Supplement to " An Iterative Smoothing Algorithm for Regression with Structured Sparsity "

Fouad Hadj-Selem, Tommy Löfstedt, Vincent Frouin, Vincent Guillemot, Edouard Duchesnay

Supplement to "An Iterative Smoothing Algorithm for Regression with Structured Sparsity"

Fouad Hadj-Selem*, Tommy Löfstedt, Vincent Frouin, Vincent Guillemot and Edouard Duchesnay*

NeuroSpin, CEA, Paris-Saclay
91191 Gif sur Yvette, France

This document contains proofs and supplementary details for the paper "An Iterative Smoothing Algorithm for Regression with Structured Sparsity". All section numbers and equation numbers in this supplementary document are preceded by the letters A, to distinguish them from those from the main paper [5].

A 1. Definitions

Here we provide the definitions of two important concepts in non-differentiable optimization.

A 1.1. Lipschitz continuous function

Definition 1. *Let $\nabla f(\boldsymbol{\beta})$ be the gradient at $\boldsymbol{\beta}$ of a smooth real function f defined on $\mathbb{R}^p$. A function f has a Lipschitz continuous gradient on a convex set $\mathcal{K}$ with Lipschitz constant $L(\nabla(f)) \geq 0$ if for all $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathcal{K}$ we have*

$$\|\nabla f(\boldsymbol{\beta}_1) - \nabla f(\boldsymbol{\beta}_2)\|_2 \leq L(\nabla(f))\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|_2.$$

A 1.2. Proximal operator

Definition 2. *Let h: $\mathbb{R}^p \to \mathbb{R}$ be a closed proper (i.e. $h(\boldsymbol{\beta}) < +\infty$ for at least one $\boldsymbol{\beta}$, and $h(\boldsymbol{\beta}) > -\infty$ for all $\boldsymbol{\beta}$) convex function [1]. The proximal operator (or proximal mapping) $\text{prox}_h(x)$: $\mathbb{R}^p \to \mathbb{R}^p$ is then defined by*

$$\text{prox}_h(\boldsymbol{\beta}) \equiv \arg\min_{\boldsymbol{u} \in \mathbb{R}^p} \left\{ \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{\beta}\|_2^2 + h(\boldsymbol{u}) \right\}, \tag{A 1.1}$$

*Corresponding authors. e-mail: hadjselemfouad@gmail.com; edouard.duchesnay@cea.fr

Note that we will often encounter the proximal operator of the scaled function $t \cdot h(\cdot)$, where $t > 0$, which can be expressed as

$$\text{prox}_{th}(\boldsymbol{\beta}) \equiv \arg\min_{\boldsymbol{u} \in \mathbb{R}^p} \left\{ \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{\beta}\|_2^2 + th(\boldsymbol{u}) \right\}, \tag{A 1.2}$$

and will be referred to as the proximal operator of h with parameter t.

A 2. Proofs

This section details the proofs of the three theorems that underlie the principal contributions of the main paper [5].

A 2.1. Proof of Theorem 2: Optimal smoothing parameter μ

In order to achieve the level of precision ε on f by minimizing f_μ, it is sufficient that

$$\mu \in \left(0, \frac{\varepsilon}{\gamma M}\right). \tag{A 2.1}$$

More explicitly, all values contained within this interval are feasible candidates for μ, but they will achieve the required precision using different numbers of iterations. By combining Equation 3.1 with the following two inequalities

$$f_\mu(\boldsymbol{\beta}_\mu^*) \leq f_\mu(\boldsymbol{\beta}^*) \quad \text{and} \quad f(\boldsymbol{\beta}^*) \leq f(\boldsymbol{\beta}_\mu^*),$$

we obtain

$$f_\mu(\boldsymbol{\beta}_\mu^*) \leq f(\boldsymbol{\beta}^*) \leq f_\mu(\boldsymbol{\beta}_\mu^*) + \underbrace{\mu\gamma M}_{<\varepsilon}, \tag{A 2.2}$$

and thus we obtain Equation A 2.1.

Note: Chen et al. [4] used a value for the smoothing parameter equal to $\varepsilon/2\gamma M$. This lies within the valid interval, specified in Equation A 2.1. However, as is illustrated in the main paper [5], it is possible to improve on this value in order to achieve convergence in fewer numbers of iterations.

By Equation 2.15 (using $\boldsymbol{\beta}^*$ instead of $\boldsymbol{\beta}_\mu^*$) and Equation 3.1 we write

$$\begin{aligned}\varepsilon = f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) &= \underbrace{f(\boldsymbol{\beta}^k) - f_\mu(\boldsymbol{\beta}^k)}_{\substack{\text{(A 2.2) at } \boldsymbol{\beta}^k, \\ \leq \gamma\mu M}} + \underbrace{f_\mu(\boldsymbol{\beta}^k) - f_\mu(\boldsymbol{\beta}^*)}_{\substack{\text{(2.15) at } \boldsymbol{\beta}^*, \\ \leq \frac{2}{t_\mu (k+1)^2}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}} + \underbrace{f_\mu(\boldsymbol{\beta}^*) - f(\boldsymbol{\beta}^*)}_{\substack{\text{(A 2.2) at } \boldsymbol{\beta}^*, \\ \leq 0}} \\ &\leq \mu\gamma M + \frac{2\left(L(\nabla(g)) + \gamma\frac{\|\boldsymbol{A}\|_2^2}{\mu}\right)}{(k+1)^2}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2. \end{aligned} \tag{A 2.3}$$

Then

$$\varepsilon - \mu\gamma M \leq \frac{2\left(L(\nabla(g)) + \gamma\frac{\|\boldsymbol{A}\|_2^2}{\mu}\right)}{(k+1)^2}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2.$$

Rearranging, we control the worst case number of required iterations by first posing

$$\frac{2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}{(k+1)^2} \geq \frac{\varepsilon - \gamma\mu M}{L(\nabla(g)) + \gamma\frac{\|\boldsymbol{A}\|_2^2}{\mu}} = \zeta(\mu),$$

expressed as a function of $\mu > 0$ and finding the $\mu \in (0, \varepsilon/\gamma M)$ that maximizes ζ. This will minimize the number of iterations required to achieve the desired precision. The derivative of this function is

$$\zeta'(\mu) = \frac{(\varepsilon - 2\mu\gamma M)(\mu L(\nabla(g)) + \gamma\|\boldsymbol{A}\|_2^2) - L(\nabla(g))(\mu\varepsilon - \mu^2\gamma M)}{(\mu L(\nabla(g)) + \gamma\|\boldsymbol{A}\|_2^2)^2}.$$

The maximum value of ζ for positive μ is at $\zeta'(\mu) = 0$, which occurs at

$$\mu_{opt}(\varepsilon) = \frac{-\gamma M\|\boldsymbol{A}\|_2^2 + \sqrt{(\gamma M\|\boldsymbol{A}\|_2^2)^2 + ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon}}{ML(\nabla(g))}. \tag{A 2.4}$$

This is the only positive μ because the associated 2nd degree polynomial has only one positive root.

A 2.2. Proof of Theorem 3: duality gap for the non-smooth problem

We begin by proving the first property of Equation 3.6 for all $\lambda \geq 0$. By using the Fenchel conjugate properties [2], we obtain

$$\psi^* = \left\{\max_{\boldsymbol{\alpha}\in\mathcal{K}}\psi_{\boldsymbol{\alpha}}\right\}^* \leq \inf_{\boldsymbol{\alpha}\in\mathcal{K}}\psi^*_{\boldsymbol{\alpha}} \leq \psi^*_{\boldsymbol{\alpha}(\boldsymbol{\beta}^k)} \equiv \psi^*_k. \tag{A 2.5}$$

Thus,

$$\begin{aligned}\widetilde{\text{GAP}}(\boldsymbol{\beta}^k) &\equiv f(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}^k) + \psi^*_k(-\boldsymbol{X}^\top\boldsymbol{\sigma}^k)\\ &\geq f(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}^k) + \psi^*(-\boldsymbol{X}^\top\boldsymbol{\sigma}^k)\\ &= \text{GAP}(\boldsymbol{\beta}^k)\\ &\geq f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*).\end{aligned}$$

Now we prove the second property of Equation 3.6. We first consider the case with $\lambda > 0$ and so use $\boldsymbol{\sigma}^k = \nabla l(\boldsymbol{X}\boldsymbol{\beta}^k)$ to compute the gap. We claim that at the optimum, *i.e.* at $\boldsymbol{\beta}^*$, we have

$$\psi^*_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}\big(-\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big) = \psi^*\big(-\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big). \tag{A 2.6}$$

Consequently, at $\boldsymbol{\beta}^*$, we would obtain

$$\widetilde{\text{GAP}}(\boldsymbol{\beta}^*) = \text{GAP}(\boldsymbol{\beta}^*) = 0. \tag{A 2.7}$$

In fact, using the definition of the Fenchel conjugate we have

$$\psi^*_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}\big(-\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big) = \max_{\boldsymbol{z}\in\mathbb{R}^p}\left\{\langle -\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)|\boldsymbol{z}\rangle - \frac{\lambda}{2}\|\boldsymbol{z}\|_2^2 - \kappa\|\boldsymbol{z}\|_1 - \gamma\langle\boldsymbol{\alpha}(\boldsymbol{\beta}^*)|\boldsymbol{A}\boldsymbol{z}\rangle\right\}.$$

The sub-differential optimality condition for this maximization problem holds at $\boldsymbol{\beta}^*$. Indeed, it is equivalent to the fact that the minimum of $f(\boldsymbol{\beta}^*)$ also minimizes $l(\boldsymbol{X}\boldsymbol{\beta}) + \psi_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}(\boldsymbol{\beta})$. This can easily be checked since $(\boldsymbol{\beta}^*, \boldsymbol{\alpha}(\boldsymbol{\beta}^*))$ is a saddle point [1] of the the min-max problem

$$f(\boldsymbol{\beta}^*) = \min_{\boldsymbol{\beta}\in\mathbb{R}^p}\max_{\boldsymbol{\alpha}\in\mathcal{K}}\big\{l(\boldsymbol{X}\boldsymbol{\beta}) + \psi_{\boldsymbol{\alpha}}(\boldsymbol{\beta})\big\}.$$

Now, we use $\boldsymbol{\beta}^*$ as a particular $\boldsymbol{z}$ and consecutively apply Equation 3.7 and the Fenchel-Young inequality (see Borwein and Lewis [2], Proposition 3.3.4) on the obtained inequality. The equality holds due to the optimality conditions satisfied by f on $\boldsymbol{\beta}^*$. We obtain

$$\begin{aligned}\psi^*_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}\big(-\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big) &= \Big\{\langle -\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)|\boldsymbol{\beta}^*\rangle - \psi(\boldsymbol{\beta}^*)\Big\}\\ &= \psi(\boldsymbol{\beta}^*) + \psi^*\big(-\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big) - \psi(\boldsymbol{\beta}^*)\\ &= \psi^*\big(-\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big).\end{aligned}$$

Therefore, we deduce Equation A 2.6 for $\lambda > 0$. Next we consider the case $\lambda = 0$. First, we claim that $\boldsymbol{\sigma}^k$ has, at the minimum $\boldsymbol{\beta}^*$, the same value as in the first case with $\lambda > 0$. Accordingly, Equation A 2.7 holds when $\lambda = 0$. We again use the fact that $\boldsymbol{\beta}^*$ minimizes $l(\boldsymbol{X}\boldsymbol{\beta}) + \psi_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}(\boldsymbol{\beta})$ to get

$$0 \in \boldsymbol{X}^\top\nabla l(\boldsymbol{X}\boldsymbol{\beta}^*) + \partial\psi_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}(\boldsymbol{\beta}^*) \equiv \boldsymbol{X}^\top\boldsymbol{\sigma}^* + \partial_{\kappa\|\cdot\|_1}(\boldsymbol{\beta}^*) + \boldsymbol{A}^\top\boldsymbol{\alpha}(\boldsymbol{\beta}^*).$$

Using the well known sub-differential of the ℓ_1 norm (see Bonnans, Gilbert and Lemarechal [1]), we deduce that for all $1 \le j \le p$

$$\Big|\big(\boldsymbol{X}^\top\boldsymbol{\sigma}^*\big)_j + \boldsymbol{s}_j\Big| \le \kappa.$$

So $\boldsymbol{k}_j^k = \big(\boldsymbol{X}^\top\boldsymbol{\sigma}^*\big)_j - \boldsymbol{s}_j$ and it follows from an easy and straight-forward computation that $\boldsymbol{\sigma}^* = \nabla l(\boldsymbol{X}\boldsymbol{\beta}^*)$ as when $\lambda > 0$ and so Equation A 2.6 holds for $\lambda = 0$.

Finally, we establish the Fenchel conjugate expression: first, we consider the case $\lambda > 0$ since ψ_k^* is always finite no scaling of $\boldsymbol{\sigma}$ is needed. In fact,

$$\begin{aligned}\psi_k^*(\boldsymbol{v}) &\equiv \max_{\boldsymbol{z}\in\mathbb{R}^p}\Big\{\langle\boldsymbol{v},\boldsymbol{z}\rangle - \psi_{\boldsymbol{\alpha}(\boldsymbol{\beta}^k)}(\boldsymbol{z})\Big\}\\ &= \sum_{j=1}^p\max_{z_j\in\mathbb{R}}\left\{z_j\Big(v_j - \gamma\big(\boldsymbol{A}^\top\boldsymbol{\alpha}(\boldsymbol{\beta}^k)\big)_j\Big) - \frac{\lambda}{2}z_j^2 - \kappa|z_j|\right\}\\ &= \frac{1}{2\lambda}\sum_{j=1}^p\left(\left[\Big|v_j - \gamma\big(\boldsymbol{A}^\top\boldsymbol{\alpha}(\boldsymbol{\beta}^k)\big)_j\Big| - \kappa\right]_+\right)^2,\end{aligned} \tag{A 2.8}$$

where $[\cdot]_+ = \max(0, \cdot)$.

When $\lambda = 0$, we check that, for all $j = 1, \dots, p$, we have

$$\max_{z_j \in \mathbb{R}} \left\{ z_j \Big(v_j - \gamma \big(\boldsymbol{A}^\top \boldsymbol{\alpha}(\boldsymbol{\beta}^k) \big)_j \Big) - \kappa |z_j| \right\} = \begin{cases} 0 & \text{if } \left| v_j - \gamma \big(\boldsymbol{A}^\top \boldsymbol{\alpha}(\boldsymbol{\beta}^k) \big)_j \right| \leq \kappa \\ +\infty & \text{if otherwise} \end{cases}$$

Thus, we need to change $\boldsymbol{\sigma}^k$ slightly, such that a new dual variable, denoted $\widetilde{\boldsymbol{\sigma}}^k$, satisfies $\psi_k^*(-\boldsymbol{X}^\top \widetilde{\boldsymbol{\sigma}}^k) < \infty$, while maintaining the other key properties in Equation A 2.6. Namely, we must obtain, for all $1 \leq j \leq p$, that

$$\left| \Big(-\boldsymbol{X}^\top \widetilde{\boldsymbol{\sigma}}^k \Big)_j - \gamma \big(\boldsymbol{A}^\top \boldsymbol{\alpha}(\boldsymbol{\beta}^k) \big)_j \Big) \right| < \kappa.$$

A straightforward way to achieve the aforementioned constraint would be to solve the linear system

$$\boldsymbol{X}^\top \widetilde{\boldsymbol{\sigma}} + \boldsymbol{s} = \kappa \mathbf{1} \tag{A 2.9}$$

as a function of the scaled dual variable $\widetilde{\boldsymbol{\sigma}}$. But that would also penalize the components of $\boldsymbol{\sigma}$ already fulfilling said constraint. In order to avoid over-scaling, we introduce a vector $\boldsymbol{k}$ to replace $\kappa \mathbf{1}$ on the right hand side of Equation A 2.9. The vector $\boldsymbol{k}$ is built such that, for all $j = 1, \dots, p$,

$$k_j = \operatorname{sign} \big((\boldsymbol{X}^\top \boldsymbol{\sigma})_j + s_j \big) \cdot \min \big(\kappa, |(\boldsymbol{X}^\top \boldsymbol{\sigma})_j + s_j| \big).$$

By construction, it has the two following properties:

(i) if $|(\boldsymbol{X}^\top \boldsymbol{\sigma})_j + s_j| \leq \kappa$, then σ_j remains unchanged,

(ii) otherwise, each component of $\boldsymbol{k}$ contains the sign of $(\boldsymbol{X}^\top \boldsymbol{\sigma})_j + s_j$, which allows us to fairly constrain the components of $\boldsymbol{\sigma}$ for which the difference is smaller than $-\kappa$ or larger than to κ.

A simple computation, assuming that $\boldsymbol{X}\boldsymbol{X}^\top$ is invertible, implies that the new dual variable can be computed in the following way

$$\widetilde{\boldsymbol{\sigma}} = (\boldsymbol{X}\boldsymbol{X}^\top)^{-1} \boldsymbol{X} (\boldsymbol{k} - \boldsymbol{s}).$$

A 2.3. Proof of Theorem 4: convergence of CONESTA

First, we recall from Algorithm 1 and Equation 3.6 that, for any positive integer i, if $\boldsymbol{\beta}_\mu^{i+1} = \text{FISTA}(\boldsymbol{\beta}_\mu^i, \mu^i, \varepsilon^i)$ then

$$f_{\mu^i}(\boldsymbol{\beta}_\mu^{i+1}) - f_{\mu^i}(\boldsymbol{\beta}_{\mu^i}^*) \leq \widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}_\mu^{i+1}) \leq \varepsilon^i. \tag{A 2.10}$$

We know from Equation 2.15 that if we apply FISTA, with any fixed $\mu > 0$, on the smoothed function, it will converge to the corresponding optimum $\boldsymbol{\beta}_\mu^*$. Consequently, $\widetilde{\text{GAP}}_{\mu^i}$ will be very small around the optimum, and thus satisfy any stopping criterion. Moreover, using the duality gap properties from Equation 3.6, the stopping rule in Algorithm 1 on Line 7 is now easy to check

by using $\widetilde{\text{GAP}}_\mu$ through the test if $\widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}^k) \leq \varepsilon^i$. Thus, Equation A 2.10 will hold at each iteration.

Next, we use Equation A 2.10 to establish the first claim. In fact, we have

$$\begin{aligned}
\varepsilon^{i+1} &= \tau \cdot \left(\mu^i \gamma M + \widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}_\mu^{i+1})\right) \\
&\leq \tau \cdot \left(\mu^i \gamma M + \varepsilon_\mu^i\right) \\
&= \tau \cdot \left(\mu^i \gamma M + \varepsilon^i - \mu^i \gamma M)\right) \\
&\leq \tau \cdot \varepsilon^i \\
&\leq \tau^i \cdot \varepsilon^0 \xrightarrow{i \to \infty} 0.
\end{aligned}$$

Next, we claim that

$$f(\boldsymbol{\beta}_\mu^i) - f(\boldsymbol{\beta}^*) \leq \varepsilon^i, \quad \forall i \in \mathbb{N}, \tag{A 2.11}$$

which involve the second statement (ii). Indeed, we know from Equation A 2.2 that

$$f_\mu(\boldsymbol{\beta}_\mu^*) - f(\boldsymbol{\beta}^*) \leq 0, \quad \forall \mu > 0.$$

It follows that

$$\begin{aligned}
f(\boldsymbol{\beta}_\mu^i) - f(\boldsymbol{\beta}^*) &= f(\boldsymbol{\beta}_\mu^i) - f_{\mu^i}(\boldsymbol{\beta}_\mu^i) \\
&\quad + f_{\mu^i}(\boldsymbol{\beta}_\mu^i) - f_{\mu^i}(\boldsymbol{\beta}_{\mu^i}^*) \\
&\quad + f_{\mu^i}(\boldsymbol{\beta}_{\mu^i}^*) - f(\boldsymbol{\beta}^*), \\
&\leq \mu^i \gamma M + f_{\mu^i}(\boldsymbol{\beta}_\mu^i) - f_{\mu^i}(\boldsymbol{\beta}_{\mu^i}^*), \\
&\leq \mu^i \gamma M + \widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}_\mu^i) \\
&\leq \mu^i \gamma M + \varepsilon_\mu^i \\
&= \mu^i \gamma M + \varepsilon^i - \mu^i \gamma M \\
&= \varepsilon^i.
\end{aligned} \tag{A 2.12}$$

Next, we consecutively prove statements (iii) and (iv) which will complete the proof. First we establish the convergence rate related to μ_{opt}. Note that this concerns the convergence of the non-smooth minimisation problem when applying the FISTA algorithm to the smoothed problem.

First, we use the upper bound established in Equation A 2.3 with the particular value of μ_{opt}. Simple calculations leads to the following estimation:

$$\begin{aligned}
f(\boldsymbol{\beta}_{\mu_{opt}}^k) - f(\boldsymbol{\beta}^*) \leq & \frac{\sqrt{8\|\boldsymbol{A}\|_2^2 M \gamma^2 \|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}}{k+1} \\
& + \frac{2L(\nabla(g))\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}{(k+1)^2}.
\end{aligned}$$

Then we seek the smallest value of k such that

$$\frac{\sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^0-\boldsymbol{\beta}^*\|_2^2}}{k+1}+\frac{2L(\nabla(g))\|\boldsymbol{\beta}^0-\boldsymbol{\beta}^*\|_2^2}{(k+1)^2}\leq\varepsilon.$$

A simple computation using a second order polynomial gives us that k needs to satisfy

$$\begin{aligned}k+1&\geq\frac{\sigma+\sqrt{\sigma^2+8L(\nabla(g))\varepsilon\|\boldsymbol{\beta}^0-\boldsymbol{\beta}^*\|_2^2}}{2\varepsilon}\\&\geq\frac{\sigma}{\varepsilon}+\frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^0-\boldsymbol{\beta}^*\|_2^2}}{\sqrt{\varepsilon}},\end{aligned}\tag{A 2.13}$$

where

$$\sigma=\sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^0-\boldsymbol{\beta}^*\|_2^2}.$$

This completes the proof of statement (iii) of the Theorem 4.

Finally, we consider the convergence rate with respect to the total number of iterations. This is equivalent to estimating the sum of the numbers of iterations, k_i, performed during the ith iteration loop using μ^i. First we estimate the maximum possible number of continuation steps, $i_{\max}$. In fact, using Equation A 2.12, we have

$$f(\boldsymbol{\beta}_\mu^i)-f(\boldsymbol{\beta}^*)<\varepsilon^i\leq\tau^i\cdot\varepsilon^0.\tag{A 2.14}$$

Thus, we conclude that

$$i_{\max}=\operatorname{int}\left(\frac{\log\left(\frac{\varepsilon}{\varepsilon^0}\right)}{\log(\tau)}\right),$$

where $\operatorname{int}(\cdot)$ is the integer part function. Now, we sum the iterations, k_i, with respect to i. From (iii) we get that

$$\begin{aligned}k_i&\geq\frac{\sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^i-\boldsymbol{\beta}^*\|_2^2}}{\varepsilon^i}+\frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^i-\boldsymbol{\beta}^*\|_2^2}}{\sqrt{\varepsilon^i}}\\&\geq\frac{\sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^i-\boldsymbol{\beta}^*\|_2^2}}{\tau^{i-1}\varepsilon^0}+\frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^i-\boldsymbol{\beta}^*\|_2^2}}{\sqrt{\tau^{i-1}\varepsilon^0}}.\end{aligned}\tag{A 2.15}$$

Thus, the total number of iterations, k, satisfies

$$k\geq\sum_{i=1}^{i_{\max}}\frac{\sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^i-\boldsymbol{\beta}^*\|_2^2}}{\tau^{i-1}\varepsilon^0}+\frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^i-\boldsymbol{\beta}^*\|_2^2}}{\sqrt{\tau^{i-1}\varepsilon^0}}.$$

Using the uniqueness of the minimum $\boldsymbol{\beta}^*$ and (ii), we obtain the convergence of the sequence $\boldsymbol{\beta}^i$ to $\boldsymbol{\beta}^*$. Hence $\|\boldsymbol{\beta}^i-\boldsymbol{\beta}^*\|_2^2$ is uniformly (with respect to i) bounded

by a constant $C(\boldsymbol{\beta}^0)$, that only depends on $\boldsymbol{\beta}^0$. For the sake of simplicity we use the following notations:

$$c_1 := \sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2 C(\boldsymbol{\beta}^0)} \text{ and } c_2 := \sqrt{2L(\nabla(g))C(\boldsymbol{\beta}^0)}.$$

Hence we obtain

$$\begin{aligned} k &\geq \sum_{i=1}^{i_{\max}} \frac{c_1}{\tau^{i-1}\varepsilon^0} + \frac{c_2}{\sqrt{\tau^{i-1}\varepsilon^0}} \\ &\geq \frac{c_1}{\varepsilon^0}\frac{1-(1/\tau)^{i_{\max}}}{1-\frac{1}{\tau}} + \frac{c_2}{\sqrt{\varepsilon^0}}\frac{1-(1/\sqrt{\tau})^{i_{\max}}}{1-\frac{1}{\sqrt{\tau}}} \end{aligned}$$

But since $\log(\tau) < 0$, we have:

$$\operatorname{int}\left(\frac{\log\left(\frac{\varepsilon}{\varepsilon^0}\right)}{\log\left(\tau\right)}\right) \leq \frac{\log\left(\varepsilon/\varepsilon^0\right)}{\log(\tau)},$$

so we get

$$\begin{aligned} 1-\left(\frac{1}{\tau}\right)^{\operatorname{int}\left(\frac{\log\left(\frac{\varepsilon}{\varepsilon^0}\right)}{\log\left(\tau\right)}\right)} &= 1-\exp\left(\operatorname{int}\left(\frac{\log\left(\frac{\varepsilon}{\varepsilon^0}\right)}{\log\left(\tau\right)}\right)\log\left(1/\tau\right)\right) \\ &\geq 1-\exp\left(\frac{\log\left(\frac{\varepsilon}{\varepsilon^0}\right)}{\log\left(\tau\right)}\log\left(1/\tau\right)\right) \\ &= 1-\exp\left(-\log\left(\varepsilon/\varepsilon^0\right)\right) \\ &= 1-\frac{\varepsilon^0}{\varepsilon}, \end{aligned}$$

and similarly we can establish that

$$1-\left(\frac{1}{\sqrt{\tau}}\right)^{\operatorname{int}\left(\frac{\log\left(\frac{\varepsilon}{\varepsilon^0}\right)}{\log\left(\sqrt{\tau}\right)}\right)} \geq 1-\frac{\varepsilon^0}{\varepsilon}.$$

Finally we deduce that

$$k \geq \frac{c_1}{\varepsilon^0(1-1/\tau)} + \frac{c_2}{\sqrt{\varepsilon^0}(1-1/\sqrt{\tau})} + \left(\frac{c_1}{1/\tau-1} + \frac{c_2}{1/\sqrt{\tau}-1}\right)\frac{1}{\varepsilon},$$

and hence, we conclude that in order to reach a precision ε, CONESTA must perform a number of iterations that is on the order of $\mathcal{O}(C/\varepsilon)$.

A 3. State-of-the-art algorithms

A 3.1. The excessive gap method

It can prove cumbersome to apply the excessive gap algorithm [6] to such a complex problem as linear regression with non-smooth penalties. In order to ease the reader's understanding of our implementation, we here explain the necessary steps of the algorithm as well as details about the required algebraic computations.

A 3.1.1. General framework

Let us first recall Equation 2.1, which describes the optimization problem under consideration, namely

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ g(\boldsymbol{\beta}) + \kappa h(\boldsymbol{\beta}) + \gamma s(\boldsymbol{\beta}) \right\}.$$

Following Nesterov [6, Section 1], since g in Equation 2.1 is a strongly convex function, we can apply the version of the excessive gap method with a $\mathcal{O}(1/k^2)$ rate of convergence toward the minimum of f, where k is the number of iterations [6, Theorem 7.6].

For the sake of completeness and notation, we recall the definition of strong convexity [1].

Definition 3. *If g is a strongly convex function on a convex set $\mathcal{K}$ then we have*

$$g(\boldsymbol{\beta}) \geq g(\boldsymbol{\beta}^*) + \sigma_g \frac{\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2^2}{2}, \quad \forall \boldsymbol{\beta} \in \mathcal{K},$$

where $\boldsymbol{\beta}^* \equiv \arg\min_{\boldsymbol{\beta}\in\mathcal{K}} \{g(\boldsymbol{\beta})\}$. The constant $\sigma_g > 0$ is called the strong convexity parameter of g.

In the excessive gap framework, f is regularized using the tools presented in Section 2.3, with the particularity that all of its non-smooth parts are regularized simultaneously. Therefore, $\kappa h + \gamma s$ are smoothed together. The smoothing parameters used in the context of the excessive gap method will be denoted ν in order to avoid any confusion. Consequently, the approximation of f used in the excessive gap method is denoted

$$f_\nu(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) + (\kappa h + \gamma s)_\nu(\boldsymbol{\beta}). \tag{A 3.1}$$

Under the hypothesis that the necessary condition for applying Nesterov's smoothing applies (see Equation 2.4), Equation 2.1 is expressed as a min-max problem.

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ g(\boldsymbol{\beta}) + \max_{\boldsymbol{\alpha}\in\mathcal{K}'} \langle \boldsymbol{\alpha} | \boldsymbol{A}'\boldsymbol{\beta} \rangle \right\}, \tag{A 3.2}$$

where we, for the sake of simplicity, let $\mathcal{K}' = \mathcal{K}_{\kappa h+\gamma s}$, and $\boldsymbol{A}' = \boldsymbol{A}_{\kappa h+\gamma s}$. With $\mathcal{K}'$ defined, we let the constant M' be equal to $\max_{\boldsymbol{\alpha}\in\mathcal{K}'} \|\boldsymbol{\alpha}\|_2^2/2$. Then, the saddle point theorem [1] allows us to write

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}\in\mathbb{R}^p} \max_{\boldsymbol{\alpha}\in\mathcal{K}'} \left\{ g(\boldsymbol{\beta}) + \langle \boldsymbol{\alpha} | \boldsymbol{A}'\boldsymbol{\beta} \rangle \right\} = \max_{\boldsymbol{\alpha}\in\mathcal{K}'} \min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ g(\boldsymbol{\beta}) + \langle \boldsymbol{\alpha} | \boldsymbol{A}'\boldsymbol{\beta} \rangle \right\}, \tag{A 3.3}$$

The saddle point theorem also allows us to define the dual objective function of the excessive gap method as

$$D_{EG}(\boldsymbol{\alpha}) = \min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ g(\boldsymbol{\beta}) + \langle \boldsymbol{\alpha} | \boldsymbol{A}'\boldsymbol{\beta} \rangle \right\}.$$

According to Nesterov [6, Lemma 7.1], D_{EG} is concave and differentiable with gradient

$$\nabla D_{EG}(\boldsymbol{\alpha}) = \boldsymbol{A}'\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha}),$$

where

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ g(\boldsymbol{\beta}) + \langle \boldsymbol{\alpha} | \boldsymbol{A}'\boldsymbol{\beta} \rangle \right\}.$$

Finally, before presenting the excessive gap method, we need to introduce an ancillary and original concept of Nesterov, namely the "gradient mapping".

Definition 4 (Gradient mapping)**.** *The gradient mapping associated with D_{EG} is defined as*

$$V(\boldsymbol{u}) = \arg\max_{\boldsymbol{v}\in\mathcal{K}'} \left\{ \langle \nabla D_{EG}(\boldsymbol{u}) | \boldsymbol{v} - \boldsymbol{u} \rangle - \frac{1}{2} L(\nabla D_{EG}) \left\| \boldsymbol{u} - \boldsymbol{v} \right\|_2^2 \right\},$$

with $L(\nabla D_{EG}) = \frac{\|\boldsymbol{A}'\|_2^2}{\sigma_g}$.

By using the aforementioned notation, the excessive gap method can be stated in a very synthetic way, as shown in Algorithm A 1. This algorithm achieves a convergence rate of $\mathcal{O}(1/k^2)$ when the differentiable part of the optimization problem is strongly convex.

Algorithm A 1 The excessive gap method

Input: $\widehat{\boldsymbol{\beta}}(\cdot)$, $\boldsymbol{\alpha}^*_\nu(\cdot)$, $V(\cdot)$, $L(\nabla D_{EG}) > 0$, $\varepsilon > 0$, $M' \geq 0$
Output: $\boldsymbol{\beta}^k$ such that $f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) < \varepsilon$

$\nu^0 = L(\nabla D_{EG})$
$\boldsymbol{\beta}^0 = \widehat{\boldsymbol{\beta}}(0)$
$\boldsymbol{\alpha}^0 = V(0)$
$k = 0$
loop
  $\tau^k = \frac{2}{k+3}$
  $\boldsymbol{u}^k = (1-\tau^k)\boldsymbol{\alpha}^k + \tau^k \boldsymbol{\alpha}^*_{\nu^k}(\boldsymbol{\beta}^k)$
  $\nu^{k+1} = (1-\tau^k)\nu^k$
  $\boldsymbol{\beta}^{k+1} = (1-\tau^k)\boldsymbol{\beta}^k + \tau^k \widehat{\boldsymbol{\beta}}(u^k)$
  $\boldsymbol{\alpha}^{k+1} = V(\boldsymbol{u}^k)$
  if $\nu^{k+1} M' < \varepsilon$ **then**
    break
  end if
  $k \leftarrow k+1$
end loop

Remark: it is necessary to smooth $\kappa h + \gamma s$ instead of just smoothing γs since a major step in the excessive gap method [6, Theorem 7.5] is the computation of $\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})$. If κh was not smoothed, we would have to use an iterative algorithm to approximate $\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})$ in each step. This would make it impossible to compute its exact value. To our knowledge, the inexact proximal method presented by Schmidt, Le Roux and Bach [7] has no equivalence in the excessive gap framework with an inexact $\widehat{\boldsymbol{\beta}}(\cdot)$.

A 3.1.2. Application to linear regression with elastic net and total variation penalties

Here we apply the excessive gap method to the regularized linear regression problem, expressed in Equation 2.2. To the authors' knowledge, the excessive gap method has never previously been used with this kind of function.

We will here detail the quantities that are essential for its implementation. These quantities are $\boldsymbol{A}'$, $\mathcal{K}'$, σ_g, $\boldsymbol{\alpha}^*(\cdot)$, $\widehat{\boldsymbol{\beta}}(\cdot)$, $L_{D_{EG}}$ and $V(\cdot)$.

First, we must separate f into two parts:

(i) A strongly convex smooth part:

$$\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_2^2+\frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2.$$

(ii) A non-smooth part that will be smoothed using Nesterov's technique (with smoothing constant ν):

$$\kappa\|\boldsymbol{\beta}\|_1+\gamma TV(\boldsymbol{\beta}).$$

We need to define the convex dual space and the associated linear operator in order to express the dual formulation of the non-smooth part of f in the form appropriate for the excessive gap method. The dual formulation of the non-smooth part of f is defined on the convex space

$$\mathcal{K}'=\{\boldsymbol{\alpha}\in\mathbb{R}^p,\|\boldsymbol{\alpha}\|_\infty\leq 1\}\times\prod_{(i,j,k)}\left\{\boldsymbol{\alpha}_{i,j,k}\in\mathbb{R}^3,\|\boldsymbol{\alpha}_{i,j,k}\|_2\leq 1\right\}.$$

The linear operator for the excessive gap method is

$$\boldsymbol{A}'=\begin{bmatrix}\kappa\boldsymbol{I}_p\\ \gamma\boldsymbol{A}_{TV}\end{bmatrix},$$

where $\boldsymbol{I}_p$ is the $p\times p$ identity matrix .

With $\mathcal{K}'$ and $\boldsymbol{A}'$ defined, the dual formulation of the non-smooth part of f is

$$\kappa\|\boldsymbol{\beta}\|_1+\gamma TV(\boldsymbol{\beta})=\max_{\boldsymbol{\alpha}\in\mathcal{K}'}\langle\boldsymbol{\alpha}|\boldsymbol{A}'\boldsymbol{\beta}\rangle.$$

It follows that the dual function D_{EG} is equal to

$$D_{EG}(\boldsymbol{\alpha})=\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left\{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_2^2+\frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2+\langle\boldsymbol{\alpha}|\boldsymbol{A}'\boldsymbol{\beta}\rangle\right\},$$

which leads to the expression of the optimal value for the primal variable

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})=\left(\boldsymbol{X}^\top\boldsymbol{X}+\lambda\boldsymbol{I}_p\right)^{-1}\left(\boldsymbol{X}^\top\boldsymbol{y}-\boldsymbol{A}'\boldsymbol{\alpha}\right).$$

The gradient of the dual function and its Lipschitz constant are

$$\nabla\left(D_{EG}(\boldsymbol{\alpha})\right)=\boldsymbol{A}'\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})\qquad\text{and}\qquad L(\nabla D_{EG})=\frac{\|\boldsymbol{A}'\|_2^2}{\lambda_{\min}(\boldsymbol{X}^\top\boldsymbol{X})+\lambda},$$

respectively, where $\lambda_{\min}(\boldsymbol{X}^\top \boldsymbol{X})$ is the smallest eigenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$. Finally, using Theorem 1, we can establish the expression for the optimal value of the dual variable

$$\boldsymbol{\alpha}_\nu^*(\boldsymbol{\beta}) = \text{proj}_{\mathcal{K}'}\left(\frac{1}{\nu}\boldsymbol{A}'\boldsymbol{\beta}\right),$$

and the gradient mapping

$$V(\boldsymbol{\alpha}) = \text{proj}_{\mathcal{K}'}\left(\boldsymbol{\alpha} + \frac{1}{L(\nabla D_{EG})}\boldsymbol{A}'\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})\right).$$

A 3.2. The Alternating Direction Method of Multipliers (ADMM)

Consider a problem on the form

$$\begin{aligned} &\text{minimize} \;\; g(\boldsymbol{x}) + h(\boldsymbol{z}), \\ &\text{subject to} \;\; \boldsymbol{x} = \boldsymbol{z}, \end{aligned}$$

where $g, h : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ are closed proper convex functions. Either or both of g and h may be non-smooth. The alternating direction method of multipliers (ADMM) [3], also known as *Douglas-Rachford splitting*, can be used to minimize this problem. The general ADMM algorithm is presented in Algorithm A 2.

Algorithm A 2 The Alternating Direction Method of Multipliers (ADMM)

Input: $g : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, $h : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$

1: **loop**
2: $\quad \boldsymbol{x}^{k+1} = \text{prox}_{\lambda_g g}(\boldsymbol{z}^k - \boldsymbol{u}^k)$
3: $\quad \boldsymbol{z}^{k+1} = \text{prox}_{\lambda_h h}(\boldsymbol{x}^{k+1} + \boldsymbol{u}^k)$
4: $\quad \boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}$
5: **end loop**

We recall the function in Equation 2.2 and restrict the structured penalty to a total variation penalty in the 1D setting. We aim to minimize the function

$$\begin{aligned} \min_{\beta \in \mathbb{R}^p]} f(\boldsymbol{\beta}) &= \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \kappa\|\boldsymbol{\beta}\|_1 + \gamma \sum_{G \in \mathcal{G}} \|\boldsymbol{A}_G \boldsymbol{\beta}_G\|_2 \\ &= \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \kappa \sum_{j=1}^{p-1} |\beta_j| + \gamma \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i|. \quad \text{(A 3.4)} \end{aligned}$$

We have adapted the ADMM-based solver described by Wahlberg et al. [8] by making the ridge regression loss function explicit; we have also added an ℓ_1 penalty to their derivation.

We rewrite the problem in Equation A 3.4 in the equivalent form

$$\begin{aligned} &\min_{\boldsymbol{\beta}, \boldsymbol{r}} \;\; \bar{f}(\boldsymbol{\beta}, \boldsymbol{r}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \kappa \sum_{j=1}^{p} |r_j| + \gamma \sum_{i=p+1}^{2p} |r_i|, \\ &\text{s.t.} \;\; (\boldsymbol{\beta}, \boldsymbol{r}) \in \mathcal{C} = \{(\boldsymbol{x}, \boldsymbol{r}) | r_j = x_j, r_i = x_{i+1} - x_i, \; j = 1, \ldots, p, \; i = p+1, \ldots, 2p\}. \end{aligned}$$

The ADMM equivalent form of this second problem is

$$\min_{\boldsymbol{\beta},\boldsymbol{r}} \ \widetilde{f}(\boldsymbol{\beta},\boldsymbol{r}) = \underbrace{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \kappa\sum_{j=1}^{p}|r_j| + \gamma\sum_{i=p+1}^{2p}|r_i|}_{g} + \underbrace{\iota_{\mathcal{C}}(\boldsymbol{z},\boldsymbol{s})}_{h}, \tag{A 3.5}$$

$$\text{s.t. } \beta_j = z_j, \quad j = 1,\dots,p$$
$$r_i = s_i, \quad i = 1,\dots,2p,$$

where $\iota_{\mathcal{C}}$ is the indicator function over the set $\mathcal{C}$, *i.e.*

$$\iota_{\mathcal{C}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in \mathcal{C}, \\ \infty & \text{otherwise.} \end{cases}$$

Equation A 3.5 is the problem that we will focus our attention on in this section.

The augmented Lagrangian of the problem in Equation A 3.5 is

$$\mathcal{L}(\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{r},\boldsymbol{s},\rho) = \widetilde{f}(\boldsymbol{\beta},\boldsymbol{r}) + \frac{\rho}{2}\big(\|\boldsymbol{\beta}-\boldsymbol{z}+\boldsymbol{u}\|_2^2 + \|\boldsymbol{r}-\boldsymbol{s}+\boldsymbol{t}\|_2^2\big), \tag{A 3.6}$$

where $\boldsymbol{u}$ and $\boldsymbol{t}$ are scaled dual variables associated with the constraints $\boldsymbol{\beta} = \boldsymbol{z}$ and $\boldsymbol{r} = \boldsymbol{s}$, respectively, and ρ is a regularization constant.

We note that $\boldsymbol{\beta}$ and $\boldsymbol{r}$ are unrelated in $\mathcal{L}$ and $\widetilde{f}$, and thus can be minimized separately. We write for $\boldsymbol{\beta}$ that

$$\boldsymbol{\beta}^{+} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left\{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{\beta}-\boldsymbol{z}+\boldsymbol{u}\|_2^2\right\}, \tag{A 3.7}$$

which we note is the proximal operator of $\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_2^2+\frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2$ at the point $\boldsymbol{z}-\boldsymbol{u}$. We solve this problem analytically as follows: the gradient of Equation A 3.7 with respect to $\boldsymbol{\beta}$ at the optimum is

$$\nabla_{\beta}\mathcal{L} = \boldsymbol{X}^{\top}(\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}) + \lambda\boldsymbol{\beta} + \rho(\boldsymbol{\beta}-\boldsymbol{z}+\boldsymbol{u}) = \boldsymbol{0},$$

and we solve for $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \big(\boldsymbol{X}^{\top}\boldsymbol{X} + (\lambda+\rho)\boldsymbol{I}_p\big)^{-1}\big(\boldsymbol{X}^{\top}\boldsymbol{y} + \rho(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)})\big).$$

For $\boldsymbol{r}$, we write

$$\begin{aligned} \boldsymbol{r}_{\ell_1}^{+} &= \arg\min_{\boldsymbol{r}}\left\{\kappa\sum_{j=1}^{p}|r_j| + \frac{\rho}{2}\sum_{j=1}^{p}(r_j - s_j + t_j)^2\right\} \\ &= \operatorname{prox}_{\frac{\kappa}{\rho}\|\cdot\|_1}(\boldsymbol{s}_{\ell_1} - \boldsymbol{t}_{\ell_1}). \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{r}_{TV}^{+} &= \arg\min_{\boldsymbol{r}} \left\{ \gamma \sum_{j=p+1}^{2p} |r_j| + \frac{\rho}{2} \sum_{j=p+1}^{2p} (r_j - s_j + t_j)^2 \right\} \\ &= \operatorname{prox}_{\frac{\gamma}{\rho}\|\cdot\|_1} (\boldsymbol{s}_{TV} - \boldsymbol{t}_{TV}), \end{aligned}$$

where $\boldsymbol{s}_{\ell_1}$ and $\boldsymbol{t}_{\ell_1}$ are the first p elements of $\boldsymbol{s}$ and $\boldsymbol{t}$, respectively; and $\boldsymbol{s}_{TV}$ and $\boldsymbol{t}_{TV}$ are the last p elements of $\boldsymbol{s}$ and $\boldsymbol{t}$, respectively. We can efficiently use the soft-thresholding operator to find the minima in these two cases. These two proximal operators correspond to Line 2 of Algorithm A 2.

The next step of the ADMM algorithm is to compute the proximal operator for h, which in our case is the projection onto the constraint set $\mathcal{C}$.

The projection

$$(\boldsymbol{z}, \boldsymbol{s}) = \operatorname{proj}_{\mathcal{C}}((\boldsymbol{w}, \boldsymbol{v})),$$

where $\boldsymbol{w} = \boldsymbol{\beta} + \boldsymbol{u}$ and $\boldsymbol{v} = \boldsymbol{r} + \boldsymbol{t}$, is computed by solving the following minimization problem

$$\begin{aligned} \min \ & \|\boldsymbol{z} - \boldsymbol{w}\|_2^2 + \|\boldsymbol{s} - \boldsymbol{v}\|_2^2 \\ \text{s.t.} \ & \boldsymbol{s} = \boldsymbol{A}\boldsymbol{z}, \end{aligned}$$

where

$$\boldsymbol{A} = \begin{matrix} \\ \ell_1 \\ \\ \\ \\ \\ \text{TV} \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ -1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

This problem is equivalent to

$$\min \ \|\boldsymbol{z} - \boldsymbol{w}\|_2^2 + \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{v}\|_2^2, \tag{A 3.8}$$

with only one variable $\boldsymbol{z}$.

We solve this problem analytically as follows: The gradient of Equation A 3.8 is at the optimum

$$\nabla\left(\|\boldsymbol{z} - \boldsymbol{w}\|_2^2 + \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{v}\|_2^2\right) = \boldsymbol{z} - \boldsymbol{w} + \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{z} - \boldsymbol{v}) = 0,$$

and we solve for $\boldsymbol{z}$ as

$$\boldsymbol{z} = (\boldsymbol{A}^\top\boldsymbol{A} + \boldsymbol{I}_p)^{-1}(\boldsymbol{A}^\top\boldsymbol{v} + \boldsymbol{w}).$$

We then compute $\boldsymbol{s} = \boldsymbol{A}\boldsymbol{z}$.

The proximal operator, on Line 3 in Algorithm A 2, thus corresponds to the projection

$$(\boldsymbol{z}^+, \boldsymbol{s}^+) = \text{proj}_{\mathcal{C}}((\boldsymbol{\beta}^+ + \boldsymbol{u}, \boldsymbol{r}^+ + \boldsymbol{t})).$$

Putting all parts together, the final algorithm is given in Algorithm A 3.

Algorithm A 3 Adapted ADMM algorithm

1: **loop**
2: $\quad \boldsymbol{\beta}^{k+1} = (\boldsymbol{X}^\top \boldsymbol{X} + (\kappa + \rho)\boldsymbol{I}_p)^{-1}(\boldsymbol{X}^\top \boldsymbol{y} + \rho(\boldsymbol{z}^k - \boldsymbol{u}^k))$
3: $\quad \boldsymbol{r}_{\ell_1}^{k+1} = \text{prox}_{\frac{\lambda}{\rho}\|\cdot\|_1}\left(\boldsymbol{s}_{\ell_1}^k - \boldsymbol{t}_{\ell_1}^k\right)$
4: $\quad \boldsymbol{r}_{TV}^{k+1} = \text{prox}_{\frac{\gamma}{\rho}\|\cdot\|_1}\left(\boldsymbol{s}_{TV}^k - \boldsymbol{t}_{TV}^k\right)$
5: $\quad \boldsymbol{z}^{k+1} = (\boldsymbol{A}^\top \boldsymbol{A} + \boldsymbol{I}_p)^{-1}(\boldsymbol{A}^\top(\boldsymbol{r}^{k+1} + \boldsymbol{t}^k) + (\boldsymbol{\beta}^{k+1} + \boldsymbol{u}^k))$
6: $\quad \boldsymbol{s}^{k+1} = \boldsymbol{A}\boldsymbol{z}^{k+1}$
7: $\quad \boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \boldsymbol{\beta}^{k+1} - \boldsymbol{z}^{k+1}$
8: $\quad \boldsymbol{t}^{k+1} = \boldsymbol{t}^k + \boldsymbol{r}^{k+1} - \boldsymbol{s}^{k+1}$
9: **end loop**

Remark: we note that the inverse on Line 2 can be computed fairly efficiently by using the singular value decomposition of $\boldsymbol{X}^\top \boldsymbol{X}$ and the Woodbury matrix identity. The major computational burden of this algorithm is found in Line 2.

Also, the linear system in Line 5 can be solved very efficiently by using the tridiagonal matrix algorithm (also called Thomas' algorithm). The solution can be obtained in $\mathcal{O}(p)$ time. We are able to do this here because of the particular (tridiagonal) form of the matrix $\boldsymbol{A}^\top \boldsymbol{A}$.

A 3.2.1. The penalty parameter ρ

As far as the authors are aware, the penalty parameter, ρ, in Equation A 3.6 is unknown, and finding good values for it is still an open problem.

We use the heuristics presented by Boyd et al. [3, Section 3.4.1], where the penalty parameter ρ^k is updated in each iteration. The aim of updating ρ^k as described below is to achieve improved practical convergence and to avoid having the performance depend on the choice of penalty parameter.

An approach to updating ρ^k is discussed by Boyd et al. [3] and is

$$\rho^{k+1} = \begin{cases} \tau^{\text{incr}}\rho^k, & \text{if } \|r^k\|_2 > \mu\|s^k\|_2, \\ \rho^k/\tau^{\text{decr}}, & \text{if } \|s^k\|_2 > \mu\|r^k\|_2, \\ \rho^k, & \text{otherwise,} \end{cases}$$

where $\mu > 1$, $\tau^{\text{incr}} > 1$ and $\tau^{\text{decr}} > 1$. Boyd et al. [3] proposed to use $\mu = 10$ and $\tau^{\text{incr}} = \tau^{\text{decr}} = 2$ and we employed the same parameters in the example simulations. The purpose of this update is to keep the primal and dual residual norms within a factor μ of each other.

A 3.3. The inexact proximal gradient method

In this section, we adapt the inexact proximal approach for solving Equation 2.1. We suppose that the non-smooth part, $\gamma s(\boldsymbol{\beta}) + \kappa h(\boldsymbol{\beta})$, satisfies Nesterov's assumption as stated in Equation 2.4; namely that

$$\gamma s(\boldsymbol{\beta}) + \kappa h(\boldsymbol{\beta}) \equiv \max_{\boldsymbol{\alpha} \in \mathcal{K}'} \langle \boldsymbol{\alpha} | \boldsymbol{A}' \boldsymbol{\beta} \rangle,$$

where $\boldsymbol{A}'$ and $\mathcal{K}'$ are the same as stated above in the section describing the excessive gap method.

The main step of the algorithm, when using a proximal gradient method, is to compute

$$\boldsymbol{\beta}^{k+1} \equiv \text{prox}_{th}(\boldsymbol{v}^k); \quad \text{where } \boldsymbol{v}^k \equiv \boldsymbol{\beta}^k - t \nabla g(\boldsymbol{\beta}^k), \text{ and } t = \frac{1}{L}.$$

In the inexact proximal gradient context, we want to approximate the proximal operator at each step k. We use Definition 2 and obtain a non-smooth minimization problem,

$$\text{prox}_{th}(\boldsymbol{v}^k) \equiv \arg\min_{\mathbf{u} \in \mathbb{R}^p} \left\{ \frac{1}{2} \|\mathbf{u} - \boldsymbol{v}^k\|_2^2 + th(\mathbf{u}) \right\}. \tag{A 3.9}$$

Following Schmidt, Le Roux and Bach [7], we are looking for a stopping criteria in the algorithm to come, and a precision $\varepsilon^k > 0$ such that

$$\frac{1}{2t} \left\| \widehat{\text{prox}}_{th}(\boldsymbol{v}^k) - \boldsymbol{v}^k \right\|_2^2 + h\left(\widehat{\text{prox}}_{th}(\boldsymbol{v}^k) \right) \leq \varepsilon^k + \min_{\mathbf{u} \in \mathbb{R}^p} \left\{ \frac{1}{2} \|\mathbf{u} - \boldsymbol{v}^k\|_2^2 + th(\mathbf{u}) \right\} \tag{A 3.10}$$

where $\widehat{\text{prox}}_{th}(\boldsymbol{v}^k)$ is the approximation of $\text{prox}_{th}(\boldsymbol{v}^k)$ obtained from a numerical approximation of Equation A 3.9, using any minimization algorithm. From Schmidt, Le Roux and Bach [7], we know that the sequence ε^k must decrease at least as fast as $1/k^4$, when using FISTA in order to keep its convergence rate and to converge to the minimum. So, in order to implement this approach, we need to define an iterative algorithm to approximate the proximal operator and a stopping criteria that allows us to satisfy Equation A 3.10.

Here we detail these two points. First, we compute the $\text{prox}_{th}(\boldsymbol{v})$ as

$$\begin{aligned}
\min_{\boldsymbol{u} \in \mathbb{R}^p} \{ \|\boldsymbol{u} - \boldsymbol{v}^k\|_2^2 + t \cdot h(\boldsymbol{u}) \} &= t \min_{\boldsymbol{u} \in \mathbb{R}^p} \left\{ \frac{1}{2t} \|\boldsymbol{u} - \boldsymbol{v}^k\|_2^2 + h(\boldsymbol{u}) \right\} \\
&= t \max_{\boldsymbol{\alpha} \in \mathcal{K}'} \min_{\boldsymbol{u} \in \mathbb{R}^p} \left\{ \langle \boldsymbol{\alpha} | \boldsymbol{A}' \boldsymbol{u} \rangle + \frac{1}{2t} \|\boldsymbol{u} - \boldsymbol{v}^k\|_2^2 \right\} \\
&= \frac{1}{2} \max_{\boldsymbol{\alpha} \in \mathcal{K}'} \left\{ \|\boldsymbol{v}^k\|_2^2 - \left\| \boldsymbol{v}^k - t \boldsymbol{A}'^{\top} \boldsymbol{\alpha} \right\|_2^2 \right\}.
\end{aligned}$$

We deduce that $\text{prox}_{th}(\boldsymbol{v}^k)$ can be approximated by minimizing

$$\boldsymbol{\alpha}_k^* \equiv \arg\min_{\boldsymbol{\alpha} \in \mathcal{K}'} \frac{1}{2} \|\boldsymbol{v}^k - t \boldsymbol{A}'^{\top} \boldsymbol{\alpha}\|_2^2 \tag{A 3.11}$$

using FISTA, and then compute

$$\widehat{\text{prox}}_{th}(\boldsymbol{v}^k) \equiv \boldsymbol{v}^k - t\boldsymbol{A}'^{\top}\widehat{\boldsymbol{\alpha}_k^*},$$

where $\widehat{\boldsymbol{\alpha}_k^*}$ is the approximation of $\boldsymbol{\alpha}_k^*$. The projection onto the compact $\mathcal{K}'$, that we need in order to use FISTA, was defined above in the section about the excessive gap method.

The gradient of the right-hand side of Equation A 3.11, with respect to $\boldsymbol{\alpha}$ at a fixed $\boldsymbol{v}^k$, is

$$\nabla_{\boldsymbol{\alpha}}\left(\frac{1}{2}\|\boldsymbol{v}^k - t\boldsymbol{A}'^{\top}\boldsymbol{\alpha}\|_2^2\right) = -t\boldsymbol{A}'(\boldsymbol{v}^k - t\boldsymbol{A}'^{\top}\boldsymbol{\alpha}),$$

and, the Lipschitz constant of the gradient is given by

$$\lambda_{max}(t^2\boldsymbol{A}'\boldsymbol{A}'^{\top}).$$

Finally, we define a stopping criterion for the FISTA loop by following Schmidt, Le Roux and Bach [7]. We use the min-max duality gap (see Bonnans, Gilbert and Lemarechal [1]) as follows. At step i of the inner FISTA loop when minimizing Equation A 3.11 at a fixed $\boldsymbol{v}^k$ (which is needed for the kth outer FISTA loop), we obtain an approximation $\boldsymbol{\alpha}_k^i$ of $\boldsymbol{\alpha}_k^*$; the corresponding dual variable is $\boldsymbol{z}_k^i \equiv \boldsymbol{v}^k - t\boldsymbol{A}'^{\top}\boldsymbol{\alpha}_k^i$. The duality gap is then computed as

$$\text{GAP}(\boldsymbol{z}_k^i) \equiv \frac{1}{2}\|\boldsymbol{z}_k^i - \boldsymbol{v}^k\|_2^2 + th(\boldsymbol{z}_k^i) - \frac{1}{2}\big(\|\boldsymbol{z}_k^i\|_2^2 - \|\boldsymbol{z}_k^i\|_2^2\big),$$

and finally the stopping criterion is that

$$\text{GAP}(\boldsymbol{z}_k^i) < \varepsilon^k < \frac{1}{k^4}.$$

A 4. ParsimonY: structured and sparse machine learning in Python

This section provides an simple example of the ParsimonY library applied on a large neuroimaging data set: $N = 199, P = 286\,217$ made of three un-penalized covariates (Age, Gender, Education) with 286 214 voxels of gray matter volume.

- To install parsimony, please visit: https://github.com/neurospin/pylearn-parsimony.
- To obtain the dataset, please visit: ftp://ftp.cea.fr/pub/unati/brainomics/papers/ols_nestv

ParsimonY is compliant with the scikit-learn API, only one supplementary step is required to transform an image mask into the linear operator denoted $\boldsymbol{A}$ throughout the paper.

```
import numpy as np
import os
```

```
import parsimony.functions.nesterov.tv as tv
import parsimony.estimators as estimators
import parsimony.functions.nesterov as nesterov
import nibabel

# Assume that the data set X, y is such that:
# - X: centered and scaled data of shape = (199, 286217):
#   Age + Gender + Education + 286 214 voxels.
#   3 first columns of X are un-penalized covariates => penalty_start = 3
#   Omit if no covariates; set to 1 with one covariate (such intercept) etc.
# - y: target vector of shape (199, 1)
penalty_start = 3

mask_ima = nibabel.load("mask.nii")
Atv = tv.linear_operator_from_mask(mask_ima.get_data())

# Global penalty of 0.01 * 1/3 of l1, 1/3 of l2 and 1/3 of tv
lambda_l1, lambda_l2, lambda_tv = 0.01 * np.array([0.3335, 0.3335, 0.333])
penalty_start=penalty_start
estimator = estimators.LinearRegressionL1L2TV(
    lambda_l1, lambda_l2,  lambda_tv, A=Atv, penalty_start=penalty_start)

# Fit the model
estimator.fit(X, y)

# Save weights map into nifti image
weight_arr = np.zeros(mask_ima.get_data().shape)
weight_arr[mask_ima.get_data() !=0 ] = estimator.beta.ravel()[penalty_start:]
weight_nii = nibabel.Nifti1Image(weight_arr, affine=mask_ima.get_affine())
weight_nii.to_filename("weight.nii")
```

References

[1] Bonnans, J. F., Gilbert, J. C. and Lemarechal, C. (2006). *Numerical Optimization: Theoretical and Practical Aspects*, 2nd ed. Springer-Verlag Berlin and Heidelberg GmbH & Co. K.

[2] Borwein, J. M. and Lewis, A. S. (2006). *Convex Analysis and Nonlinear Optimization: Theory and Examples. CMS Books in Mathematics*. Springer.

[3] Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* **3** 1–122.

[4] Chen, X., Lin, Q., Kim, S., Carbonell, J. G. and Xing, E. P. (2012). Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics* **6** 719–752.

[5] Hadj-Selem, F., Lofstedt, T., Frouin, V., Guillemot, V. and Duchesnay, E. (2016). An Iterative Smoothing Algorithm for Regression with Structured Sparsity. *arXiv:1605.09658 [stat]*. arXiv: 1605.09658.

[6] Nesterov, Y. (2005). Excessive Gap Technique in Nonsmooth Convex Minimization. *SIAM Journal on Optimization* **16** 235–249.

[7] Schmidt, M., Le Roux, N. and Bach, F. (2011). Convergence Rates of Inexact Proximal-Gradient Methods for Convex Optimization. In *NIPS'11 - 25 th Annual Conference on Neural Information Processing Systems*.

[8] Wahlberg, B., Boyd, S., Annergren, M. and Wang, Y. (2012). An ADMM Algorithm for a Class of Total Variation Regularized Estimation Problems. *ArXiv e-prints*.