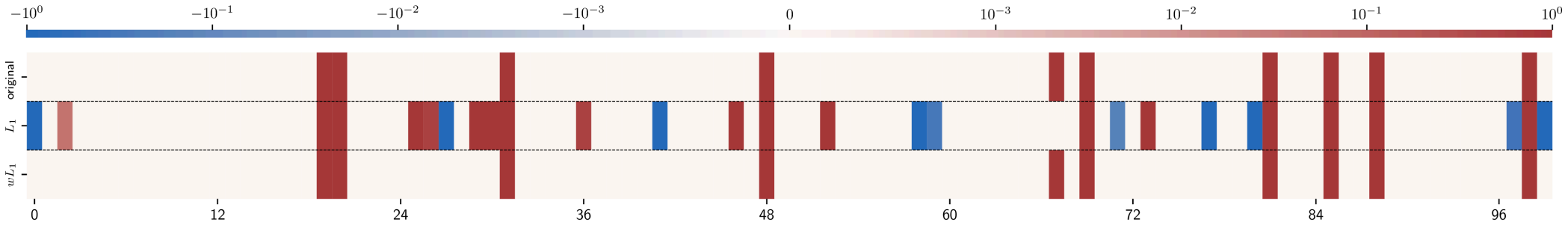


Figure 3: $n = 100, m = 100, \text{nz} = 10$

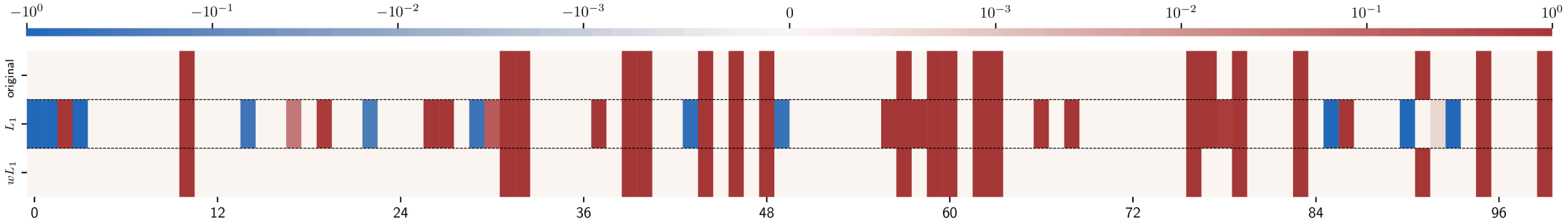


Figure 4: $n = 100, m = 100, \text{nz} = 20$

Weighted Elastic Net Linear Regression

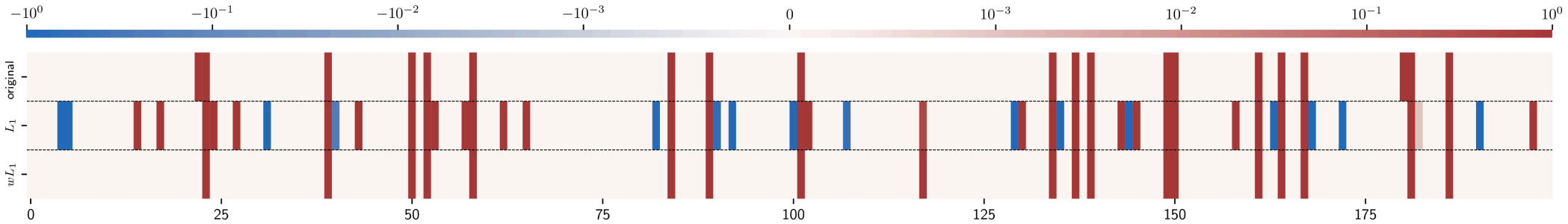


Figure 5: $n = 100, m = 200, \text{nz} = 20$

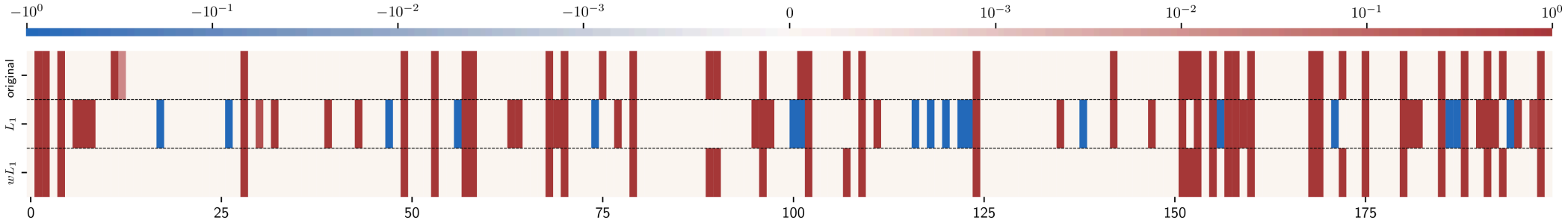


Figure 6: $n = 100, m = 200, \text{nz} = 40$

Weighted Elastic Net Linear Regression

n	m	nz	NMSE_{val} scalar	NMSE_{val} weighted	$\text{NMSE}_{\text{test}}$ scalar	$\text{NMSE}_{\text{test}}$ weighted
100	100	5	0.0247 ± 0.009	0.0098 ± 0.003	0.0245 ± 0.010	0.0097 ± 0.003
	100	10	0.0458 ± 0.017	0.0124 ± 0.004	0.0456 ± 0.017	0.0127 ± 0.005
	100	20	0.0637 ± 0.028	0.0110 ± 0.004	0.0594 ± 0.025	0.0122 ± 0.004
	200	10	0.0493 ± 0.009	0.0103 ± 0.002	0.0469 ± 0.009	0.0102 ± 0.002
	200	20	0.1005 ± 0.028	0.0126 ± 0.003	0.1052 ± 0.036	0.0147 ± 0.005
	200	40	0.2739 ± 0.073	0.0185 ± 0.007	0.2767 ± 0.097	0.0223 ± 0.006
	300	15	0.0896 ± 0.020	0.0107 ± 0.002	0.0841 ± 0.017	0.0114 ± 0.002
	300	30	0.2787 ± 0.076	0.0203 ± 0.007	0.2849 ± 0.070	0.0225 ± 0.007
	300	60	0.5562 ± 0.063	0.0574 ± 0.058	0.5867 ± 0.061	0.1227 ± 0.075
200	200	10	0.0242 ± 0.005	0.0102 ± 0.001	0.0246 ± 0.005	0.0100 ± 0.001
	200	20	0.0320 ± 0.005	0.0100 ± 0.002	0.0327 ± 0.005	0.0103 ± 0.001
	200	40	0.0597 ± 0.011	0.0121 ± 0.003	0.0633 ± 0.012	0.0132 ± 0.003
	400	20	0.0602 ± 0.022	0.0118 ± 0.003	0.0608 ± 0.022	0.0125 ± 0.003
	400	40	0.1249 ± 0.032	0.0136 ± 0.003	0.1223 ± 0.035	0.0158 ± 0.004
	400	80	0.2793 ± 0.057	0.0196 ± 0.004	0.2745 ± 0.052	0.0296 ± 0.005
	600	30	0.1169 ± 0.023	0.0138 ± 0.003	0.1164 ± 0.021	0.0151 ± 0.004
	600	60	0.3326 ± 0.043	0.0175 ± 0.003	0.3265 ± 0.045	0.0210 ± 0.005
	600	120	0.6042 ± 0.082	0.0315 ± 0.013	0.5943 ± 0.046	0.0850 ± 0.025

Weighted Elastic Net Logistic Regression

$$\min_{x \in \mathbb{R}^m} l\big(y^*(x); A^{\text{val}}, b^{\text{val}}\big)$$

$$\text{s.t. } y^*(x) = \arg\min_{y \in \mathbb{R}^n} \left\{ l(y; A^{\text{train}}, b^{\text{train}}) + \frac{\alpha}{2} \|y\|^2 + \sum_{i=1}^{m} \exp(x_i)|y_i| \right\}$$

where $l(y; A, b)$ is the binary cross-entropy loss.

$$l(y; A, b) := -\frac{1}{N} \sum_{i=1}^{N} [b_i \log(A_i^\top y) + (1 - b_i) \log(1 - A_i^\top y)]$$

Weighted Elastic Net Logistic Regression

n	m	nz	F_1^{val} scalar	F_1^{val} weighted	F_1^{test} scalar	F_1^{test} weighted
100	100	5	0.8695 ± 0.022	0.9598 ± 0.019	0.8846 ± 0.027	0.9276 ± 0.024
	100	10	0.8690 ± 0.037	0.9647 ± 0.017	0.8526 ± 0.015	0.9009 ± 0.032
	100	20	0.8259 ± 0.056	0.9709 ± 0.023	0.8212 ± 0.070	0.8687 ± 0.031
	200	10	0.8355 ± 0.036	0.9663 ± 0.007	0.8288 ± 0.019	0.8924 ± 0.034
	200	20	0.7420 ± 0.111	0.9189 ± 0.050	0.7651 ± 0.039	0.8320 ± 0.048
	200	40	0.5075 ± 0.284	0.6335 ± 0.367	0.5420 ± 0.305	0.5844 ± 0.327
200	200	10	0.9157 ± 0.019	0.9805 ± 0.010	0.9110 ± 0.006	0.9392 ± 0.002
	200	20	0.8858 ± 0.022	0.9828 ± 0.018	0.8563 ± 0.020	0.9182 ± 0.019
	200	40	0.8266 ± 0.061	0.9633 ± 0.018	0.8178 ± 0.029	0.8861 ± 0.013
	400	20	0.8559 ± 0.033	0.9688 ± 0.016	0.8685 ± 0.031	0.9339 ± 0.011
	400	40	0.7838 ± 0.051	0.9229 ± 0.069	0.7644 ± 0.047	0.8356 ± 0.061
	400	80	0.7138 ± 0.045	0.8915 ± 0.068	0.6928 ± 0.043	0.7450 ± 0.039

7. Conclusions

- We proposed a **bilevel optimization-based framework for tuning parameters** of models that consider a weighted l_1 regularizer.
- We proposed a **FB reformulation for a lower-level** problem that maintains the same solution set.
- We provided a **computable characterization of a limiting subgradient** of a bilevel optimization problem with a nonsmooth lower-level problem involving a weighted l_1 regularizer.

Thank you for your attention!

https://david.villacis.net