Supplementary material: Continuation of Nesterov's Smoothing for Regression with Structured Sparsity in High-Dimensional Neuroimaging

Fouad Hadj-Selem, Tommy Löfstedt, Elvis Dohmatob, Vincent Frouin, Mathieu Dubois, Vincent Guillemot, Edouard Duchesnay, Tommy Lofstedt

Supplementary material: Continuation of Nesterov's Smoothing for Regression with Structured Sparsity in High-Dimensional Neuroimaging

Fouad Hadj-Selem[*,1,3], **Tommy Löfstedt**[*,1,4], **Elvis Dohmatob**[1,2], **Vincent Frouin**[1], **Mathieu Dubois**[1], **Vincent Guillemot**[1], **Edouard Duchesnay**[*,1]

[1] *NeuroSpin - CEA, Université Paris-Saclay - France*

[2] *PARIETAL Team, INRIA / CEA, Université Paris-Saclay- France.*

[3] *Energy Transition Institute: VeDeCoM - France.*

[4] *Department of Radiation Sciences, Umeå University, Umeå - Sweden.*

This document contains proofs and supplementary details for the paper "Continuation of Nesterov's Smoothing for Regression with Structured Sparsity in High-Dimensional Neuroimaging". All sections and equation numbers in this supplementary document are preceded by the letters *SM*, to distinguish them from those from the main paper.

After a brief introduction of the addressed optimization problem, the background section SM 2 provides the definitions that are used as the foundations of our contribution. Then, we present an extensive review of the state-of-the-art solvers that points out their limitations justifying our proposition (the CONESTA solver).

Section SM 3 provides the proofs that support our contribution: (i) for the duality gap (SM 3.2); (ii) for the optimal smoothing parameter (SM 3.2); (iii) for the convergence rate of CONESTA (SM 3.3).

Section SM 4 provides supplementary information about the MRI experimental data and Python the ParsimonY library (SM 5)

Finally sec. SM 6 presents the technical details of solvers used in the comparison studies.

*Contributed equally and corresponding authors. e-mail: fouad.hadjselem@vedecom.fr; tommy.lofstedt@umu.se; edouard.duchesnay@cea.fr

†Alzheimer's Disease Neuroimaging Initiative: Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

Contents

SM 1 Introduction . . . 3
SM 2 Background . . . 3
SM 2.1 Definitions . . . 3
SM 2.1.1 Lipschitz continuous gradient . . . 3
SM 2.1.2 Proximal operator . . . 4
SM 2.1.3 FISTA . . . 4
SM 2.2 State-of-the-art solvers . . . 4
SM 2.2.1 Inexact proximal-gradient . . . 4
SM 2.2.2 Primal-dual . . . 5
SM 2.2.3 Excessive gap . . . 6
SM 2.2.4 Smoothing proximal gradient . . . 6
SM 2.2.5 ADMM . . . 6
SM 2.2.6 PRISMA . . . 7
SM 3 Proofs . . . 7
SM 3.1 Duality gap with proofs . . . 7
SM 3.1.1 Introduction . . . 7
SM 3.1.2 Proof of the duality gap for the non-smooth problem . . . 9
SM 3.1.3 Proof of the duality gap for the smoothed problem (Theorem 1) . . . 12
SM 3.2 Proof of the optimal smoothing parameter, μ (Theorem 2) 13
SM 3.3 Proof of the convergence rate of CONESTA (Theorem 3) 14
SM 3.3.1 Proof of statement (i) . . . 15
SM 3.3.2 Proof of statement (ii) . . . 16
SM 3.3.3 Proof of statement (iii) . . . 16
SM 3.3.4 Proof of statement (iv) . . . 19
SM 4 Experiments on a structural MRI data set . . . 20
SM 4.1 MRI data acquisition and processing . . . 20
SM 4.2 Effect of the τ parameter on the convergence speed of CONESTA . . . 21
SM 4.2.1 Required precision and its gap estimate . . . 21
SM 5 ParsimonY: Structured and sparse machine learning in Python . 22
SM 6 Technical details of solvers used in the comparison studies . . . 24
SM 6.1 Smoothing proximal gradient or FISTA with fixed μ . . 24
SM 6.2 The excessive gap method . . . 24
SM 6.3 The Alternating Direction Method of Multipliers (ADMM) 27
SM 6.4 The Inexact proximal gradient method . . . 31
References . . . 33

SM 1. Introduction

This supplementary document provides details about the minimization of the non-smooth convex function

$$f(\boldsymbol{\beta}) = \overbrace{\underbrace{\mathcal{L}(\boldsymbol{\beta}) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2}_{g(\boldsymbol{\beta})}}^{\text{smooth}} + \overbrace{\underbrace{\kappa\,\|\boldsymbol{\beta}\|_1}_{h(\boldsymbol{\beta})} + \gamma \underbrace{\sum_{i,j,k} \|\boldsymbol{A}_{\phi(i,j,k)}\boldsymbol{\beta}\|_2}_{s(\boldsymbol{\beta})}}^{\text{non-smooth}}, \tag{SM 1.1}$$

where $\boldsymbol{\beta}$ is the vector of parameters to be estimated, g is the sum of a differentiable loss, *e.g.*, the least-squares loss: $\mathcal{L}(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$, and a ridge penalty; h is a sparsity-inducing penalty whose proximal operator is known, *e.g.*, the ℓ_1 penalty; and s is a complex penalty on the structure of the input variables, for which we either do not know the proximal operator, or for which the proximal operator is too expensive to compute.

Our proposed solution to this program is based on Nesterov's smoothing method, where we are minimizing an auxiliary (smoothed) function, closely related to Eq. SM 1.1, and which is

$$f_\mu(\boldsymbol{\beta}) = \overbrace{\underbrace{\mathcal{L}(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_2^2}_{g(\boldsymbol{\beta})} + \gamma \underbrace{\left\{\boldsymbol{\alpha}_\mu^*(\boldsymbol{\beta})^\top \boldsymbol{A}\boldsymbol{\beta} - \frac{\mu}{2}\|\boldsymbol{\alpha}^*\|_2^2\right\}}_{s_\mu(\boldsymbol{\beta})}}^{\text{smooth}} + \overbrace{\kappa \underbrace{\|\boldsymbol{\beta}\|_1}_{h(\boldsymbol{\beta})}}^{\text{non-smooth}}, \tag{SM 1.2}$$

where all the quantities are defined in Sec. II of the main paper.

SM 2. Background

SM 2.1. Definitions

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

SM 2.1.1. Lipschitz continuous gradient

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*

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

where $L(\nabla(f))$ is the smallest positive real value satisfying this inequality.

SM 2.1.2. 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 [4]. The proximal operator (or proximal mapping) $\text{prox}_h(x)$: $\mathbb{R}^P \to \mathbb{R}^P$ is then defined as*

$$\text{prox}_h(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{u}\in\mathbb{R}^P} \left\{ \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{\beta}\|_2^2 + h(\boldsymbol{u}) \right\}, \tag{SM 2.1}$$

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

$$\text{prox}_{th}(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{u}\in\mathbb{R}^P} \left\{ \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{\beta}\|_2^2 + th(\boldsymbol{u}) \right\}, \tag{SM 2.2}$$

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

SM 2.1.3. FISTA

We briefly summarize some information about FISTA that can also be found in [3].

FISTA convergence rate Since we do not make the assumption that g in Eq. SM 1.1 is strongly convex, the convergence rate of FISTA, when applied to Eq. SM 1.2, is governed by the expression [3]

$$f_\mu(\boldsymbol{\beta}^k) - f_\mu(\boldsymbol{\beta}_\mu^*) \leq \frac{2}{t_\mu(k+1)^2}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}_\mu^*\|_2^2, \tag{SM 2.3}$$

where k is the iteration counter.

SM 2.2. State-of-the-art solvers

Here, we provide an overview and mention some limitations of the current state-of-the-art solvers that can be used to address problem Eq. SM 1.1. Theoretical foundations and technical details of the solvers used in the comparison studies can be found in Sec. SM 6.

SM 2.2.1. Inexact proximal-gradient

Schmidt, Le Roux and Bach [20] gave a general sufficient condition for applying *inexact* proximal gradient algorithms where the proximal operator (or the gradient) is approximated. They established a sufficient condition for the approximation of the unknown proximal operator to be applied at each step of the proximal gradient algorithm such that the optimal convergence speed is maintained.

This method can be used in the application of complex penalties such as total variation (TV) or group lasso in linear regression. The method solves a subproblem that finds an approximation to the proximal operator. Details are provided in Sec. SM 2.1.2. The authors of [21, 11] implemented this method in the case of TV-regularized problems (like the ones considered in the present paper), and showed that it outperforms splitting and primal-dual methods.

However, in the case of high-dimensional problems (like those we are concerned with here), solving a subproblem such as approximating the proximal operator of g is potentially a very time-consuming process. Indeed, to ensure convergence, the precision of the inner loop that solves the subproblem must decrease as $\mathcal{O}\big(1/k^{4+\delta}\big)$ for any $\delta > 0$ (where k is the number of outer iterations) [20, Proposition 2]. Therefore, if we target a global precision of $\varepsilon \leq 10^{-3}$, using Eq. SM 2.3, we see that this would require on the order of $k \approx 31$ outer FISTA iterations [3]. Then, according to [20], the required precision in each inner loop, ε_k, should be smaller than 10^{-6} ($\approx 1/31^4$) leading to, at most [3], $\sqrt{1/10^{-6}} = 10^3$ iterations to solve the inner approximation problem. Experiments performed in this study shown that in practice $\approx 10^2$ iterations were usually necessary to solve the approximation problem after few dozens ≈ 30 of outer FISTA iterations.

Moreover, it could be argued that iterations of the the inner loop are fast since they do not involve computation on the full data $\boldsymbol{X}$ but only the $\boldsymbol{\beta}$ vector. However, if at least hundreds of iterations are required to solve the approximation problem, this will impose a real practical limitation for the use of Inexact FISTA for high-dimensional problems, such as in large neuroimaging or genetic data with small N and large P.

Therefore, the inexact proximal gradient algorithm is suitable for problems with moderate dimensionality, where the required precision will not involve a large number of expensive iterations of the outer FISTA loop [12].

With a loss of generality (specific fast implementation for 1D/2D/3D array), this problem can be alleviated by developing a specific optimized code to speed-up the computation of the proximal operator of given structured penalty. In contrast, the method that we propose here, CONESTA, does not suffer from this limitation since the approximation is not found numerically, and so no inner loop is needed. Moreover, the smoothing approach offers a generic framework in which a large range of non-smooth convex structured penalties can be minimized without computing their proximal operators. Moreover, the conducted comparison study demonstrated that CONESTA is much faster than the *Inexact* proximal gradient method in terms of the convergence time.

SM 2.2.2. Primal-dual

Chambolle and Pock [7] proposed a general primal-dual method that minimizes a loss function with two different penalties, such as *e.g.* TV and ℓ_1 [14]. They consider a function in the form $g(Kx) + h(x)$ where K is a linear operator and proved an optimal convergence rate for such non-smooth objective functions.

However they assume to have access to the proximal operators of both the smooth function, g, and the non-smooth one, h. This is the main shortcoming of this method, as it would require the approximation of said proximal operators when they are not available, like in the case with logistic regression, for instance. It would then require an inexact approach like the one for proximal gradient methods developed by Schmidt, Le Roux and Bach [20] and discussed above. To the best of our knowledge, the inexact issue for this method is an open problem.

SM 2.2.3. Excessive gap

Nesterov [17] presented an optimal primal-dual method called the excessive gap method, or the excessive gap technique. As we will show later, this method can be used to minimize Eq. SM 1.1 with optimal convergence rate. However, it shares the same shortcomings as the method by Chambolle and Pock [7]. In fact, a necessary step in the excessive gap algorithm is the computation of $\widehat{\boldsymbol{\beta}}$, the primal variable that corresponds to a particular dual variable, $\boldsymbol{u}$ (see Sec. SM 6.2). As far as the authors know, there is no explicit expression for this function when dealing with the general case of any convex loss function. An explicit application of the excessive gap method can be found in [8] where the authors applied it to canonical correlation analysis with group and fused lasso penalties. A problem with the excessive gap method (illustrated in [8] and reiterated below) is that it imposes a smoothing of all non-smooth parts, and in particular of the ℓ_1 penalty. This implies that the found solution will not be strictly sparse.

SM 2.2.4. Smoothing proximal gradient

Nesterov's smoothing can also be used in conjunction with the proximal gradient algorithm [9]. The main issue of this approach is that an accurate solution, with a small smoothing parameter, results in a slow convergence. In Sec. II we provide more details on this approach, since resolving this issue is one of the main contributions of this paper.

SM 2.2.5. ADMM

The alternating direction method of multipliers (ADMM) [6] is commonly used to minimize the sum of two convex functions. It can be adapted to the loss function of interest in this paper, Eq. SM 1.1, but has some drawbacks in this context. In particular, it suffers from the same shortcoming as the method discussed above by Chambolle and Pock [7]. In fact, when dealing with a smooth loss function that has an unknown proximal operator, the proximal operator would have to be approximated numerically.

Another computational limitation of ADMM when working with the ordinary least-squares loss and the ℓ_1-norm and TV penalties, is that each update involves

solving two linear systems, or, equivalently, computing the inversion of two large $p \times p$ linear operators, possibly ill-conditioned. In the case of TV and ℓ_1 penalties, this is not a problem, because of the particular form of the resulting linear operator. Yet for more general linear operators, solving the associated linear systems would quickly become intractable, in particular for large p.

Moreover, the regularization parameter, ρ, in the associated augmented Lagrangian function is difficult to set (this is still an open problem), and even though ADMM converges for any value of this parameter, under mild conditions, the convergence rate depends heavily on it. We have employed some heuristics for selecting this parameter. This issue, mentioned in [11, 20], is discussed in Sec. SM 6.3 along with the details of our implementation of ADMM as described in [22]. For the sake of method comparison, we had to generalize it slightly and have added an ℓ_1 penalty. We note that this kind of modification would not be as trivial with general complex penalties.

SM 2.2.6. PRISMA

PRISMA [19] is a continuation algorithm for minimizing a convex objective function that decomposes into three parts: a smooth part, a simple non-smooth Lipschitz continuous part, and a simple non-smooth non-Lipschitz continuous part. They use a smoothing strategy similar to that used in this paper. The main limitation is that the two different penalties have to be simple such that their proximal operators are explicit (see Algorithm 1 in [19]). Thus, as there is no inexact approach that allows to approximate any unknown proximal operator, while preserving the convergence, we can not apply PRISMA in a rigorous way when dealing with group lasso or TV.

Our proposed continuation algorithm addresses the two main aforementioned deficiencies. Indeed, CONESTA (i) is relevant in the context of any smooth convex loss function because it only requires the computation of the gradient and (ii) estimates weights that are strictly sparse because it does not require smoothing the sparsity-inducing penalties. Additionally, CONESTA does not require solving any linear systems in P dimensions, or inverting very large matrices ($\boldsymbol{X}\boldsymbol{X}^\top$ is inverted in the gap, but is assumed to be small in the $N \ll P$ paradigm), and can easily be applied with a variety of convex smooth loss functions and many different complex convex penalties.

SM 3. Proofs

SM 3.1. Duality gap with proofs

SM 3.1.1. Introduction

Duality formulations are often used to control the achieved precision when minimizing convex functions. They can be used to provide an estimation of the error $f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}^*)$, for any $\boldsymbol{\beta}$, without knowing the minimum $f(\boldsymbol{\beta}^*)$, which we

never know in practice. The duality gap is the cornerstone of CONESTA (see Algorithm 2), and it is used three times:

1. In the ith CONESTA iteration, as a way to estimate the current error $f(\boldsymbol{\beta}^i) - f(\boldsymbol{\beta}^*)$. The error will be estimated using the gap of the smoothed problem. This is justified in Sec. III. This value is then used to deduce all the other parameters for the next application of FISTA. The next desired precision and the smoothing parameter, μ^i, are derived from this value.
2. As the stopping criterion in the inner FISTA loop. The criterion will be such that FISTA will stop as soon as the current precision is achieved using the current smoothing parameter, μ^i. This prevents non-essential convergence toward the approximated (smoothed) objective function.
3. Finally, as the global stopping criterion within CONESTA. This will guarantee that the obtained approximation of the minimum, $\boldsymbol{\beta}^i$, satisfies $f(\boldsymbol{\beta}^i) - f(\boldsymbol{\beta}^*) < \varepsilon$ at convergence.

We first establish an expression of the duality gap for the problem in Eq. SM 1.1. Next, we consider the smoothed version in Eq. SM 1.2.

The Fenchel duality can be used as in [16] to rewrite the objective function in Eq. SM 1.1 as

$$f(\boldsymbol{\beta}) = \underbrace{\frac{1}{2}\left\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\right\|_2^2}_{l(\boldsymbol{X}\boldsymbol{\beta})} + \underbrace{\frac{\lambda}{2}\left\|\boldsymbol{\beta}\right\|_2^2 + \kappa\left\|\boldsymbol{\beta}\right\|_1 + \gamma \max_{\boldsymbol{\alpha}\in\mathcal{K}} \langle \boldsymbol{\alpha}, A\boldsymbol{\beta}\rangle}_{\Omega(\boldsymbol{\beta})} \equiv l(\boldsymbol{X}\boldsymbol{\beta}) + \Omega(\boldsymbol{\beta}), \tag{SM 3.1}$$

where we used a dual norm formulation for the complex penalty $s(\boldsymbol{\beta})$ of Eq. SM 1.1. Note that the squared loss is expressed as a function of $\boldsymbol{X}\boldsymbol{\beta}$ using $l(\boldsymbol{z}) \equiv \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|_2^2$. The rules from classical Fenchel duality [5] provide the duality gap for the problem in Eq. SM 3.1 at the current value of $\boldsymbol{\beta}^k$

$$\textsc{Gap}(\boldsymbol{\beta}^k) \equiv f(\boldsymbol{\beta}^k) + l^*\big(\boldsymbol{\sigma}(\boldsymbol{\beta}^k)\big) + \Omega^*\big(-\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^k)\big), \tag{SM 3.2}$$

where l^* and Ω^* are the Fenchel conjugates of l and Ω, respectively. It is straightforward to show that l^* can be expressed as $l^*(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{z}\|_2^2 + \langle \boldsymbol{z}, \boldsymbol{y}\rangle$. Following Mairal [16], the dual variable $\boldsymbol{\sigma}(\boldsymbol{\beta}^k)$ given the primal variable $\boldsymbol{\beta}^k$ can be computed as $\boldsymbol{\sigma}(\boldsymbol{\beta}^k) \equiv \nabla l(\boldsymbol{X}\boldsymbol{\beta}^k)$. The duality gap is finite, it vanishes at the minimum and provides an estimate of the difference with the optimal value of the objective function. The duality gap has the following properties:

$$\begin{aligned} \textsc{Gap}(\boldsymbol{\beta}^k) \geq f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) &\geq 0; \\ \textsc{Gap}(\boldsymbol{\beta}^*) &= 0. \end{aligned} \tag{SM 3.3}$$

This requires computing l^* and Ω^*. However, to the best of our knowledge, there is no explicit expression for Ω^* when using a complex penalty such as TV or group lasso. Therefore, we use an approximation that maintains the properties

stated in Eq. SM 3.3. Lets reformulate the penalty term $\Omega(\boldsymbol{\beta})$:

$$\begin{aligned}\Omega(\boldsymbol{\beta}) &= \frac{\lambda}{2}\left\|\boldsymbol{\beta}\right\|_2^2 + \kappa\left\|\boldsymbol{\beta}\right\|_1 + \gamma \max_{\boldsymbol{\alpha}\in\mathcal{K}} \langle \boldsymbol{\alpha}, \boldsymbol{A}\boldsymbol{\beta}\rangle \\ &= \max_{\boldsymbol{\alpha}\in\mathcal{K}} \left\{ \frac{\lambda}{2}\left\|\boldsymbol{\beta}\right\|_2^2 + \kappa\left\|\boldsymbol{\beta}\right\|_1 + \gamma\langle \boldsymbol{\alpha}, \boldsymbol{A}\boldsymbol{\beta}\rangle \right\} \\ &= \max_{\boldsymbol{\alpha}\in\mathcal{K}} \Omega_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) \end{aligned} \tag{SM 3.4}$$

Then, we approximate the Fenchel conjugate of Ω by that of $\Omega_k \equiv \Omega_{\boldsymbol{\alpha}(\boldsymbol{\beta}^k)}$, which is a local approximation of Ω at the current $\boldsymbol{\beta}^k$. Next, we define an approximation of the gap by explicitly computing Ω_k^*. Note that, in the following, we also address the specific situation where $\lambda = 0$, that leads to undefined Ω_k^*, since λ appears in the denominator. Therefore, we slightly change it in order to obtain feasible values. An explicit expression for the duality gap is presented in the following theorem.

SM 3.1.2. Proof of the duality gap for the non-smooth problem

Theorem *Let λ, κ and γ be non-negative real numbers. The following estimation of the gap satisfies Eq. SM 3.3:*

$$\widetilde{\text{GAP}}(\boldsymbol{\beta}^k) \equiv f(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}(\boldsymbol{\beta}^k)) + \Omega_k^*(-\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^k)) \tag{SM 3.5}$$

with $\Omega_k^*(\boldsymbol{v})$ being the Fenchel conjugate:

$$\Omega_k^*(\boldsymbol{v}) \equiv \begin{cases} \frac{1}{2\lambda}\sum_{j=1}^P \left(\left[|v_j - s_j^k| - \kappa \right]_+ \right)^2 & \text{if } \lambda > 0, \\ 0 & \text{if } \lambda = 0, \end{cases}$$

and dual variable

$$\boldsymbol{\sigma}(\boldsymbol{\beta}^k) \equiv \begin{cases} \nabla l(\boldsymbol{X}\boldsymbol{\beta}^k) & \text{if } \lambda > 0 \\ (\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{X}(\boldsymbol{k}^k - \boldsymbol{s}^k) & \text{if } \lambda = 0. \end{cases}$$

The vectors $\boldsymbol{s}^k, \boldsymbol{k}^k \in \mathbb{R}^P$ are given by

$$\boldsymbol{s}^k = \gamma \boldsymbol{A}^\top \boldsymbol{\alpha}(\boldsymbol{\beta}^k)$$

and the elements of $\boldsymbol{k}^k$ are

$$k_j^k = \text{sign}\Big(\big(\boldsymbol{X}^\top \nabla l(\boldsymbol{X}\boldsymbol{\beta}^k)\big)_j + s_j^k \Big) \cdot \min\Big(\kappa, \big| \big(\boldsymbol{X}^\top \nabla l(\boldsymbol{X}\boldsymbol{\beta}^k)\big)_j + s_j^k \big| \Big),$$

for $j = 1, \ldots, P$.

Remark 1: It is worth noting that in spite of the massive use of $\boldsymbol{X}$ and $\boldsymbol{A}$ in this duality gap computation, it does not limit this approach and the obtained formula to the least-squares loss or the elastic net and TV penalties. In fact, the

very same expression holds for more general models with different $\boldsymbol{X}$, $\boldsymbol{A}$ and l. The only requirement is that the loss function can be expressed in the form $l(\boldsymbol{X}\boldsymbol{\beta})$ with an explicit l^*. Moreover, any constraint that admit a dual norm formulation could be used and eventually combined with the ℓ_1 or ℓ_2 norms as in Eq. SM 1.1. For instance, by using results in Section D.2.3 of [16], one could easily adapt this for the logistic loss function as in [13].

Remark 2: Note that we use $\widetilde{\text{GAP}}(\boldsymbol{\beta}^k)$, which is an upper-bound (see the proof below) of the true $\text{GAP}(\boldsymbol{\beta}^k)$. For the sake of simplicity, we dropped this notation in the main paper.

Proof. We begin by proving the first property of Eq. SM 3.3 for all $\lambda \geq 0$. By using the Fenchel conjugate properties [5], we obtain

$$\Omega^* = \Big\{ \max_{\boldsymbol{\alpha} \in \mathcal{K}} \Omega_{\boldsymbol{\alpha}} \Big\}^* \leq \inf_{\boldsymbol{\alpha} \in \mathcal{K}} \Omega^*_{\boldsymbol{\alpha}} \leq \Omega^*_{\boldsymbol{\alpha}(\boldsymbol{\beta}^k)} \equiv \Omega^*_k. \tag{SM 3.6}$$

Thus,

$$\begin{aligned}
\widetilde{\text{GAP}}(\boldsymbol{\beta}^k) &\equiv f(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}^k) + \Omega^*_k(-\boldsymbol{X}^\top \boldsymbol{\sigma}^k) \\
&\geq f(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}^k) + \Omega^*(-\boldsymbol{X}^\top \boldsymbol{\sigma}^k) \\
&= \text{GAP}(\boldsymbol{\beta}^k) \\
&\geq f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) \\
&\geq 0.
\end{aligned}$$

Now we prove the second property of Eq. SM 3.3. 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

$$\Omega^*_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}\big(- \boldsymbol{X}^\top \boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big) = \Omega^*\big(- \boldsymbol{X}^\top \boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big). \tag{SM 3.7}$$

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

$$\widetilde{\text{GAP}}(\boldsymbol{\beta}^*) = \text{GAP}(\boldsymbol{\beta}^*) = 0. \tag{SM 3.8}$$

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

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

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 $l(\boldsymbol{X}\boldsymbol{\beta}) + \Omega_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}(\boldsymbol{\beta})$. This can easily be checked since $(\boldsymbol{\beta}^*, \boldsymbol{\alpha}(\boldsymbol{\beta}^*))$ is a saddle point [4] of the the min-max problem

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

Now, we use $\boldsymbol{\beta}^*$ as a particular $\boldsymbol{z}$ and consecutively apply Eq. SM 3.4 and the Fenchel-Young inequality (see Borwein and Lewis [5], Proposition 3.3.4) on

the obtained inequality. The equality holds due to the optimality conditions satisfied by f on $\boldsymbol{\beta}^*$. We obtain

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

Therefore, we deduce Eq. SM 3.7 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, Eq. SM 3.8 holds when $\lambda = 0$. We again use the fact that $\boldsymbol{\beta}^*$ minimizes $l(\boldsymbol{X}\boldsymbol{\beta}) + \Omega_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}(\boldsymbol{\beta})$ to get

$$0 \in \boldsymbol{X}^\top\nabla l(\boldsymbol{X}\boldsymbol{\beta}^*) + \partial\Omega_{\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}(\boldsymbol{\beta}^*) \equiv \boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*) + \kappa\partial\|\boldsymbol{\beta}^*\|_1 + \underbrace{\gamma\boldsymbol{A}^\top\boldsymbol{\alpha}(\boldsymbol{\beta}^*)}_{\boldsymbol{s}^*}.$$

Using the well known sub-differential of the ℓ_1 norm (see Bonnans, Gilbert and Lemarechal [4]), we deduce that for all $1 \le j \le P$

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

where $\boldsymbol{s}^*$ (and $\boldsymbol{k}^*$) is defined exactly like $\boldsymbol{s}^k$ (and $\boldsymbol{k}^k$), but using $\boldsymbol{\beta}^*$ instead of $\boldsymbol{\beta}^k$. Hence, $\boldsymbol{k}^*_j = \big(\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^*)\big)_j + \boldsymbol{s}^*_j$, where we just used the definition of $\boldsymbol{k}^k$ together with the this inequality and the fact that $\operatorname{sign}(x)\cdot|x| = x$). It then follows from an easy and straight-forward computation (plug this $\boldsymbol{k}^*$ into the definition of $\boldsymbol{\sigma}(\cdot)$ for $\lambda = 0$) that $\boldsymbol{\sigma}(\boldsymbol{\beta}^*) = \nabla l(\boldsymbol{X}\boldsymbol{\beta}^*)$ as when $\lambda > 0$ and so Eq. SM 3.7 holds for $\lambda = 0$.

Finally, we establish the Fenchel conjugate expression: First, we consider the case $\lambda > 0$ since Ω^*_k is always finite no transformation of $\boldsymbol{\sigma}(\cdot)$ is needed. In fact,

$$\begin{aligned}\Omega^*_k(\boldsymbol{v}) &\equiv \max_{\boldsymbol{z}\in\mathbb{R}^P}\Big\{\langle\boldsymbol{v},\boldsymbol{z}\rangle - \Omega_{\boldsymbol{\alpha}(\boldsymbol{\beta}^k)}(\boldsymbol{z})\Big\}\\ &= \sum_{j=1}^P \max_{z_j\in\mathbb{R}}\Big\{z_j\Big(v_j - \gamma\big(\boldsymbol{A}^\top\boldsymbol{\alpha}(\boldsymbol{\beta}^k)\big)_j\Big) - \frac{\lambda}{2}z_j^2 - \kappa|z_j|\Big\}\\ &= \frac{1}{2\lambda}\sum_{j=1}^P\left(\left[\Big|v_j - \gamma\big(\boldsymbol{A}^\top\boldsymbol{\alpha}(\boldsymbol{\beta}^k)\big)_j\Big| - \kappa\right]_+\right)^2, \qquad \text{(SM 3.9)}\end{aligned}$$

where $[\,\cdot\,]_+ = \max(0,\cdot)$ and $\boldsymbol{v} = -\boldsymbol{X}^\top\boldsymbol{\sigma}(\boldsymbol{\beta}^k)$.

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

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

Thus, we need to change $\boldsymbol{\sigma}(\boldsymbol{\beta}^k)$ slightly, such that a new dual variable, denoted $\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\beta}^k)$, satisfies $\Omega^*_k(-\boldsymbol{X}^\top\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\beta}^k)) < \infty$, while maintaining the other key

properties from Eq. SM 3.7. Namely, we must obtain, for all $1 \leq j \leq P$, that

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

A straight-forward way to achieve the aforementioned constraint would be to solve the linear system

$$\boldsymbol{X}^\top \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\beta}^k) + \boldsymbol{s}^k = \kappa \mathbf{1} = \boldsymbol{k}^k \tag{SM 3.10}$$

as a function of the scaled dual variable $\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\beta}^k)$. But that would also penalize the components of $\boldsymbol{\sigma}(\boldsymbol{\beta}^k)$ already fulfilling the constraint. In order to avoid over-scaling, we introduce a vector $\boldsymbol{k}^k$ to replace $\kappa \mathbf{1}$ on the right hand side of Eq. SM 3.10. The vector $\boldsymbol{k}^k$ is created such that, for all $j = 1, \ldots, P$,

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

By construction, it has the two following properties:

(i) if $|(\boldsymbol{X}^\top \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\beta}^k))_j + s_j| \leq \kappa$, then $(\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\beta}^k))_j = (\boldsymbol{\sigma}(\boldsymbol{\beta}^k))_j$ and thus remains unchanged,

(ii) otherwise, each k_j^k is bounded at κ and maintains the sign of $(\boldsymbol{X}^\top \boldsymbol{\sigma}(\boldsymbol{\beta}^k))_j + s_j$, which allows us to fairly constrain the components of $\boldsymbol{\sigma}(\boldsymbol{\beta}^k)$ that yields values smaller than $-\kappa$ or larger than κ.

A simple rearrangement of Eq. SM 3.10, assuming that $\boldsymbol{X}\boldsymbol{X}^\top$ is invertible (a reasonable assumption with high-dimensional neuroimaging data and $N \ll P$), gives the new dual variable as

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

□

SM 3.1.3. Proof of the duality gap for the smoothed problem (Theorem 1)

Theorem (Duality gap for the smooth problem) *The following estimation of the duality gap satisfies Eq. 14, for any iterate $\boldsymbol{\beta}^k$:*

$$\textsc{Gap}_\mu(\boldsymbol{\beta}^k) \equiv f_\mu(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}(\boldsymbol{\beta}^k)) + \Omega^*_{\mu,k}(-\boldsymbol{X}^\top \boldsymbol{\sigma}(\boldsymbol{\beta}^k)), \tag{SM 3.11}$$

with dual variable

$$\boldsymbol{\sigma}(\boldsymbol{\beta}^k) \equiv \nabla l(\boldsymbol{X}\boldsymbol{\beta}^k) = \boldsymbol{X}\boldsymbol{\beta}^k - \boldsymbol{y}, \tag{SM 3.12}$$

and the Fenchel conjugates

$$\begin{aligned} l^*(\boldsymbol{z}) =& \frac{1}{2}\|\boldsymbol{z}\|_2^2 + \langle \boldsymbol{z}, \boldsymbol{y} \rangle \\ \Omega^*_{\mu,k}(\boldsymbol{z}) \equiv& \frac{1}{2\lambda} \sum_{j=1}^{P} \left(\left[\left| z_j - \gamma \left(\boldsymbol{A}^\top \boldsymbol{\alpha}^*_\mu(\boldsymbol{\beta}^k) \right)_j \right| - \kappa \right]_+^2 \right) \\ &+ \frac{\gamma\mu}{2} \left\| \boldsymbol{\alpha}^*_\mu(\boldsymbol{\beta}^k) \right\|_2^2, \end{aligned} \tag{SM 3.13}$$

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

Proof. The same approach than with the non-smoothed problem can be used for the smoothed problem given in Eq. 13. The only difference concerns the estimation of maximal values in SM 3.1.2. In fact, basic calculations using Eq. 8 instead of Eq. 7 show that the gap expression is now approximated by

$$\widetilde{\mathrm{GAP}}_\mu(\boldsymbol{\beta}^k) \equiv f_\mu(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}(\boldsymbol{\beta}^k)) + \Omega^*_{\mu,k}(-\boldsymbol{X}^\top \boldsymbol{\sigma}(\boldsymbol{\beta}^k)), \tag{SM 3.14}$$

where the Fenchel conjugate of Theorem 1 should be replaced by

$$\Omega^*_{\mu,k}(\boldsymbol{v}) = \begin{cases} \frac{1}{2\lambda}\sum_{j=1}^{P}\left(\left[\left|v_j - \gamma\left(\boldsymbol{A}^\top \boldsymbol{\alpha}^*_\mu(\boldsymbol{\beta}^k)\right)_j\right| - \kappa\right]_+^2\right) + \frac{\gamma\mu}{2}\left\|\boldsymbol{\alpha}^*_\mu(\boldsymbol{\beta}^k)\right\|_2^2, & \text{if } \lambda > 0, \\ \frac{\gamma\mu}{2}\left\|\boldsymbol{\alpha}^*_\mu(\boldsymbol{\beta}^k)\right\|_2^2 & \text{if } \lambda = 0. \end{cases}$$

where $[\,\cdot\,]_+ = \max(0, \cdot\,)$ and $\boldsymbol{\alpha}^*_\mu$ maximizes Eq. 8. □

SM 3.2. Proof of the optimal smoothing parameter, μ (Theorem 2)

This section provides a proof for the expression of the optimal smoothing parameter μ given in Theorem 2.

Theorem (Optimal smoothing parameter, μ) *For any given $\varepsilon > 0$, selecting the smoothing parameter as*

$$\mu_{opt}(\varepsilon) = \frac{-\gamma M\|\boldsymbol{A}\|_2^2 + \sqrt{(\gamma M\|\boldsymbol{A}\|_2^2)^2 + ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon}}{ML(\nabla(g))}, \tag{SM 3.15}$$

minimizes the worst case bound on the number of iterations required to achieve the precision $f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) < \varepsilon$. Note that $M = P/2$ (Eq. 12) and the Lipschitz constant of the gradient of g as defined in Eq. 13 is $L(\nabla(g)) = \lambda_{\max}(\boldsymbol{X}^\top\boldsymbol{X}) + \lambda$, where $\lambda_{\max}(\boldsymbol{X}^\top\boldsymbol{X})$ is the largest eigenvalue of $\boldsymbol{X}^\top\boldsymbol{X}$.

Proof. The smoothed function, f_μ, as mentioned in Eq. 12 and Eq. 13, provides upper and lower bounds on the original function, f, such that

$$f_\mu \le f \le f_\mu + \gamma\mu M, \tag{SM 3.16}$$

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

$$\mu \in \left(0, \frac{\varepsilon}{\gamma M}\right). \tag{SM 3.17}$$

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 Eq. SM 3.16 with the following two inequalities,

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

we obtain

$$f_\mu(\boldsymbol{\beta}^*_\mu) \le f(\boldsymbol{\beta}^*) \le f_\mu(\boldsymbol{\beta}^*_\mu) + \underbrace{\mu\gamma M}_{<\varepsilon}, \tag{SM 3.18}$$

and thus we obtain Eq. SM 3.17.

Note: Chen et al. [9] used a value for the smoothing parameter equal to $\varepsilon/2\gamma M$. This lies within the valid interval, specified in Eq. SM 3.17. However, as is illustrated in the main paper, it is possible to improve on this value in order to achieve convergence in fewer numbers of iterations.

By Eq. SM 2.3 (using $\boldsymbol{\beta}^*$ instead of $\boldsymbol{\beta}^*_\mu$) and Eq. SM 3.16 we write

$$\varepsilon = f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) = \underbrace{f(\boldsymbol{\beta}^k) - f_\mu(\boldsymbol{\beta}^k)}_{\substack{\text{(SM 3.18) at } \boldsymbol{\beta}^k, \\ \leq \gamma\mu M}} + \underbrace{f_\mu(\boldsymbol{\beta}^k) - f_\mu(\boldsymbol{\beta}^*)}_{\substack{\text{(SM 2.3) at } \boldsymbol{\beta}^*, \\ \leq \frac{2}{t_\mu(k+1)^2}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}} + \underbrace{f_\mu(\boldsymbol{\beta}^*) - f(\boldsymbol{\beta}^*)}_{\substack{\text{(SM 3.18) at } \boldsymbol{\beta}^*, \\ \leq 0}}$$

$$\leq \mu\gamma M + \frac{2\left(L(\nabla(g)) + \gamma\frac{\|\boldsymbol{A}\|_2^2}{\mu}\right)}{(k+1)^2}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2. \quad \text{(SM 3.19)}$$

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

$$\mu_{opt}(\varepsilon) = \frac{-\gamma M\|\boldsymbol{A}\|_2^2 + \sqrt{(\gamma M\|\boldsymbol{A}\|_2^2)^2 + ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon}}{ML(\nabla(g))}. \quad \text{(SM 3.20)}$$

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

SM 3.3. Proof of the convergence rate of CONESTA (Theorem 3)

This section provides the proof of the convergence rate of CONESTA given in Theorem 3.

Theorem (Convergence of CONESTA) *Let* $(\mu^i)_{i=0}^\infty$ *and* $(\varepsilon^i)_{i=0}^\infty$ *be defined recursively by CONESTA (Algorithm 2). Then, we have that*

(i) $\lim_{i\to\infty} \varepsilon^i = 0$, *and*

(ii) $f(\boldsymbol{\beta}^i) \xrightarrow{i\to\infty} f(\boldsymbol{\beta}^*)$.

(iii) *Convergence rate of CONESTA with fixed smoothing (without continuation): For any given desired precision $\varepsilon > 0$, using a fixed smoothing (line 6 of Algorithm 2) with an optimal value of μ, equal to $\mu_{opt}(\varepsilon)$, if the number of iterations k is larger than*

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

then the obtained $\boldsymbol{\beta}^k$ satisfies $f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) < \varepsilon$.

(iv) *Convergence rate of CONESTA (with continuation), assuming uniqueness of the minimum (β^*): For any given desired precision $\varepsilon > 0$, if the total sum of all the inner FISTA iterations is larger than*

$$C/\varepsilon,$$

where $C > 0$ is a constant, then the obtained solution (obtained from (iii)), i.e. $\boldsymbol{\beta}^i$, satisfies $f(\boldsymbol{\beta}^i) - f(\boldsymbol{\beta}^*) < \varepsilon$.

SM 3.3.1. Proof of statement (i)

Proof. First, we recall from Algorithm 1 and Eq. SM 3.3 that, for any positive integer i, if $\boldsymbol{\beta}_\mu^{i+1} = \text{FISTA}(\boldsymbol{\beta}_\mu^i, \mu^i, \varepsilon^i)$ then

$$f_{\mu^i}(\boldsymbol{\beta}_\mu^{i+1}) - f_{\mu^i}(\boldsymbol{\beta}_\mu^*) \leq \widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}_\mu^{i+1}) \leq \varepsilon_\mu^i. \tag{SM 3.21}$$

We know from Eq. SM 2.3 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 Eq. SM 3.3, the stopping rule in Algorithm 1 on Line 6 is now easy to check by using $\widetilde{\text{GAP}}_{\mu^i}$ through the test if $\widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}^k) \leq \varepsilon_\mu^i$. Thus, Eq. SM 3.21 will hold at each iteration.

Next, we use Eq. SM 3.21 to establish the first claim. In fact, we have

$$\begin{aligned}
\varepsilon^{i+1} &= \tau \cdot \left(\mu^i\gamma M + \widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}_\mu^{i+1})\right) \\
&\leq \tau \cdot \left(\mu^i\gamma M + \varepsilon_\mu^i\right) \\
&= \tau \cdot \left(\mu^i\gamma M + \varepsilon^i - \mu^i\gamma M\right) \\
&= \tau \cdot \varepsilon^i \\
&\leq \tau^i \cdot \varepsilon^0 \xrightarrow{i\to\infty} 0.
\end{aligned} \tag{SM 3.22}$$

□

SM 3.3.2. Proof of statement (ii)

Proof. Next, we claim that

$$f(\boldsymbol{\beta}_{\mu}^{i}) - f(\boldsymbol{\beta}^{*}) \leq \varepsilon^{i}, \quad \forall i \in \mathbb{N}, \tag{SM 3.23}$$

which involve the second statement (ii). Indeed, we know from Eq. SM 3.18 that

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

It follows that

$$\begin{aligned}
f(\boldsymbol{\beta}_{\mu}^{i}) - f(\boldsymbol{\beta}^{*}) &= f(\boldsymbol{\beta}_{\mu}^{i}) - f_{\mu^i}(\boldsymbol{\beta}_{\mu}^{i}) \\
&\quad + f_{\mu^i}(\boldsymbol{\beta}_{\mu}^{i}) - f_{\mu^i}(\boldsymbol{\beta}_{\mu^i}^{*}) \\
&\quad + f_{\mu^i}(\boldsymbol{\beta}_{\mu^i}^{*}) - f(\boldsymbol{\beta}^{*}), \\
&\leq \mu^i \gamma M + f_{\mu^i}(\boldsymbol{\beta}_{\mu}^{i}) - f_{\mu^i}(\boldsymbol{\beta}_{\mu^i}^{*}), \\
&\leq \mu^i \gamma M + \widetilde{\mathrm{GAP}}_{\mu^i}(\boldsymbol{\beta}_{\mu}^{i}) \\
&\leq \mu^i \gamma M + \varepsilon_{\mu}^{i} \\
&= \mu^i \gamma M + \varepsilon^i - \mu^i \gamma M \\
&= \varepsilon^i.
\end{aligned} \tag{SM 3.24}$$

□

SM 3.3.3. Proof of statement (iii)

For the sake of simplicity, we use Eq. SM 3.19 and denote by $H(\mu, k)$ the upper-bound of the current error:

$$\varepsilon = f(\boldsymbol{\beta}^{k}) - f(\boldsymbol{\beta}^{*}) \leq \mu\gamma M + \frac{2\left(L(\nabla(g)) + \gamma\frac{\|\boldsymbol{A}\|_2^2}{\mu}\right)}{(k+1)^2}\|\boldsymbol{\beta}^{0} - \boldsymbol{\beta}^{*}\|_2^2 = H(\mu, k). \tag{SM 3.25}$$

We seek for an expression of k (number of iterations) as a function of ε (the desired precision), such that $H(\mu = \mu_{opt}(\varepsilon), k) < \varepsilon$. The proof is decomposed in three steps, including two preliminary lemmas:

1. Lemma 1, provides an expression of $H(\mu, k)$ as a function of only k.
2. Lemma 2, provides an expression of k as a function of ε such that $H(\mu, k) \leq \varepsilon$ holds.
3. Then, the proof uses those two lemmas to calculate an expression of k as a function of ε in the specific case where $\mu = \mu_{opt}(\varepsilon)$.

Lemma 1. *Any $k \geq 0$ and $\mu > 0$, such that*

$$\begin{aligned} \mu &= \frac{\|\boldsymbol{A}\|_2}{k+1}\sqrt{\frac{2}{M}}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2 \\ \Leftrightarrow k+1 &= \frac{\|\boldsymbol{A}\|_2}{\mu}\sqrt{\frac{2}{M}}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2, \end{aligned} \tag{SM 3.26}$$

will minimize $H(\mu, k)$ with respect to μ such that

$$H\left(\frac{\|\boldsymbol{A}\|_2}{k+1}\sqrt{\frac{2}{M}}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2, k\right) = \frac{c_1}{k+1} + \frac{2L(\nabla(g))\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}{(k+1)^2}, \tag{SM 3.27}$$

where

$$c_1 = \sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}.$$

Proof.

$$\begin{aligned} \frac{\partial H(\mu, k)}{\partial \mu} &= \gamma M - \frac{1}{\mu^2}\frac{2\gamma\|\boldsymbol{A}\|_2^2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}{(k+1)^2} = 0 \\ \Leftrightarrow \mu &= \pm\sqrt{\frac{2\gamma\|\boldsymbol{A}\|_2^2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}{(k+1)^2\gamma M}} \\ \Leftrightarrow \mu &= \frac{\|\boldsymbol{A}\|_2}{k+1}\sqrt{\frac{2}{M}}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2 \end{aligned}$$

Then we use this value of μ in $H(\mu, k)$ (Eq. SM 3.25) to obtain Eq. SM 3.27. □

Lemma 2. *If*

$$k+1 = \frac{c_1 + \sqrt{c_1^2 + 8L(\nabla(g))\varepsilon\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}}{2\varepsilon},$$

where again

$$c_1 = \sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2},$$

then

$$H\left(\frac{\|\boldsymbol{A}\|_2}{k+1}\sqrt{\frac{2}{M}}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2, k\right) \leq \varepsilon.$$

Proof. We seek the smallest value of k such that

$$\frac{c_1}{k+1} + \frac{2L(\nabla(g))\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}{(k+1)^2} \leq \varepsilon.$$

Multiplying by $(k+1)^2$ and solving the obtained quadratic equation for $k+1$ leads to the desired equation (we take the positive root). □

We can now complete the proof of statement (iii).

Proof. For a given positive ε, we can get (Theorem SM 3.2) the optimal value of $\mu = \mu_{opt}(\varepsilon)$. After k iterations with this fixed smoothing, the FISTA algorithm will achieve the following precision level from Eq. SM 3.19:

$$f(\boldsymbol{\beta}_k) - f(\boldsymbol{\beta}^*) \leq H(\mu_{opt}(\varepsilon), k).$$

We look for the minimum value of k for which this inequality holds. In other words, in order to make this upper bound minimal, we just need to use an iteration number for which the $\mu_{opt}(\varepsilon)$ is optimal as given by Eq. SM 3.26 of Lemma 1. That is,

$$k+1 = \frac{\|\boldsymbol{A}\|_2}{\mu_{opt}(\varepsilon)}\sqrt{\frac{2}{M}}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2.$$

Using the expression of $\mu_{opt}(\varepsilon)$ (Eq. SM 3.15), we can check that the obtained value of k satisfies

$$\begin{aligned}
k+1 &= \frac{\|\boldsymbol{A}\|_2}{\mu_{opt}(\varepsilon)}\sqrt{\frac{2}{M}}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2 \\
&= \frac{\sqrt{2}ML(\nabla(g))\|\boldsymbol{A}\|_2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2}{\big(-\gamma M\|\boldsymbol{A}\|_2^2 + \sqrt{(\gamma M\|\boldsymbol{A}\|_2^2)^2 + ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon}\big)\sqrt{M}} \\
&= \frac{\sqrt{2}\sqrt{M}L(\nabla(g))\|\boldsymbol{A}\|_2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2}{\big(-\gamma M\|\boldsymbol{A}\|_2^2 + \sqrt{(\gamma M\|\boldsymbol{A}\|_2^2)^2 + ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon}\big)} \\
&\quad \times \frac{\big(\gamma M\|\boldsymbol{A}\|_2^2 + \sqrt{(\gamma M\|\boldsymbol{A}\|_2^2)^2 + ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon}\big)}{\big(\gamma M\|\boldsymbol{A}\|_2^2 + \sqrt{(\gamma M\|\boldsymbol{A}\|_2^2)^2 + ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon}\big)} \\
&= \frac{\sqrt{2}\sqrt{M}L(\nabla(g))\|\boldsymbol{A}\|_2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2}{ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon} \\
&\quad \times \big(\gamma M\|\boldsymbol{A}\|_2^2 + \sqrt{(\gamma M\|\boldsymbol{A}\|_2^2)^2 + ML(\nabla(g))\|\boldsymbol{A}\|_2^2\varepsilon}\big) \\
&= \frac{c_1 + \sqrt{c_1^2 + 8L(\nabla(g))\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2\varepsilon}}{2\varepsilon}.
\end{aligned}$$

Next, we use $\mu_{opt}(\varepsilon)$ with this iteration number in Lemma 2, and obtain

$$f(\boldsymbol{\beta}_k) - f(\boldsymbol{\beta}^*) < H\Big(\mu = \mu_{opt}(\varepsilon), k+1 = \frac{\|\boldsymbol{A}\|_2}{\mu_{opt}(\varepsilon)}\sqrt{\frac{2}{M}}\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2\Big) < \varepsilon.$$

In conclusion, we have proved that for a given $\varepsilon > 0$, if we smooth our objective function using $\mu = \mu_{opt}(\varepsilon)$, then after

$$\frac{\sqrt{2}\|\boldsymbol{A}\|_2\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2}{\mu_{opt}(\varepsilon)\sqrt{M}} = \frac{c_1 + \sqrt{c_1^2 + 8L(\nabla(g))\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2\varepsilon}}{2\varepsilon}$$

iterations, we have $f(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) < \varepsilon$. Finally, we note that this remains true for larger iteration numbers, since $H(\mu_{opt}(\varepsilon), k)$ is non-increasing function of k

for a fixed ε. Especially, this holds for

$$k := \frac{c_1}{\varepsilon} + \frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}}{\sqrt{\varepsilon}} \geq \frac{c_1 + \sqrt{c_1^2 + 8L(\nabla(g))\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2\varepsilon}}{2\varepsilon}$$

due to the following inequality: $\sqrt{x} + \sqrt{x+y} \leq 2\sqrt{x} + \sqrt{y}$, where $x = \frac{c_1^2}{4\varepsilon^2}$ and $y = \frac{8L(\nabla(g))\varepsilon\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}{4\varepsilon^2}$. This completes the proof of Statement (iii). □

SM 3.3.4. Proof of statement (iv)

The final demonstration will complete the proof of Theorem 3. Here we consider the convergence rate with respect to the total number of iterations. In order to estimate an upper bound for the smallest needed number of iterations, we just need to find an integer k at which the desired precision level is achieved. This is equivalent to estimating the sum of the number 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 Eq. SM 3.24, we have

$$f(\boldsymbol{\beta}_\mu^i) - f(\boldsymbol{\beta}^*) < \varepsilon^i \leq \tau^i \cdot \varepsilon^0. \tag{SM 3.28}$$

Thus, we conclude that

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

where int is the integer part function.

Now, we sum the iterations, k_i, with respect to i. From (iii) we get that

$$\begin{aligned} k_i &\geq \frac{\sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^i - \boldsymbol{\beta}^*\|_2^2}}{\varepsilon^i} + \frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^i - \boldsymbol{\beta}^*\|_2^2}}{\sqrt{\varepsilon^i}} \\ &\geq \frac{\sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^i - \boldsymbol{\beta}^*\|_2^2}}{\tau^{i-1}\varepsilon^0} + \frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^i - \boldsymbol{\beta}^*\|_2^2}}{\sqrt{\tau^{i-1}\varepsilon^0}}. \end{aligned} \tag{SM 3.29}$$

Thus, the total number of iterations, k, satisfies

$$k \geq \sum_{i=1}^{i_{\max}} \frac{\sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2\|\boldsymbol{\beta}^i - \boldsymbol{\beta}^*\|_2^2}}{\tau^{i-1}\varepsilon^0} + \frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^i - \boldsymbol{\beta}^*\|_2^2}}{\sqrt{\tau^{i-1}\varepsilon^0}}. \tag{SM 3.30}$$

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:

$$c_2 := \sqrt{8\|\boldsymbol{A}\|_2^2 M\gamma^2 C(\boldsymbol{\beta}^0)} \quad \text{and} \quad c_3 := \sqrt{2L(\nabla(g))C(\boldsymbol{\beta}^0)}.$$

Hence we obtain

$$\begin{aligned} k &\geq \sum_{i=1}^{i_{\max}} \frac{c_2}{\tau^{i-1}\varepsilon^0} + \frac{c_3}{\sqrt{\tau^{i-1}\varepsilon^0}} \\ &\geq \frac{c_2}{\varepsilon^0} \frac{1-(1/\tau)^{i_{\max}}}{1-\frac{1}{\tau}} + \frac{c_3}{\sqrt{\varepsilon^0}} \frac{1-(1/\sqrt{\tau})^{i_{\max}}}{1-\frac{1}{\sqrt{\tau}}} \end{aligned}$$

But since $\log(\tau) < 0$ and $\log\left(\frac{\varepsilon}{\varepsilon^0}\right) < 0$, we have:

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

for the global minimum, ε. Hence, we obtain

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

and similarly we can establish that

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

Finally we deduce that

$$k \geq \frac{c_2}{\varepsilon^0(1-1/\tau)} + \frac{c_3}{\sqrt{\varepsilon^0}(1-1/\sqrt{\tau})} + \left(\frac{c_2}{1/\tau-1} + \frac{\sqrt{\varepsilon^0}c_3}{1/\sqrt{\tau}-1}\right)\frac{1}{\varepsilon}, \quad \text{(SM 3.31)}$$

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 $\mathcal{O}(C/\varepsilon)$.

SM 4. Experiments on a structural MRI data set

SM 4.1. MRI data acquisition and processing

The data used in the preparation of this paper were obtained from the database of the Alzheimer's disease (AD) neuroimaging initiative (ADNI) (http://adni.loni.usc.edu/).

The MRI data set included standard T1-weighted images obtained with different 1.5-T scanner types using a three-dimensional MP-RAGE sequence or equivalent protocols with varying resolutions. The images were post-processed to correct for some artifacts [15]. As as result, 509 images [10] were segmented into Gray Matter (GM), White Matter (WM) and Cerebrospinal Fluid (CSF) using the SPM8 unified segmentation routine [2].

A total of 456 images were retained after quality control on GM probability. These images were spatially normalized into a template (dimension: $121\times145\times121$, voxels size: 1.5 mm isotropic) using DARTEL [1] and modulated with the Jacobian determinants of the nonlinear deformation field.

We retain GM voxels with a minimum value of 0.01 and at least 10^{-6} of standard-deviation across the cohort. Then each voxel was centered and scaled at the cohort level. Those masked, warped and modulated GM images (286 214 voxels) were completed with three demographic predictors (age, gender and education level) leading to $P = 286\,217$ input features. The three demographic predictors were excluded from any penalization.

The ultimate goal of this machine learning approach is to predict the clinical evolution outcome of the subjects, and we used as the target variable ($\boldsymbol{y}$), the ADAS (Alzheimer's Disease Assessment Scale-Cognitive Subscale) score measured 800 days after the acquisition of the brain images. The ADAS score is one of the most frequently used tests to measure cognition in clinical trials and it is provided in the ADNI data set.

As participants, we considered one group of 119 control subjects (CTL) that never converted to AD within the six years of the study. As patients, we considered one group of patients with mild cognitive impairment (MCI) that converted to AD within 800 days. We pooled those two groups leading to a data set with $n = 199$ subjects.

SM 4.2. Effect of the τ parameter on the convergence speed of CONESTA

Fig. 1 illustrates the convergence speed, on the MRI ADNI data, of the CONESTA solver for different values of the factor τ (Sec. III-C) that decreases the sequence of precisions. Values of $\tau = 0.5$ and $\tau = 0.2$ led to a similar increased convergence speed compared to the lager value of $\tau = 0.8$.

SM 4.2.1. Required precision and its gap estimate

In Sec. IV-B3, p. 9 we evaluate the required prediction errors of real-life experiments. The Fig. 4) provides the similarity (correlation) between the coefficient maps $\boldsymbol{\beta}^k$ and the true solution $\boldsymbol{\beta}^*$ as a function of the true precision (red line) and precision estimated with the duality gap (blue line). We found that precision of 10^{-3} was required to obtain a solution similar to the true solution.

However, similar conclusions can be drawn if we measure the (normalized) sum of absolute error (SAE) of prediction between $\boldsymbol{\beta}^k$ and the true solution

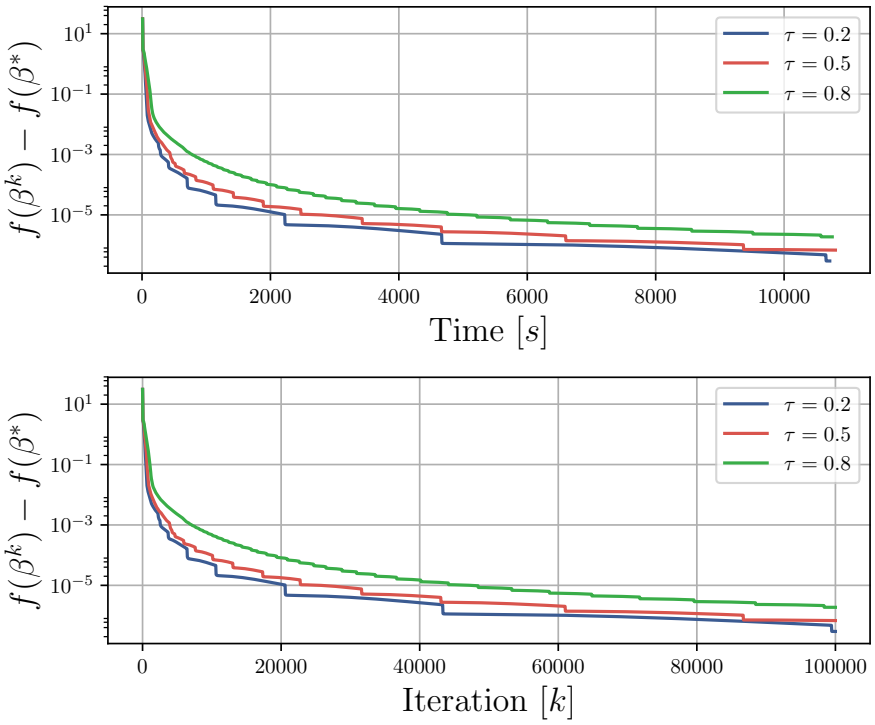

FIG 1. *The error as a function of the computational time (top plot) and the number of iterations (bottom plot) for different values of τ parameter with the CONESTA solver.*

$\boldsymbol{\beta}^*$, *i.e.* $\|\boldsymbol{X}\boldsymbol{\beta}^k - \boldsymbol{X}\boldsymbol{\beta}^*\|_1 / \|\boldsymbol{X}\boldsymbol{\beta}^*\|_1$ (see Fig. 2). Stopping at $\varepsilon = 10^{-3}$, estimated using either the duality gap or with the true precision, leads to less that 1% of sum of absolute error in all cases. An early stopping at 10^{-2} would lead to almost 3% of sum of absolute error, when considering the true precision.

SM 5. ParsimonY: Structured and sparse machine learning in Python

This section provides a simple example of the ParsimonY library applied on a large neuroimaging data set: $N = 199, P = 286\ 217$ made up of three unpenalized covariates (Age, Gender, Education) and 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

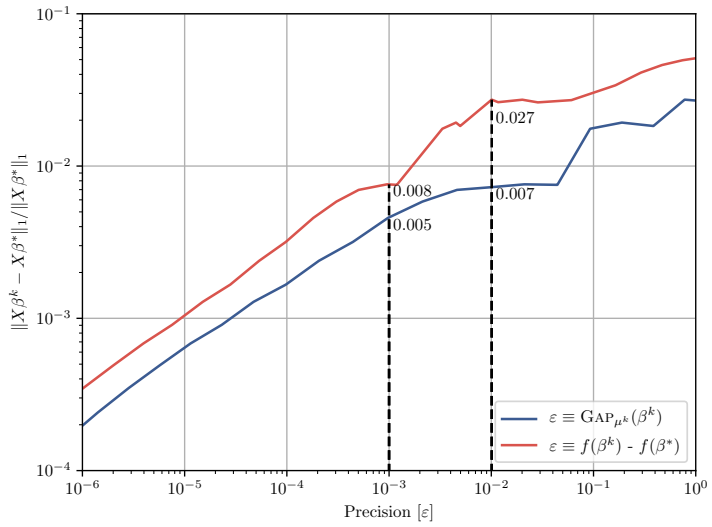

FIG 2. *Normalized sum of absolute error (SAE)* $\|\boldsymbol{X\beta}^k - \boldsymbol{X\beta}^*\|_1/\|\boldsymbol{X\beta}^*\|_1$ *as a function of the true precision (red line) and precision estimated with the duality gap (blue line).*

ParsimonY is stives to be 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, nibabel
import parsimony.functions.nesterov.tv as tv
import parsimony.estimators as estimators

# 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 left un-penalized covariates
#    => penalty_start = 3
#    Omit if no covariates; set to 1 with one covariate (such as the
#    intercept).
#
# - y: target vector of shape (199, 1)

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

estimator = estimators.LinearRegressionL1L2TV(
    l1=0.01/3, l2=0.01/3, ltv=0.01/3, A=Atv,
    penalty_start=3)

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

# Save weight map to a 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("weights.nii")
```

SM 6. Technical details of solvers used in the comparison studies

SM 6.1. Smoothing proximal gradient or FISTA with fixed μ

A detailed description is provided in Sec. II of the main document. Chen *et al.* [9] demonstrated that the convergence rate obtained with a single value of μ, even optimized, is $\mathcal{O}(1/\varepsilon) + \mathcal{O}(1/\sqrt{\varepsilon})$.

SM 6.2. The excessive gap method

It can prove cumbersome to apply the excessive gap method [17] 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.

General framework Let us first recall Eq. 1, which describes the optimization problem under consideration, namely

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

Following Nesterov [17, Section 1], since g in Eq. 1 is a strongly convex function, we can apply the version of the excessive gap method with an $\mathcal{O}\big(1/k^2\big)$ rate of convergence toward the minimum of f, where k is the number of iterations [17, Theorem 7.6].

For the sake of completeness and notation, we recall the definition of strong convexity [4].

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 Sec. II, 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

$$f_\nu(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) + (\kappa h + \gamma s)_\nu(\boldsymbol{\beta}). \tag{SM 6.1}$$

Under the hypothesis that the necessary condition for applying Nesterov's smoothing applies (see Eq. 7), Eq. 1 is expressed as a min-max problem.

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^P} f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}\in\mathbb{R}^P} \left\{ g(\boldsymbol{\beta}) + \max_{\boldsymbol{\alpha}\in\mathcal{K}'} \langle \boldsymbol{\alpha}, \boldsymbol{A}'\boldsymbol{\beta}\rangle \right\}, \tag{SM 6.2}$$

where we, for the sake of readability, 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 $\max_{\boldsymbol{\alpha}\in\mathcal{K}'} \frac{1}{2}\|\boldsymbol{\alpha}\|_2^2$.

The saddle point theorem [4] allows us to write

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^P} f(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}\in\mathbb{R}^P} \max_{\boldsymbol{\alpha}\in\mathcal{K}'} \left\{ g(\boldsymbol{\beta}) + \langle \boldsymbol{\alpha}, \boldsymbol{A}'\boldsymbol{\beta}\rangle \right\} = \max_{\boldsymbol{\alpha}\in\mathcal{K}'} \min_{\boldsymbol{\beta}\in\mathbb{R}^P} \left\{ g(\boldsymbol{\beta}) + \langle \boldsymbol{\alpha}, \boldsymbol{A}'\boldsymbol{\beta}\rangle \right\}, \tag{SM 6.3}$$

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 [17, 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*

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

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 SM 1. This algorithm achieves a convergence rate of $\mathcal{O}(1/k^2)$ when the differentiable part of the optimization problem is strongly convex.

Remark: it is necessary to smooth $\kappa h + \gamma s$ instead of just smoothing γs since a major step in the excessive gap method [17, 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 [20] has no equivalence in the excessive gap framework with an inexact $\widehat{\boldsymbol{\beta}}(\cdot)$.

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 Eq. SM 1.1. 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):

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

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

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

The linear operator for the excessive gap method is

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

where $\boldsymbol{I}_P$ is the $P \times P$ identity matrix .

Algorithm SM 1 The excessive gap method

Require: $\widehat{\boldsymbol{\beta}}(\cdot)$, $\boldsymbol{\alpha}_\nu^*(\cdot)$, $V(\cdot)$, $L(\nabla D_{EG}) > 0$, $\varepsilon > 0$, $M' \geq 0$
Ensure: $\boldsymbol{\beta}^k$ 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

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

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

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

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

respectively, where $\lambda_{\min}(\boldsymbol{X}^\top\boldsymbol{X})$ is the smallest eigenvalue of $\boldsymbol{X}^\top\boldsymbol{X}$. Finally, using Nesterov's Theorem [18], we can establish the expression for the optimal value of the dual variable

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

and the gradient mapping

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

SM 6.3. The Alternating Direction Method of Multipliers (ADMM)

Consider a problem of 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) [6], also known as *Douglas-Rachford splitting*, can be used to minimize this problem. The general ADMM algorithm is presented in Algorithm SM 2.

We recall the function in Eq. SM 1.1 and restrict the structured penalty to a total variation penalty in the 1D setting. We aim to minimize the function

$$\begin{aligned} f(\boldsymbol{\beta}) &= \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \kappa\|\boldsymbol{\beta}\|_1 + \gamma\sum_i \|\boldsymbol{A}_{\phi(i)}\boldsymbol{\beta}\|_2 \\ &= \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \kappa\sum_{j=1}^{P}|\beta_j| + \gamma\sum_{i=1}^{P-1}|\beta_{i+1} - \beta_i| \end{aligned} \qquad \text{(SM 6.4)}$$

over $\beta \in \mathbb{R}^P$. We have adapted the ADMM-based solver described by Wahlberg et al. [22] by making the ridge regression loss function explicit; we have also added an ℓ_1 penalty to their derivation.

We rewrite the minimization of the function in Eq. SM 6.4 as the equivalent problem

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

The ADMM equivalent form of this second problem is

$$\begin{aligned}
&\min_{\boldsymbol{\beta},\boldsymbol{r}} \ \widetilde{f}(\boldsymbol{\beta},\boldsymbol{r}) = \underbrace{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \kappa\sum_{j=1}^{P}|r_j| + \gamma\sum_{i=P+1}^{2P}|r_i|}_{g} + \underbrace{\iota_{\mathcal{C}}(\boldsymbol{z},\boldsymbol{s})}_{h},\\
&\text{s.t.} \ \ \beta_j = z_j, \quad j = 1,\dots,P\\
&\qquad r_i = s_i, \quad i = 1,\dots,2P,
\end{aligned} \tag{SM 6.5}$$

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

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

Eq. SM 6.5 is the problem that we will focus our attention on in this section. The augmented Lagrangian of the problem in Eq. SM 6.5 is

$$\mathcal{L}(\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{r},\boldsymbol{s},\rho) = \widetilde{f}(\boldsymbol{\beta},\boldsymbol{r}) + \frac{\rho}{2}\big(\|\boldsymbol{\beta}-\boldsymbol{z}+\boldsymbol{u}\|_2^2 + \|\boldsymbol{r}-\boldsymbol{s}+\boldsymbol{t}\|_2^2\big), \tag{SM 6.6}$$

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

$$\boldsymbol{\beta}^+ = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^P}\left\{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{\beta}-\boldsymbol{z}+\boldsymbol{u}\|_2^2\right\}, \tag{SM 6.7}$$

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

Require: $g : \mathbb{R}^P \to \mathbb{R}\cup\{+\infty\}$, $h : \mathbb{R}^P \to \mathbb{R}\cup\{+\infty\}$

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**

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 Eq. SM 6.7 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

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

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

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

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 SM 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

$$\min \ \|\boldsymbol{z}-\boldsymbol{w}\|_2^2+\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{v}\|_2^2, \tag{SM 6.8}$$

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

We solve this problem analytically as follows: The gradient of Eq. SM 6.8 is at the optimum

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

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 SM 2, thus corresponds to the projection

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

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

Algorithm SM 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: $\quad\boldsymbol{r}_{\ell_1}^{k+1}=\mathrm{prox}_{\frac{\lambda}{\rho}\|\cdot\|_1}\left(\boldsymbol{s}_{\ell_1}^k-\boldsymbol{t}_{\ell_1}^k\right)$
4: $\quad\boldsymbol{r}_{TV}^{k+1}=\mathrm{prox}_{\frac{\gamma}{\rho}\|\cdot\|_1}\left(\boldsymbol{s}_{TV}^k-\boldsymbol{t}_{TV}^k\right)$
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 $\boldsymbol{X}^\top\boldsymbol{X}$ once, and then 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 $\boldsymbol{A}^\top\boldsymbol{A}$, but note that this is not necessarily possible with other penalties.

The penalty parameter ρ: As far as the authors are aware, the penalty parameter, ρ, in Eq. SM 6.6 is unknown, and finding good values for it is still an open problem.

We use the heuristics presented by Boyd et al. [6, 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.

The approach to updating ρ^k discussed by Boyd et al. [6] 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. [6] 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.

SM 6.4. The Inexact proximal gradient method

In this section, we adapt the inexact proximal approach for solving Eq. 1. We suppose that the non-smooth part, $\gamma s(\boldsymbol{\beta}) + \kappa h(\boldsymbol{\beta})$, satisfies Nesterov's assumption as stated in Eq. 7; 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,

$$\text{prox}_{th}(\boldsymbol{v}^k) = \arg\min_{\mathbf{u}\in\mathbb{R}^P} \left\{ \frac{1}{2}\|\mathbf{u} - \boldsymbol{v}^k\|_2^2 + th(\mathbf{u}) \right\}. \tag{SM 6.9}$$

Following Schmidt, Le Roux and Bach [20], we are looking for a stopping criterion in the algorithm to come, and a precision $\varepsilon^k > 0$ such that

$$\frac{1}{2t}\left\|\widehat{\text{prox}}_{th}(\boldsymbol{v}^k) - \boldsymbol{v}^k\right\|_2^2 + h\left(\widehat{\text{prox}}_{th}(\boldsymbol{v}^k)\right) \leq \varepsilon^k + \min_{\mathbf{u}\in\mathbb{R}^P} \left\{ \frac{1}{2}\|\mathbf{u} - \boldsymbol{v}^k\|_2^2 + th(\mathbf{u}) \right\} \tag{SM 6.10}$$

where $\widehat{\text{prox}}_{th}(\boldsymbol{v}^k)$ is the approximation of $\text{prox}_{th}(\boldsymbol{v}^k)$ obtained from a numerical approximation of Eq. SM 6.9, using any minimization algorithm. From Schmidt, Le Roux and Bach [20], 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 Eq. SM 6.10.

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

$$\boldsymbol{\alpha}_k^* \equiv \arg\min_{\boldsymbol{\alpha}\in\mathcal{K}'}\frac{1}{2}\|\boldsymbol{v}^k-t\boldsymbol{A}'^{\top}\boldsymbol{\alpha}\|_2^2 \tag{SM 6.11}$$

using FISTA, and then compute

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

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 Eq. SM 6.11, 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 [20]. We use the min-max duality gap (see Bonnans, Gilbert and Lemarechal [4]) as follows. At step i of the inner FISTA loop when minimizing Eq. SM 6.11 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

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

and finally the stopping criterion is that

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

References

[1] Ashburner, J. (2007). A fast diffeomorphic image registration algorithm. *NeuroImage* **38** 95–113.

[2] Ashburner, J. and Friston, K. J. (2005). Unified segmentation. *NeuroImage* **26** 839–851.

[3] Beck, A. and Teboulle, M. (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences* **2** 183–202.

[4] 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.

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

[6] 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.

[7] Chambolle, A. and Pock, T. (2011). A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *J. Math. Imaging Vis.* **40** 120–145.

[8] Chen, X. and Liu, H. (2011). An Efficient Optimization Algorithm for Structured Sparse CCA, with Applications to eQTL Mapping. *Statistics in Biosciences* **4** 3–26.

[9] 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.

[10] Cuingnet, R., Gerardin, E., Tessieras, J., Auzias, G., Lehericy, S., Habert, M.-O., Chupin, M., Benali, H., Colliot, O. and Alzheimer's Disease Neuroimaging Initiative (2011). Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the ADNI database. *NeuroImage* **56** 766–781.

[11] Dohmatob, E., Gramfort, A., Thirion, B. and Varoquaux, G. (2014). Benchmarking solvers for TV-l1 least-squares and logistic regression in brain imaging. *Pattern Recognition in Neuroimaging (PRNI).*

[12] Dohmatob, E., Eickenberg, M., Thirion, B. and Varoquaux, G. (2015). Speeding-up model-selection in GraphNet via early-stopping and univariate feature-screening. In *PRNI.*

[13] Dubois, M., Hadj-Selem, F., Lfstedt, T., Perrot, M., Fischer, C., Frouin, V. and Duchesnay, E. (2014). Predictive support recovery with TV-Elastic Net penalty and logistic regression: An application to structural MRI. In *2014 International Workshop on Pattern Recognition in Neuroimaging* 1–4.

[14] Gramfort, A., Thirion, B. and Varoquaux, G. (2013). Identifying

predictive regions from fMRI with TV-L1 prior. In *Pattern Recognition in Neuroimaging (PRNI)*.

[15] Jack, C. R., Bernstein, M. A., Fox, N. C., Thompson, P., Alexander, G., Harvey, D., Borowski, B., Britson, P. J., L Whitwell, J., Ward, C., Dale, A. M., Felmlee, J. P., Gunter, J. L., Hill, D. L. G., Killiany, R., Schuff, N., Fox-Bosetti, S., Lin, C., Studholme, C., DeCarli, C. S., Krueger, G., Ward, H. A., Metzger, G. J., Scott, K. T., Mallozzi, R., Blezek, D., Levy, J., Debbins, J. P., Fleisher, A. S., Albert, M., Green, R., Bartzokis, G., Glover, G., Mugler, J. and Weiner, M. W. (2008). The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI methods. *Journal of magnetic resonance imaging: JMRI* **27** 685–691.

[16] Mairal, J. (2010). Sparse coding for machine learning, image processing and computer vision PhD thesis, École normale supérieure de Cachan.

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

[18] Nesterov, Y. (2005b). Smooth minimization of non-smooth functions. *Mathematical Programming* **103** 127–152.

[19] Orabona, F., Argyriou, A. and Srebro, N. (2012). PRISMA: PRoximal Iterative SMoothing Algorithm.

[20] 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*.

[21] Varoquaux, G., Eickenberg, M., Dohmatob, E. and Thirion, B. (2015). FAASTA: A fast solver for total-variation regularization of ill-conditioned problems with application to brain imaging. In *Colloque GRETSI*. P. Gonçalves, P. Abry.

[22] 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*.