ACTIVE-SET STRATEGY BASED ON A GENERAL MODIFIED NEWTON-RAPHSON ALGORITHM FOR VARIABLE SELECTION IN HIGHLY ILL-POSED INVERSE PROBLEMS

Mayrim Vega-Hernández*. **, Darío Palmero-Ledón*, Agustín Lage-Castellanos*, José M. Sánchez-Bornot***, Pedro A. Valdés-Sosa*, **, Eduardo Martínez-Montes[1]*
*Cuban Center for Neurosciences, Havana, Cuba.
**The Clinical Hospital of Chengdu Brain Science Institute, MOE Key Lab for Neuroinformation, University of Electronic Science and Technology of China, Chengdu, China.
***School of Computing and Intelligent Systems, Ulster University, UK.

ABSTRACT:
We propose a novel algorithm to perform efficient modified-Newton-Raphson optimization over the active set of selected features (AMNR), and show that it allows to estimate multiple penalized least-squares (MPLS) models. MPLS models are used to find flexible and adaptive least-squares solutions to highly ill-posed linear inverse problems, mainly requiring them to be simultaneously sparse and smooth. This is relevant for applications where there is no ground truth, e.g., estimating electrophysiological sources. In this work, we step on a modified Newton-Raphson algorithm that can be interpreted as a generalization of the Minorization-Maximization algorithm to include combinations of several constraints, and derive the AMNR algorithm following an approach similar to that used for Least Angle Regression. This algorithm allows us to implement many different MPLS models, including novel models such as the Smooth Nonnegative Garrote and the Nonnegative Smooth LASSO. The performance of the algorithm is evaluated using simulated data from a simple ill-posed linear regression and from a realistic Electroencephalographic setup. Different models containing one or two penalty functions, and including sign constraints were evaluated in both cases. The algorithm allowed the recovering of solutions in a fast way and with adequate quality in simulated scenarios for different n/p ratios.

KEYWORDS*:* multiple penalized least-squares, active set, inverse problem, EEG, LARS.

MSC: 90C53

RESUMEN
Proponemos un algoritmo novedoso para realizar una optimización eficiente de Newton-Raphson modificada, sobre el conjunto activo de variables (AMNR). Mostramos que este permite estimar modelos de Mínimos Cuadrados con Múltiple Penalización (MPLS). Los modelos MPLS se utilizan para encontrar soluciones de mínimos-cuadrados flexibles y adaptables, para problemas inversos mal planteados de alta dimensionalidad, restringiéndolas a que sean simultáneamente ralas y suaves. Esto es relevante para resolver problemas inversos reales donde no hay criterio de verdad, como en la estimación de las fuentes electrofisiológicas. En este trabajo, partimos de un algoritmo de Newton-Raphson modificado que generaliza el algoritmo de *Minorization-Maximization* para incluir combinaciones de penalizadores, y derivamos el algoritmo AMNR siguiendo un enfoque similar al utilizado para la regresión LARS. El AMNR nos permite implementar muchos modelos MPLS diferentes, incluidos modelos novedosos como el Garrote suave no-negativo y el LASSO suave no-negativo. El rendimiento del algoritmo se evalúa utilizando datos simulados de una regresión lineal mal planteada y datos reales de electroencefalografía, para un grupo de modelos que contienen una o dos funciones de penalización e incluyen además restricciones de signo. El algoritmo permitió recuperar soluciones de forma más rápida y con suficiente calidad en escenarios simulados para diferentes razones n/p.

PALABRAS CLAVES*:* mínimos-cuadrados con múltiple penalización, conjunto activo, problema inverso, EEG, LARS.

1. INTRODUCTION

We consider the linear regression model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + e$, where the columns of matrix $\mathbf{X}$ ($\mathbf{x}_j \in \mathbb{R}^n$, j=1… p) are predictors, $\mathbf{y} \in \mathbb{R}^n$ the response vector, $\boldsymbol{\beta} \in \mathbb{R}^p$ the vector of coefficients to be estimated and $\varepsilon \in \mathbb{R}^n$ the error term with $\varepsilon \sim N(\vec{0}, \sigma^2\mathbf{I})$. The interesting underdetermined scenario where $p \gg n$ leads to highly ill-posed problems with no unique solution. To find a plausible solution to a particular problem, the only way is to use additional constraints to the solution. The classical approach for this is the Tikhonov regularization [28] which formulates the problem as a penalized regression, where the penalty term $\Psi(\boldsymbol{\beta})$ defines the characteristics of the solution sought:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\lambda}\Psi(\boldsymbol{\beta})\},$$

Here λ is a regularization parameter which determines the relative weight of the penalty function versus the likelihood term, which measures the level of data fitting. In the original Tikhonov approach, the solution is taken as the limit when the regularization parameter tends to zero. However, this usually tends to the Ordinary Least Squares (OLS) solution, which has the minimum variance among all unbiased estimators, but which was quickly recognized as not being the optimal one in real-world problems. An alternative way is to directly apply penalized regression methods, which lead to biased estimators with smaller variance than that of the OLS estimator [8]. This approach has been argued to be useful for both developing predictive models and selecting key indicators from a substantially larger pool of available indicators [9]. Following this penalized least-squares regression approach, many new extensions of regularization techniques have been applied to this type of problems in the last decade [32]. These techniques produce biased but stable linear solutions when using L2-norm penalties, being Ridge regression [12] the classical example, which can be stated in its general form as:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\lambda}\|\boldsymbol{\beta}\|_2^2\},$$

The advent of the least absolute shrinkage and selection operator (LASSO) [26] and the emergence of the more general penalized least squares formulation [4], allowed the recovery of sparse solutions, where only a small number of coefficients are nonzero, while Ridge does not produce sparse estimators. In this context, LASSO can be stated as

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\lambda}\|\boldsymbol{\beta}\|_1\},$$

which mainly differs from Ridge by using the L1-norm of coefficients instead of the L2-norm as the regularization term. As this type of penalty leads to sparse solutions, the estimation process can be seen as a variable selection technique. The penalty function can be based on the norm of linear combinations of the coefficients to impose smoother estimators and may also involve both L1-norm and L2-norm functions. A general formulation can thus be obtained when the penalty functional is a sum of several convex and non-convex constraints or penalty terms. This general model has been named as Multiple Penalized Least Squares (MPLS) [31, 24] and in this broad sense, specific methods that have been proposed and validated in the literature such as the Fused LASSO [27], the Elastic Net (ENET) [34], the Smooth LASSO (SLASSO) [11], and the recently proposed Liu-LASSO [8], can be seen as particular instances.

From the computational point of view, the local quadratic approximation (LQA) [5] and the minorization-maximization (MM) [13, 17] algorithms provide a numerical engine to implement specific penalized least-squares models. These algorithms can be seen as applications of the modified Newton-Raphson (NR) technique using an approximation of the objective function. LQA and MM inherits the virtues of the NR, but they cannot deal with any combination of penalty terms in more general MPLS models. To overcome this difficulty, Valdés-Sosa and colleagues introduced a generalized MM method for the estimation of massive autoregressive models of neuroimaging data [30], and for solving the M/EEG inverse problem [31]. Another drawback is that neither LQA nor MM produce true sparse solutions and depend on an additional parameter to assure numerical stability [19]. Although Kim et al. (2018) [14] found that MM are quite robust to correlated predictors and show stable convergence for a wide range of the regularization parameter, they also commented that in practice these algorithms can be combined with an active set strategy, which reduces the number of variables before the algorithm runs by setting the coefficients of the discarded variables to zero. Some examples of this approach are the least angle regression (LARS) [4] and the Shooting algorithm [7] also known as coordinate-wise descent [6]. These algorithms offer efficient implementations for some specific models (e.g., LASSO, ENET) with the advantage that they make variable selection and estimation simultaneously. However, their application scope is not as extensive as in the case of LQA and MM approaches [33].

The recovery of simultaneously sparse and smooth estimators constitutes a challenge that was addressed by our group using a computational "trick" and the MM algorithm [30]. However, it cannot ensure strict sparseness or produce combined smooth and sign constrained solutions. Indeed, the use of NR techniques to solve penalized least-squares methods with nonnegative constraints has been much less explored (see [19] and [18] for alternative and closely related procedures for imposing nonnegativity in linear regression). Using direct variable selection methods seems to offer a more feasible approach, as shown by Mørup et al., 2008 [20], who introduced a version of the LARS algorithm to implement the LASSO model with nonnegative constraints.

In this work, we step on a modified NR algorithm that can be interpreted as a generalization of the Minorization-Maximization algorithm to include combinations of several constraints, and then propose an algorithm based on the active-set technique (AMNR). The AMNR can be seen as an extension of the LARS algorithm for convex and continuously differentiable cost functions in possible nonnegative (nonpositive) scenarios. The AMNR algebraic details and proof of the optimality conditions are discussed in this paper for the Adaptive LASSO [35], which makes it easy to use the algorithm for estimating a wider class of optimization problems using the combination of penalties based on L1 and L2 norms. In particular, it provides a straightforward application of the AMNR algorithm to estimate the "whole path of coefficients" for the Nonnegative Garrote. Then, we also extend it to include an L2-norm penalty to create the Smooth Nonnegative Garrote (SNNG) and to include sign constraints on the Smooth LASSO model to create the Nonnegative Smooth LASSO (NN-SLASSO) method.

This article is organized as follows: The MNR algorithm for implementing MPLS methods to extract sparse and smooth solutions is formulated in Section 2, while the proposed general AMNR algorithm is shown in Section 3. Sections 4 and 5 will provide the proof of the optimality conditions and the implementation of the AMNR algorithm for the Adaptive LASSO model, respectively. Then, Section 6 is devoted to illustrate the performance of the proposed algorithm using simulated data, both with a general random design matrix and with realistic EEG sources. The Conclusions of the study follows, together with a discussion of limitations and future research.

2. FORMULATION OF THE MODIFIED NEWTON-RAPHSON (MNR) ALGORITHM AND RELATION WITH THE MINORIZATION-MAXIMIZATION (MM) ALGORITHM

The Multiple Penalized Least Squares (MPLS) model [31,24] is stated as the minimization of the following functional:

$$f(\boldsymbol{\beta}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \Psi(\boldsymbol{\beta}) \tag{1}$$

where the penalty term takes the form of a sum of several constraints or penalty functions, i.e., $\Psi(\boldsymbol{\beta}) = \sum_{r=1}^{R} \lambda_r \sum_{i=1}^{N_r} g_r\left(\left|\theta_i^{(r)}\right|\right)$. This is evaluated at the components of the vector $\boldsymbol{\theta}^{(r)} = \mathbf{L}^{(r)}\boldsymbol{\beta}$, with $\mathbf{L}^{(r)} \in \mathbb{R}^{N_r \times p}$ being linear operators that impose a structural relationship among coefficients, (e.g., the matrix of first or second differences). The regularization parameters λ_r, for $r = 1, \ldots, R$, establish the relative importance of each constraint. As can be easily shown, LASSO and Ridge regression minimize instances of equation (1), setting $R = 1$, $\mathbf{L} = \mathbf{I}_p$ (the $p \times p$ identity matrix) and using the L1 and L2 norms as penalty functions, respectively. These and other known particular examples of this general model are summarized in Table 1.

In order to derive a general modified Newton-Raphson algorithm for the MPLS model, it is important to find the gradient and the Hessian of the objective function of equation (1):

$$\begin{aligned}\nabla f(\boldsymbol{\beta}) &= -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \sum_{r=1}^{R} \lambda_r \sum_{i=1}^{N_r} \nabla\theta_i^{(r)}(\boldsymbol{\beta}) g_r{}'\left(\left|\theta_i^{(r)}\right|\right) sgn\left(\theta_i^{(r)}\right) \\ &= -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \sum_{r=1}^{R} \lambda_r \sum_{i=1}^{N_r} \mathbf{L}_i^{(r)T}\left(g_r{}'\left(\left|\theta_i^{(r)}\right|\right)/\left|\theta_i^{(r)}\right|\right)\mathbf{L}_i^{(r)}\boldsymbol{\beta} \\ \nabla^2 f(\boldsymbol{\beta}) &= \mathbf{X}^T\mathbf{X} + \sum_{r=1}^{R} \lambda_r \sum_{i=1}^{N_r} \mathbf{L}_i^{(r)T}\left(g_r{}''\left(\left|\theta_i^{(r)}\right|\right)\right)\mathbf{L}_i^{(r)}\end{aligned}$$

where the scalar magnitude $\theta_i^{(r)}$ is the i-th element of $\boldsymbol{\theta}^{(r)}$ and $sgn(\cdot)$ represents the sign function, which was conveniently written as $sgn(x) = x/|x|$; $\forall x \neq 0$ and $sgn(0) = 0$. In this expression, both $g_r{}'$ and $g_r{}''$ represent the first and second derivatives of the scalar function g_r with respect to $\left|\theta_i^{(r)}\right|$, while $\mathbf{L}_i^{(r)}$ represents the i-th row of the matrix $\mathbf{L}^{(r)}$. We now follow the same rationale used by the minorization-maximization (MM) algorithm of Hunter and Li [13]. They showed that the local quadratic approximation is an instance of an MM algorithm. Therefore, we need to verify that our general penalty function $\Psi(\boldsymbol{\beta}) = \sum_{r=1}^{R} \lambda_r \sum_{i=1}^{N_r} g_r\left(\left|\theta_i^{(r)}\right|\right)$ satisfies the conditions established in Proposition 3.1 presented in [5] that we rewrite here with our notation for an easier understanding of its application to the general MPLS case.

Name	Penalty term	Function definition
Ridge I	$\Psi = \lambda \sum_{i=1}^{p} g(\lvert\theta_i\rvert)$	$g(\lvert\theta\rvert) = \lvert\theta\rvert^2$; $\boldsymbol{\theta} = \boldsymbol{\beta}$
Ridge L		$g(\lvert\theta\rvert) = \lvert\theta\rvert^2$; $\boldsymbol{\theta} = \mathbf{L}\boldsymbol{\beta}$
LASSO		$g(\lvert\theta\rvert) = \lvert\theta\rvert$; $\boldsymbol{\theta} = \boldsymbol{\beta}$
Fusion LASSO		$g(\lvert\theta\rvert) = \lvert\theta\rvert$; $\boldsymbol{\theta} = \mathbf{L}\boldsymbol{\beta}$
Smooth LASSO (SLASSO)	$\Psi = \lambda_1 \sum_{i=1}^{p} g_1\left(\left\lvert\theta_i^{(1)}\right\rvert\right) + \lambda_2 \sum_{i=1}^{p} g_2\left(\left\lvert\theta_i^{(2)}\right\rvert\right)$	$g_1(\lvert\theta\rvert) = \lvert\theta\rvert$; $\boldsymbol{\theta}^{(1)} = \boldsymbol{\beta}$ $g_2(\lvert\theta\rvert) = \lvert\theta\rvert^2$; $\boldsymbol{\theta}^{(2)} = \boldsymbol{\Omega}\boldsymbol{\beta}$
Elastic Net (ENET L)		$g_1(\lvert\theta\rvert) = \lvert\theta\rvert$; $\boldsymbol{\theta}^{(1)} = \mathbf{L}\boldsymbol{\beta}$ $g_2(\lvert\theta\rvert) = \lvert\theta\rvert^2$; $\boldsymbol{\theta}^{(2)} = \mathbf{L}\boldsymbol{\beta}$
Adaptive LASSO (ALASSO)	$\Psi = \lambda \sum_{i=1}^{p} \gamma_i\, g(\lvert\theta_i\rvert)$	$g(\lvert\theta\rvert) = \lvert\theta\rvert$; $\boldsymbol{\theta} = \boldsymbol{\beta}$ with $\gamma_i > 0$ for i = $1, \ldots, p$

Nonnegative Garrote (NNG)	$\Psi = \lambda_1 \sum_{i=1}^{p} g_1(\lvert\theta_i^{(1)}\rvert) + \lambda_2 \sum_{i=1}^{p} \gamma_i\, g_2(\lvert\theta_i^{(2)}\rvert)$	$g(\lvert\theta\rvert) = \lvert\theta\rvert$; $\boldsymbol{\theta} = \boldsymbol{\beta}$ with $\gamma_i = 1/\lvert\beta_i^{ols}\rvert$ for i = 1, … , p $\boldsymbol{\beta} \geq 0 \quad if \quad \boldsymbol{\beta}^{ols} > 0$ $\boldsymbol{\beta} \leq 0 \quad if \quad \boldsymbol{\beta}^{ols} < 0$
Smooth Non-Negative Garrote (SNNG)		$g_1(\lvert\theta\rvert) = \lvert\theta\rvert$; $\boldsymbol{\theta}^{(1)} = \boldsymbol{\beta}$ $g_2(\lvert\theta\rvert) = \lvert\theta\rvert^2$; $\boldsymbol{\theta}^{(2)} = \mathbf{L}\boldsymbol{\beta}$ with $\gamma_i = 1/\lvert\beta_i^{ref}\rvert$; $\boldsymbol{\beta} \geq 0$
Non-Negative Smooth LASSO (NN-SLASSO)		$g_1(\lvert\theta\rvert) = \lvert\theta\rvert$; $\boldsymbol{\theta}^{(1)} = \boldsymbol{\beta}$ $g_2(\lvert\theta\rvert) = \lvert\theta\rvert^2$; $\boldsymbol{\theta}^{(2)} = \boldsymbol{\Omega}\boldsymbol{\beta}$ with $\gamma_i = 1$; $\boldsymbol{\beta} \geq 0$

Table 1: Known models represented as instances of the general MPLS model (equation (1)). Here, $\boldsymbol{\Omega} \in \mathbb{R}^{p\times p}$ is the first-differences operator and $\mathbf{L} \in \mathbb{R}^{p\times p}$ is a matrix used for imposing a correlation structure in the solution, typically being the first- or second-differences operator. The $\boldsymbol{\beta}^{ols}$ represents the ordinary least squares solution.

Proposition 1: (adapted from Proposition 3.1 in Hunter & Li, 2005 [13]**).** Suppose that on (0, ∞), $g(\cdot)$ is piecewise differentiable, nondecreasing and convex. Furthermore, it is continuous at 0 and $g'(0_+) < \infty$ (i.e., the limit of $g'(x)$ as $x \to 0$ from positive values is finite). Then, for all $\theta_0 \neq 0$, the magnitude defined as: $\Phi_{\theta_0}(\theta) = g(|\theta_0|) + (\theta^2 - \theta_0^2)g'(|\theta_0|_+) \,/\, 2\,|\theta_0|$, satisfies

a) $\Phi_{\theta_0}(\theta) \geq g(|\theta|)$ for all θ, with equality when $\theta = \pm|\theta_0|$; *i.e.*, $\Phi_{\theta_0}(\theta)$ majorizes $g(|\theta|)$

b) $\Phi_{\theta_0}(\theta) < \Phi_{\theta_0}(\theta_0)$ implies that $g(|\theta|) < g(|\theta_0|)$

Carrying out simple mathematical transformations, the penalty term of the objective function for MPLS models can be written as the sum of new functions:

$$\Psi(\boldsymbol{\beta}) = \sum\nolimits_{r=1}^{R} \lambda_r \sum\nolimits_{i=1}^{N_r} g_r\left(\lvert\theta_i^{(r)}\rvert\right) = \sum\nolimits_{i=1}^{N_r} \sum\nolimits_{r=1}^{R} \lambda_r g_r\left(\lvert\theta_i^{(r)}\rvert\right) = \sum\nolimits_{i=1}^{N_r} \mathcal{G}(|\theta_i|)$$

In simple terms, the sums can be exchanged only if all linear operators $\mathbf{L}^{(r)}$ are the same, such that the vector functions $\boldsymbol{\theta}^{(r)}(\boldsymbol{\beta}) = \mathbf{L}^{(r)}\boldsymbol{\beta}$ lead to the same variable with components $\theta_i^{(r)}$ for all *r*. In that case, it is easy to see that if every penalty function $g_r\left(\lvert\theta_i^{(r)}\rvert\right)$ satisfies the conditions of Proposition 1, then $\mathcal{G}(|\theta_i|) = \sum_{r=1}^{R} \lambda_r g_r\left(\lvert\theta_i^{(r)}\rvert\right)$ is piecewise differentiable on (0,+∞) and the results of Proposition 1 will hold. However, an algorithm for a more general case when the $\mathbf{L}^{(r)}$ are not the same, can be derived with the use of the local quadratic approximation proposed by [13] for all functions g_r, which converts the original penalty term in an equivalent quadratic penalty. To avoid numerical problems when $\theta_i \approx 0$, they proposed to modify the objective function (i.e., an approximation $f_\varepsilon(\boldsymbol{\beta})$) by perturbing the penalty function g, using some small $\varepsilon > 0$, as: $g_\varepsilon(|\theta|) = g(|\theta|) - \varepsilon \int_0^{|\theta|} \frac{g'(t)}{\varepsilon + t}\, dt$.

Applying this perturbation to every penalty function g_r in the MPLS model, we can obtain a local quadratic approximation for all of them as a function of a common variable $|\theta|$:

$$g_\varepsilon^{(r)}(|\theta|) \approx g_r\left(\lvert\theta_i^{(r)}\rvert\right) + \frac{\left(|\theta|^2 - \left|\theta_i^{(r)}\right|^2\right) g_r'\left(\left|\theta_i^{(r)}\right|_+\right)}{2\left(\varepsilon + \left|\theta_i^{(r)}\right|\right)},$$

where the symbol $f(\theta_+)$ denotes the limit of $f(x)$ as $x \to \theta$ from positive values. The Newton-Raphson technique is then used to minimize the perturbed objective function through its first and second derivatives:

$$\nabla f_\varepsilon(\boldsymbol{\beta}) = -\mathbf{X}^T\mathbf{y} + \left(\mathbf{X}^T\mathbf{X} + \textstyle\sum_{r=1}^{R} \lambda_r \mathbf{L}^{(r)T}\mathbf{D}^{(r)}\mathbf{L}^{(r)}\right)\boldsymbol{\beta}$$

$$\nabla^2 f_\varepsilon(\boldsymbol{\beta}) = \mathbf{X}^T\mathbf{X} + \textstyle\sum_{r=1}^{R} \lambda_r \mathbf{L}^{(r)T}\mathbf{D}^{(r)}\mathbf{L}^{(r)}$$

where $\mathbf{D}^{(r)}$ is a diagonal matrix with diagonal elements defined as $d_i^{(r)} = g_r'\left(\lvert\theta_i^{(r)}\rvert\right)/\left(\varepsilon + \lvert\theta_i^{(r)}\rvert\right)$ for $i = 1, \ldots, N_r$ and some very small $\varepsilon > 0$. Note that in this case, the diagonal matrix $\mathbf{D}^{(r)}$ is the same in both the gradient and the Hessian, as the penalty functions are now approximated as locally quadratic where $g''(|\theta|) = g'(|\theta|)/|\theta|$. Then, we can locally minimize the perturbed objective function $f_\varepsilon(\boldsymbol{\beta})$ for some $\alpha_k > 0$, using the iterative formula:

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - \alpha_k\{\nabla^2 f_\varepsilon(\boldsymbol{\beta}_k)\}^{-1}\nabla f_\varepsilon(\boldsymbol{\beta}_k) = \boldsymbol{\beta}_k + \alpha_k\left[\left(\mathbf{X}^T\mathbf{X} + \sum_{r=1}^{R} \lambda_r \mathbf{L}^{(r)T}\mathbf{D}^{(r)}\mathbf{L}^{(r)}\right)^{-1}\mathbf{X}^T\mathbf{y} - \boldsymbol{\beta}_k\right]$$

Note that $\|f(\boldsymbol{\beta}) - f_\varepsilon(\boldsymbol{\beta})\| \to 0$ and $\|\nabla f(\boldsymbol{\beta}) - \nabla f_\varepsilon(\boldsymbol{\beta})\| \to 0$ uniformly whenever $\varepsilon \to 0$. Thus, any limit point of the estimated sequence $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots$ represent a critical point of the original objective function $f(\boldsymbol{\beta})$ [14].

In our case, the objective function may be more complex if it combines different convex and non-convex penalty functions with correlation structure. In those cases, the NR algorithm can be stuck at saddle or local stationary points. However, the function $f(\boldsymbol{\beta})$ is convex for penalty functions based on L1 and L2 norms, e.g., Fused LASSO [27], Fusion LASSO [15] (Land and Friedman 1996) and Smooth LASSO [11]. Therefore, for these cases the MNR implementation and, in particular, the canonical version ($\alpha_k = 1$), achieves the global minimum. Finally, the parameter ε can be selected as proposed by Hunter & Li, (2005):

$$\varepsilon = \frac{tol}{2RM}\min\left\{\left|\theta_i^{(r)}\right| : \theta_i^{(r)} \neq 0\right\}, \text{for } i = 1, \dots, N_r \text{ and } r = 1, \dots, R,$$

where $M = \max\{g_r'(0_+)\}$, for $r = 1, \dots, R$, and $tol > 0$ is the convergence parameter (i.e. convergence is determined when an absolute change in every element of the vector solution is below a predefined value tol, such that $\left|\partial_j f_\varepsilon(\boldsymbol{\beta})\right| < tol/2$). The parameter ε becomes smaller through iterations but it is usually fixed after the first iteration to avoid numerical instability (see [10]).

We then propose a canonical version of the MNR algorithm in the table Algorithm 1, following the same rationale as Hunter and Li on the Minorization-Maximization algorithm. The basic difference with the MM is that, in step 5 of Algorithm 1, the sum of the derivatives of all penalty functions are included in the regularization of the design matrix. This is precisely what makes this algorithm general for any differentiable penalty functions g_r and linear operators $\mathbf{L}^{(r)}$.

This algorithm depends on the regularization parameters $\lambda_1, \dots, \lambda_R$, which can be chosen from a given grid of values or from an automatically determined range according to the singular values of $\mathbf{X}$. The selection of the 'optimal values' for these parameters is a process that will not be considered here in detail. This is usually done by minimizing information criteria such as Akaike (AIC), Bayesian (BIC) or the generalized cross-validation (GCV) function. For this purpose, it is necessary to compute the degrees of freedom, which can be estimated as proposed in Hunter & Li, (2005)[13]. In order to avoid the selection of optimal parameters in an R-dimensional grid, we prefer to set $\lambda_r = \lambda\mu_r$ and set ad hoc values for the proportions $\mu_r > 0$, for $\forall r = 1, \dots, R$, (such that $\sum \mu_r = 1$), which represent prior assumptions about relative penalty contributions and allow simplifying the process to estimating only the overall weight as a single parameter λ.

Algorithm 1. MNR for MPLS ($\mathbf{y} \in \mathbb{R}^{n\times 1}, \mathbf{X} \in \mathbb{R}^{n\times p}, \lambda_1, \dots, \lambda_R, \mathbf{L}^{(1)}, \dots, \mathbf{L}^{(r)} \in \mathbb{R}^{N_r\times p}$)

1. Start with k: $k \leftarrow 0$ and set $\tau \leftarrow 10^{-8}, \varepsilon \leftarrow 10^{-8}$, MaxIter$\leftarrow 100$ and $\boldsymbol{\Omega} \leftarrow \mathbf{I}_p$

2. Set $k \leftarrow k + 1$ and compute $\boldsymbol{\beta}_k \leftarrow (\mathbf{X}^T\mathbf{X} + \boldsymbol{\Omega})^{-1}\mathbf{X}^T\mathbf{y}$

3. Set $\boldsymbol{\theta}^{(r)} \leftarrow \mathbf{L}^{(r)}\boldsymbol{\beta}_k$ for $r = 1, \dots, R$ and compute

$$\mathbf{D}^{(r)} \leftarrow \operatorname{diag}\left(g_r'\left(\left|\theta_1^{(r)}\right|\right)/\left(\varepsilon + \left|\theta_1^{(r)}\right|\right), \dots, g_r'\left(\left|\theta_{N_r}^{(r)}\right|\right)/\left(\varepsilon + \left|\theta_{N_r}^{(r)}\right|\right)\right)$$

4. If $k = 1$, then set $M \leftarrow \max\{g_r'(0_+)\}$ and $\varepsilon \leftarrow \frac{\tau}{2RM}\min\left\{\left|\theta_i^{(r)}\right| : \theta_i^{(r)} \neq 0\right\}$

5. Set $\boldsymbol{\Omega} \leftarrow \sum \lambda_r \mathbf{L}^{(r)T}\mathbf{D}^{(r)}\mathbf{L}^{(r)}$ and compute $\boldsymbol{\delta} \leftarrow -\mathbf{X}^T\mathbf{y} + (\mathbf{X}^T\mathbf{X} + \boldsymbol{\Omega})\boldsymbol{\beta}_k$

6. If $\left|\delta_j\right| < \tau/2$ for all $j \in \{1, \dots, p\}$ such that$\left|\beta_j\right| \geq \varepsilon$, then **goto** Step 8.

7. If $k <$ MaxIter, then **goto** Step 2.

8. Stopping criterion: if convergence is reached then the solution is $\widehat{\boldsymbol{\beta}} \leftarrow \boldsymbol{\beta}_k$.

3. DERIVATION OF THE GENERAL ACTIVE-SET MODIFIED NEWTON-RAPHSON TECHNIQUE

Consider the MPLS optimization problem defined in its unconstrained variant, as the minimization of the functional stated in equation (1). Recall that $\Psi(\boldsymbol{\beta})$ is a sum of convex functions, which guarantees the convexity of the objective or cost function $f(\boldsymbol{\beta})$. An algorithm that solves this minimization problem can be a sequence of steps in the space of $\boldsymbol{\beta}$ that reduces the values of $f(\boldsymbol{\beta})$ until no more reduction can be achieved. Alternatively, the same algorithm can be interpreted as a sequence of steps in the space of $\boldsymbol{\beta}$ that solves the equation $\nabla f(\boldsymbol{\beta}) = 0$, as this become a sufficient condition of the global minimum when f is a convex function. Let's assume we are at step k of the minimization of $f(\boldsymbol{\beta})$ with coefficients vector $\boldsymbol{\beta}_k$ that will be updated as $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \mathbf{b}_k$. If we take into account that the effect of previous steps can be absorbed by the residuals: $\mathbf{r}_k = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}_k$, the cost function at step $k + 1$ can be expressed as a function of the vector update $\mathbf{b}_k$:

$$f_{k+1}(\mathbf{b}_k) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}_{k+1}\|_2^2 + \Psi(\boldsymbol{\beta}_k + \mathbf{b}_k)$$

$$f_{k+1}(\mathbf{b}_k) = \frac{1}{2}\|\mathbf{r}_k - \mathbf{X}\mathbf{b}_k\|_2^2 + \Psi(\boldsymbol{\beta}_k + \mathbf{b}_k)$$

The change in the cost function from the previous iteration $\left(f_k = f(\boldsymbol{\beta}_k) = f_{k+1}(\boldsymbol{b}_k)|_{\boldsymbol{b}_k=0}\right)$ to the next iteration $(f_{k+1}(\mathbf{b}_k) = f(\boldsymbol{\beta}_k + \mathbf{b}_k))$, can be found from the second term in the first-order Taylor approximation of $f_{k+1}(\mathbf{b}_k)$ around $\mathbf{b}_k = 0$:

$$f_{k+1}(\mathbf{b}_k) \approx f_{k+1}(\mathbf{b}_k)|_{\mathbf{b}_k=0} + \left(\nabla_{\mathrm{b}} f_{k+1}(\mathbf{b}_k)|_{\mathbf{b}_k=0}\right)^T \mathbf{b}_k$$

$$f_{k+1}(\mathbf{b}_k) \approx f_k - \left(\mathbf{X}^T\mathbf{r}_k - \nabla\Psi(\boldsymbol{\beta}_k)\right)^T \mathbf{b}_k \quad (2)$$

where we have used that the gradient of the continuous and differentiable penalty function with respect to $\mathbf{b}_k$, evaluated at $\mathbf{b}_k = 0$, is equal to the gradient with respect to $\boldsymbol{\beta}_k$, i.e. $\nabla_{\mathrm{b}}\Psi(\boldsymbol{\beta}_k + \mathbf{b}_k)|_{\mathbf{b}_k=0} = \nabla_{\boldsymbol{\beta}_k}\Psi(\boldsymbol{\beta}_k) = \nabla\Psi(\boldsymbol{\beta}_k)$, which is easy to verify.

Interestingly, the change in the cost function $f_{k+1} - f_k = -\left(\mathbf{X}^T\mathbf{r}_k - \nabla\Psi(\boldsymbol{\beta}_k)\right)^T\mathbf{b}_k = -\nabla f(\boldsymbol{\beta}_k)^T(\boldsymbol{\beta}_{k+1} - \boldsymbol{\beta}_k)$, for small update vectors, is proportional to the product between the residuals and the predictors (columns of $\mathbf{X}$) minus the gradient of the penalization term, and convergence will be achieved when these terms become equal. From this equation, we can then follow a procedure similar to the one used to derive the LARS algorithm (Efron et al. 2004), in order to find the update (size and direction) $\mathbf{b}_k$ that ensures the minimization of the cost function in each iteration (i.e., $f_{k+1} \leq f_k$ for all k). This implies that $f_{k+1} - f_k \leq 0$, which leads to the local-global minimum condition $\nabla f(\boldsymbol{\beta}_k)^T(\boldsymbol{\beta}_{k+1} - \boldsymbol{\beta}_k) \geq 0$. In the case that $\boldsymbol{\beta}_k$ reaches a minimum of the cost function, say $\boldsymbol{\beta}_k = \boldsymbol{\beta}^*$, this is a sufficient condition for $\boldsymbol{\beta}^*$ to be a global minimum of f over a convex set S, as long as the cost function is continuously differentiable and its gradient is continuous on S [1], [2]. A general optimality condition will be formally introduced in the next Section, from which -and a proper definition of S- two important properties of the solutions are derived. In this work we will refer to those properties as the two optimality conditions of the problem, i.e., a set of sufficient conditions for the estimator $\boldsymbol{\beta}_k$ at each iteration to be a local minimum, which means that such algorithm will provide the path of optimal solutions along iterations.

The optimality conditions allow us to derive a general Active-set Modified Newton-Raphson (AMNR) algorithm for MPLS models, as they impose a relationship between the gradient of the cost function and its minimum. As in classical optimization problems, the active set is defined as the set of coefficients that comply with the (usually inequality) constraints of the problem. We will show that in our formulation and derived algorithm the solution in every iteration will fulfill this condition. Since the solution consists in a few nonzero coefficients, in this work we restrict our definition of the active set to be the set of coefficients that are nonzero in every iteration, i.e. $\mathcal{A}=\{j = 1,..., l: \beta_j \neq 0\}$. This is convenient to represent the subset of the design matrix that contains only those columns corresponding to coefficients in the active set (changing in every iteration) and also to formulate the properties of the algorithm that only depend on those coefficients.

The first optimality condition implies that nonzero coefficients (components of the vector solution included in the active set) and their corresponding derivative of the cost function (components of the gradient), must have opposite signs. The other optimality condition implies that the absolute value of the gradient for all coefficients included in the active set is the same and is the highest among all predictors. These conditions can be fulfilled by selecting in every iteration one coefficient to be included in the active set, as the one with the highest absolute value of the gradient, while using other empirical procedures to ensure that the sign of the gradient is opposite to that of the coefficient itself. In particular, a general LARS-type constraint to be fulfilled in every iteration k is:

$$|\nabla f(\boldsymbol{\beta}_k)| = \left|\mathbf{X}_{\mathcal{A}}^{\mathrm{T}}\mathbf{r}_k - \nabla\Psi(\boldsymbol{\beta}_k)\right| = C_{max}\mathbf{1}_{\mathcal{A}} \text{ with } C_{max} > 0$$

where the active set is represented as $\mathcal{A}$, the columns of the corresponding predictors form the matrix $\mathbf{X}_{\mathcal{A}}$, and $\mathbf{1}_{\mathcal{A}}$ represents a vector of ones with length equal to the cardinality of the active set (later denoted by $|\mathcal{A}|$ as it changes in every iteration). This implies that the absolute value of the gradient for all coefficients included in the active set will be equal to the value C_{max}. The common use of the absolute value in this condition in LARS aims at controlling the sign of the updated coefficients and ensuring that the last term in equation (2) is positive [4]. We propose that a more natural way of taking this into account is by explicitly including the sign of the solution in the equation (i.e., $sgn\left(\nabla f(\boldsymbol{\beta}_k)\right) = -sgn(\boldsymbol{\beta}_k)$, as imposed by the first optimality condition):

$$\nabla f(\boldsymbol{\beta}_k) = |\nabla f(\boldsymbol{\beta}_k)| sgn\left(\nabla f(\boldsymbol{\beta}_k)\right)$$

$$-\left(\mathbf{X}_{\mathcal{A}}^{\mathrm{T}}\mathbf{r}_k - \nabla\Psi(\boldsymbol{\beta}_k)\right) = -C_{max} sgn(\boldsymbol{\beta}_k)$$

$$\mathbf{X}_{\mathcal{A}}^{\mathrm{T}}\mathbf{r}_k - \nabla\Psi(\boldsymbol{\beta}_k) = C_{max} sgn(\boldsymbol{\beta}_k) \quad (3)$$

where the component-wise sign function of a vector ($sgn(\mathbf{x})$) returns a vector containing the sign of each component of the argument ($\mathbf{x}$).

The derivation of the more general MPLS problem deserves a subsequent theoretical paper. We here follow a simpler approach consisting in making use of the quadratic approximation of the MNR procedure explained in the previous Section. Briefly, an MPLS model consisting of multiple penalties $\left(\Psi(\boldsymbol{\beta}) = \sum_{r=1}^{R}\lambda_r \sum_{i=1}^{N_r} g_r\left(\left|\theta_i^{(r)}\right|\right)\right)$, can be rewritten as a general quadratic model by using the quadratic approximations of g_r functions; i.e. obtaining $\Psi(\boldsymbol{\beta}) = \|\mathbf{W}\boldsymbol{\beta}\|_2^2$ (where $\mathbf{W} = \left(\sum_{r=1}^{R}\lambda_r \mathbf{L}^{(r)T}\mathbf{D}^{(r)}\mathbf{L}^{(r)}\right)^{1/2}$ is a general matrix that combines all linear operators for the different penalty functions). However, this also means that any model formed by a combination of penalty functions based on L1 and L2 norms can be taken to a simple LASSO or Adaptive LASSO model, by joining all other penalties (except one based on the L1-norm) into a quadratic term and using the trick of data augmentation

[11]. Therefore, in this work we derive and implement one of the simplest versions of the AMNR algorithm, which corresponds to the Adaptive LASSO model, conveniently established on the constrained equivalent formulation:

$$\text{minimize } \|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 \text{ subject to } \sum_{j=1}^{p}\gamma_j|\beta_j| \le \tau,$$

where γ_j are positive weights, thus reducing this model to LASSO when they are all set to 1. This formulation avoids the explicit use of regularization parameters by replacing them by a thresholding parameter ($\tau > 0$). Importantly, it is easy to see, from Table 1, that this model includes the NonNegative Garrote (NNG), introduced by Breiman as a variable selection technique that shrinks the ordinary least squares (OLS) estimator, in order to give intermediate results between OLS and subset selection [3]. This means that the AMNR can be used to implement this method but also to extend to use other reference solutions instead of the OLS, and to include an L2-norm term that allows to impose smoothness with this method, which we will call the Smooth Nonnegative Garrote (SNNG). A similar method would be the Smooth LASSO, but this one does not use weights obtained from other reference solutions. Anyway, the AMNR algorithm allowed us to implement a nonnegative version of the Smooth LASSO that will be called NN-SLASSO, in order to compare both models and specifically the influence of the reference solutions. These two extensions are also formulated in Table 1.

The AMNR implementation is very similar to the LARS algorithm (see [4]) for LASSO; however, it does not require predictors to be standardized and can also be used to minimize a continuously differentiable objective function while imposing sign constraints over the parameters. As in LARS, the active set is updated at every iteration in one of two ways: 1) including a new predictor or 2) excluding an existent one; such that the active sets in two consecutive iterations differ only by one predictor. If the optimality conditions are satisfied within the algorithm, the solutions (determined by nonzero coefficients included in the active set in every iteration) will always comply with the constraint of the optimization problem ($\sum_{j=1}^{p}\gamma_j|\beta_j| \le \tau$). Therefore, in practice, we do not need to check that this condition holds and it is usually easier to use a simpler stop criterion based on the convergence of the absolute value of the gradient to a value smaller than a tolerance (tol). Although it is not straightforward, there exist a relationship such that we can fix a value for τ and find the corresponding tolerance to be used in the algorithm (i.e., hard constraint). If the tol is fixed first, then the level of the constraint τ will depend on the values attained by coefficients in the final solution (i.e., soft constraint).

4. OPTIMALITY CONDITIONS FOR ADAPTIVE LASSO

In this section we formulate and prove a Theorem that proposes two important properties of the solution of the Adaptive LASSO model in the multiple penalized regression context. This model is particularly useful because many MPLS models based on combinations of L1 and L2 norms can be reduced to its formulation. Although the properties are derived from the classical optimality condition for this type of optimization problem, in this work we will refer to the two properties as the *optimality conditions* of the optimization problem.

Theorem 1: Consider the optimization problem $\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{f(\boldsymbol{\beta})\}$, subject to $\boldsymbol{\beta} \in S$, where $f: \mathbb{R}^{p\times 1} \to \mathbb{R}$ is a smooth (in particular, continuously differentiable) convex function, $S = \{\boldsymbol{\beta} : \sum \gamma_j|\beta_j| \le \tau\}$, $\gamma_j > 0\ \forall j$, and $\tau > 0$ is a given scalar

Then, if $\boldsymbol{\beta}^*$ is a local minimum of f over S, the following properties hold:

a) If $\beta_j^* > 0$ then $\partial_j f(\boldsymbol{\beta}^*) \le 0$.

b) If $\beta_j^* < 0$ then $\partial_j f(\boldsymbol{\beta}^*) \ge 0$.

c) If $|\beta_j^*| > 0$ then $\gamma_i|\partial_j f(\boldsymbol{\beta}^*)| \ge \gamma_j|\partial_i f(\boldsymbol{\beta}^*)|, \forall i$.

Proof:

First note that as $|\beta_j|$ is a convex function and $\gamma_j > 0$ the constraint describing S is convex. So, S is a convex subset of $\mathbb{R}^p$. As f is a smooth convex function, the Propositions 4.7.1 and 4.7.2 of Bertsekas et al. 2003 [1] establish that $\boldsymbol{\beta}^*$ is a local minimum of f over S if and only if the following optimality condition is satisfied:

$$\nabla f(\boldsymbol{\beta}^*)^T(\boldsymbol{\beta}-\boldsymbol{\beta}^*) \ge 0 \text{ for all } \boldsymbol{\beta} \in S \qquad \text{(T1)}$$

which can also be expressed as:

$$\sum_{j=1}^{p}\partial_j f(\boldsymbol{\beta}^*)(\beta_j-\beta_j^*) \ge 0, \qquad \forall \boldsymbol{\beta} \in S$$

In order to verify a), suppose that $\beta_j^* > 0$ for some $j \in \{1, \dots, p\}$ and $\boldsymbol{\beta}$ is a feasible solution to the problem (that is $\sum \gamma_j|\beta_j| \le \tau$). For instance, take $\beta_i = \beta_i^*$, for every $i \ne j$, and $\beta_j = \beta_j^* - \varepsilon$, for $0 < \varepsilon < \beta_j^*$. Therefore, if we apply the condition (T1), we get inequality $-\varepsilon\partial_j f(\boldsymbol{\beta}^*) \ge 0$, which means that $\partial_j f(\boldsymbol{\beta}^*) \le 0$.

Similarly, if $\beta_j^* < 0$ for some $j \in \{1, \dots, p\}$, we can take $\beta_i = \beta_i^*$, for every $i \neq j$, and $\beta_j = \beta_j^* + \varepsilon$, for $0 < \varepsilon < -\beta_j^*$. Applying the optimality condition (T1), we get $\varepsilon \partial_j f(\boldsymbol{\beta}^*) \geq 0$, which means that $\partial_j f(\boldsymbol{\beta}^*) \geq 0$. This shows b)

To prove c), suppose $\beta_j^* \neq 0$, in particular $\beta_j^* > 0$, for some $j \in \{1, \dots, p\}$, and let i be another index. Take a feasible solution β such that $\beta_j = \beta_j^* - \varepsilon/\gamma_j$, for some $0 < \varepsilon < \gamma_j \beta_j^*$, while $\beta_i = \beta_i^* + \varepsilon/\gamma_i$ and $\beta_k = \beta_k^*$ for all other indices $k \notin \{i, j\}$. Applying the optimality condition (T1), we obtain $\varepsilon(\partial_i f(\boldsymbol{\beta}^*)/\gamma_i - \partial_j f(\boldsymbol{\beta}^*)/\gamma_j) \geq 0$, which implies that $\gamma_j \partial_i f(\boldsymbol{\beta}^*) \geq \gamma_i \partial_j f(\boldsymbol{\beta}^*)$. Similarly, by taking $\beta_i = \beta_i^* - \varepsilon/\gamma_i$, we obtain $-\varepsilon(\partial_i f(\boldsymbol{\beta}^*)/\gamma_i - \partial_j f(\boldsymbol{\beta}^*)/\gamma_j) \geq 0$, which implies that $-\gamma_j \partial_i f(\boldsymbol{\beta}^*) \geq \gamma_i \partial_j f(\boldsymbol{\beta}^*)$. From these and condition a), we conclude that $\gamma_i |\partial_j f(\boldsymbol{\beta}^*)| = -\gamma_i \partial_j f(\boldsymbol{\beta}^*) \geq \gamma_j |\partial_i f(\boldsymbol{\beta}^*)|$. On the other hand, if $\beta_j^* < 0$ is chosen with the same considerations, the same procedure leads us to $\gamma_i |\partial_j f(\boldsymbol{\beta}^*)| = \gamma_i \partial_j f(\boldsymbol{\beta}^*) \geq \gamma_j |\partial_i f(\boldsymbol{\beta}^*)|$ and thus, condition c) is proved. ■

5. AMNR ALGORITHM FOR THE ADAPTIVE LASSO MODEL

In this section we show that an algorithm based on the AMNR technique for the Adaptive LASSO model guarantees that optimality conditions in Theorem 1 are sufficient conditions for the estimator at each iteration to be a local minimum, and thus allows obtaining the path of optimal solutions. In principle, similar algorithms can be derived from equation (3) for any other model that is an instance of the MPLS as shown in Section 3. We present the pseudo code of the AMNR algorithm for Adaptive LASSO (Algorithm 2) and then provide two Propositions to prove that the procedure for selecting the coefficients to be included in the active set and the selection of the update direction and step size, ensure that the algorithm offers feasible solutions that are also optimal in every iteration.

Algorithm 2. AMNR algorithm for Adaptive LASSO (Input: $\mathbf{y} \in \mathbb{R}^{n \times 1}$, $\mathbf{X} \in \mathbb{R}^{n \times p}, \gamma_1, \dots, \gamma_p$)

Step description	Formula
Initialization	$k \leftarrow 0, tol \leftarrow 10^{-8}, \mathcal{A} \leftarrow \{\ \}, \mathcal{A}^C \leftarrow \{1,2,\dots,p\},\ \boldsymbol{\beta}_0 \leftarrow \mathbf{0}_p$
Loop over k until stop condition	$k \leftarrow k + 1$
Step 1. Compute the residuals and correlations between predictor and residuals.	$\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{X}\boldsymbol{\beta}_{k-1}$ $\mathbf{c}_k \leftarrow \mathbf{X}^T \mathbf{r}_k$
Step 2. Select the next predictor j^* to be included in the active set $\mathcal{A}$. Exclude it from $\mathcal{A}^C$, and update the maximum correlation C_k.	$j^* = argmax_j \{ \lvert c_{kj} \rvert / \gamma_j : j \in \mathcal{A}^C \}$ $\mathcal{A} \leftarrow [\mathcal{A} \cup \{j^*\}]$ $C_k \leftarrow \lvert c_{kj^*} \rvert / \gamma_{j^*}$
Step 3. Compute the update direction $\boldsymbol{\delta}_k$ (OLS over $\mathcal{A}$ and residuals $\mathbf{r}_k$).	$\boldsymbol{\delta}_k \leftarrow (\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}})^{-1} \mathbf{X}_{\mathcal{A}}^T \mathbf{r}_k,\ j \in \mathcal{A};\quad \overline{\boldsymbol{\delta}}_k \leftarrow [\boldsymbol{\delta}_k^T\ \mathbf{0}^T]^T$ $\mathbf{a}_k \leftarrow \mathbf{X}^T \mathbf{X}_{\mathcal{A}} \boldsymbol{\delta}_k$
Step 4. Compute the step size α for this iteration by selecting the minimum of three possible conditions.	$\alpha_k{}^+ = \min\{1, (C_k \gamma_j - c_{kj})/(C_k \gamma_j - a_{kj}) : a_{kj} < C_k \gamma_j, j \in \mathcal{A}^C\}$ $\alpha_k{}^- = \min\{1, (C_k \gamma_j + c_{kj})/(C_k \gamma_j + a_{kj}) : a_{kj} > -C_k \gamma_j, j \in \mathcal{A}^C\}$ $\alpha_k{}^0 = \min\{1, -\beta_{k-1j}/\delta_{kj} : \beta_{k-1j}(\beta_{k-1j} + \delta_{kj}) < 0, j \in \mathcal{A}\}$ $\alpha_k \leftarrow \min\{\alpha^+, \alpha^-, \alpha^0\}$
Step 5. Update coefficients	$\boldsymbol{\beta}_k \leftarrow \boldsymbol{\beta}_{k-1} + \alpha_k \overline{\boldsymbol{\delta}}_k$
Step 6. Verify stopping conditions	$\mathcal{A}^C$ is empty or if $\lvert c_{jk} \rvert / \gamma_j \leq tol$ for all $j \in \mathcal{A}^C$

Proposition 2. Given the model for obtaining the Adaptive LASSO solution to the problem of minimizing Equation (1), where $\Psi = \sum_{i=1}^{p} \gamma_i\, g(|\beta_i|)$ with $g(|\beta|) = |\beta|$ and $\gamma_i > 0, \forall i$, then:

a) The LARS-type constraint (also known as the stationarity condition) to ensure that the gradient of the cost function is minimum in each iteration becomes:

$$\mathbf{X}_{\mathcal{A}}^T \mathbf{r}_k = C_k \boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_k) \tag{4}$$

b) The minimum value of the gradient of the cost function corresponds to taking the maximum of the absolute gradient, weighted by the inverse of penalty weights:

$$C_k = \max(|\boldsymbol{\Gamma}^{-1} \mathbf{X}^T \mathbf{r}_k|)$$

and this selection ensures the fulfillment of part b) of Theorem 1.

Proof:

The gradient of the penalty function of the Adaptive LASSO model can be conveniently written as $\nabla\Psi = \boldsymbol{\Gamma} sgn(\boldsymbol{\beta})$, where the diagonal (squared) matrix $\boldsymbol{\Gamma}$ contains the positive penalty weights γ_i for each coefficient β_i. It is straightforward to see that its value is the same if we reduce the matrix and the vector of signs to the subset of

coefficients that are in the active set at iteration k, since the rest of the coefficients are zero: $\nabla\Psi(\boldsymbol{\beta}_k) = \boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_k)$. Introducing this in the general stationarity condition given by Equation (3) we can obtain:

$$\mathbf{X}_{\mathcal{A}}^{\mathrm{T}}\mathbf{r}_k - \boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_k) = C_{max} sgn(\boldsymbol{\beta}_k)$$
$$\mathbf{X}_{\mathcal{A}}^{\mathrm{T}}\mathbf{r}_k = (C_{max}\boldsymbol{\Gamma}_{\mathcal{A}}^{-\mathbf{1}} + 1)\boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_k)$$

Then, if we select $C_k = C_{max}\boldsymbol{\Gamma}_{\mathcal{A}}^{-\mathbf{1}} + 1$, this magnitude will be positive since C_{max} is positive by definition and substituting it we obtain the equation (4) and prove the part a) of this Proposition.

For proving part b) it is easy to use the definition of C_k to find that $C_{max} = (C_k - 1)\boldsymbol{\Gamma}_{\mathcal{A}}$. Then, the coefficient that makes C_{max} to have the maximum value in each iteration will be the one that also makes C_k maximum. From equation (3), the maximum value of C_{max} is the maximum of the full negative gradient, while from equation (4) we obtain $C_k = \max(|\boldsymbol{\Gamma}_{\mathcal{A}}^{-\mathbf{1}}\mathbf{X}_{\mathcal{A}}^{\mathrm{T}}\mathbf{r}_k|)$, which means that C_k will take the (absolute) value of the component of the vector $\boldsymbol{\Gamma}_{\mathcal{A}}^{-\mathbf{1}}\mathbf{X}_{\mathcal{A}}^{\mathrm{T}}\mathbf{r}_k$ with the maximum absolute value. Equation (4) then implies that the absolute values of the gradient, divided by the weights of the penalty, is the same for all coefficients included in the active set. However, in order to decide which coefficient will enter the active set, the index corresponding to this maximum value must be found among all coefficients that are not in the active set in every iteration.

The generalization of this expression to compute C_k for the whole set of coefficients is straightforward, just including all columns of $\mathbf{X}$ and the corresponding columns and rows of $\boldsymbol{\Gamma}$, leading to $C_k = \max(|\boldsymbol{\Gamma}^{-1}\mathbf{X}^T\mathbf{r}_k|$. Then, it is easy to verify that for each coefficient in the active set we get $C_k \leftarrow |\mathbf{x}_{j^*}^T\mathbf{r}_k|/\gamma_{j^*}$, where j^* is last index selected to be included in the active set (Step 2 of the algorithm). Now recognizing that the gradient of the unconstrained problem is $\nabla f = -\mathbf{X}^T\mathbf{r}_k$, with $\partial_{j*}f = \mathbf{x}_{j^*}^T\mathbf{r}_k$, we obtain that $\frac{|\partial_{j*}f|}{\gamma_{j*}} \geq \frac{|\partial_i f|}{\gamma_i}, \forall\, i = 1,2,\ldots,p$, since $C_k = \frac{|\partial_{j*}f|}{\gamma_{j*}}$ is the same for all indices of coefficients included in the active set and it was selected as the maximum among all indices of coefficients not in the active set. This leads directly to the condition given in part b) of Theorem 1, which completes the proof of the second part of this Proposition. ■

Proposition 3. The selection of the Newton-Raphson direction and the step size for updating the coefficients, given in Steps 3-4 of the AMNR algorithm for the Adaptive LASSO model, ensures the fulfillment of the global-local minima condition for the model. It also ensures the optimality condition a) given in Theorem 1, which states that the sign of every coefficient included in the active set does not change from one iteration to another, as long as they remain in the active set.

Proof:

In order to prove both statements of the Proposition, we show that if we start from equation (4) -which was derived from the original local-global minimum condition- we can obtain the Newton Raphson direction and the step size shown in Step 4 of the AMNR algorithm, used for updating the coefficients. We then write the update vector as $\mathbf{b}_k = \alpha_k\boldsymbol{\delta}_k$ to explicitly separate it in size (α_k) and direction ($\boldsymbol{\delta}_k$), with the latter being nonzero for those coefficients in the active set, thus we will work with $\boldsymbol{\delta}_k$ and $\boldsymbol{\beta}_k$ as vectors with only $|\mathcal{A}|$ components in each iteration. Writing equation (4) for iteration $k+1$, and substituting residuals $\mathbf{r}_{k+1} = \mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\beta}_k = \mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\beta}_{k-1} - \mathbf{X}_{\mathcal{A}}\mathbf{b}_k = \mathbf{r}_k - \alpha_k\mathbf{X}_{\mathcal{A}}\boldsymbol{\delta}_k$, for those coefficients included in the active set, we get:

$$\mathbf{X}_{\mathcal{A}}^T\mathbf{r}_{k+1} = C_{k+1}\boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_{k+1})$$
$$\mathbf{X}_{\mathcal{A}}^T\mathbf{r}_k - \alpha_k\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}\boldsymbol{\delta}_k = C_{k+1}\boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_{k+1})$$
$$C_k\boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_k) - \alpha_k\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}\boldsymbol{\delta}_k = C_{k+1}\boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_{k+1})$$
$$\alpha_k\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}\boldsymbol{\delta}_k = (C_k - C_{k+1})\boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_k)$$

The last step assumed that the sign of coefficients included in the active set does not change from one iteration to the next one. This is ensured within the algorithm, as will be proved in the following paragraphs. From this equation we can solve for both the size and direction by splitting it in vector and scalar part. The vector part leads to the Newton-Raphson direction when conveniently dividing by C_k in both sides, and using equation (4) again:

$$\boldsymbol{\delta}_k = (\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}})^{-1}C_k\boldsymbol{\Gamma}_{\mathcal{A}} sgn(\boldsymbol{\beta}_k)$$
$$\boldsymbol{\delta}_k = (\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{X}_{\mathcal{A}}^T\mathbf{r}_k$$

while the scalar part leads to $\alpha_k = (C_k - C_{k+1})/C_k$, or equivalently $C_{k+1} = C_k(1 - \alpha_k)$. This means that the AMNR algorithm uses the Newton-Raphson direction to update the coefficients in the active set and move over the space of optimal solutions $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \alpha\boldsymbol{\delta}_{k+1}$ for some $\alpha \in (0, \alpha_{k+1}]$, where $0 < \alpha_{k+1} \leq 1$ and $\boldsymbol{\beta}_{k+1}$ not including here any new predictor yet.

The second part of the Proposition implies that the algorithm will comply with the optimality condition a) only if $sgn(\boldsymbol{\beta}_k) = sgn(\boldsymbol{\beta}_{k+1})$ for all coefficients that remain in the active set. Therefore, at every iteration we first need to check if exists a step size $\alpha \in (0, \alpha_{k+1}]$ such that any active coefficient becomes zero and should be removed from the active set, i.e., if for any coefficient $i \in \mathcal{A}$, it holds that $\beta_{k_i} + \alpha\delta_{k+1_i} = 0$. This leads to compute, at iteration k, the value $\alpha_k^0 = \min^+\left\{\alpha_j^0 = -\beta_{k_j}/\delta_{k_j} : \beta_{k_j}\left(\beta_{k_j} + \delta_{k_j}\right) < 0,\ j \in \mathcal{A}\right\}$, where $\min^+$ indicates that the minimum is taken considering only the positive values for computed elements. This means that if any coefficient in the active set will change the sign when updating its value, the step size is chosen to make that coefficient 0,

i.e., to remove it from the active set. As the algorithm continues, this coefficient is again available to be included in the active set in subsequent iterations with any sign, as explained in the next paragraph.

For those non-active predictors ($\mathbf{x}_j: j \in \mathcal{A}^C$), equation (4) is not valid at iteration k, but it must be valid for the new predictor $\mathbf{x}_{j^*}$ selected to join the active set at iteration $k+1$. Therefore, we have $\mathbf{x}_{j^*}^T\mathbf{r}_k - \alpha_k\mathbf{x}_{j^*}^T\mathbf{X}_{\mathcal{A}}\boldsymbol{\delta}_k = C_{k+1}\gamma_{j^*}sgn\left(\beta_{k+1\,j^*}\right)$. From this condition, using the relation $C_{k+1} = C_k(1-\alpha_k)$ and a shorter notation for the correlation of the predictor with residuals ($c_{k\,j^*} = \mathbf{x}_{j^*}^T\mathbf{r}_k$) and with the projection of the update direction ($a_{k\,j^*} = \mathbf{x}_{j^*}^T\mathbf{X}_{\mathcal{A}}\boldsymbol{\delta}_k$), we obtain the analytical expression for the step size α_k needed for computing the coefficient corresponding to the non-active predictor $\beta_{k+1\,j^*}$ to be included in the active set:

$$c_{k\,j^*} - \alpha_k a_{k\,j^*} = C_{k+1}\gamma_{j^*}sgn\left(\beta_{k+1\,j^*}\right)$$
$$c_{k\,j^*} - \alpha_k a_{k\,j^*} = C_k(1-\alpha_k)\gamma_{j^*}sgn\left(\beta_{k+1\,j^*}\right)$$
$$\alpha_k = \frac{C_k\gamma_{j^*}sgn\left(\beta_{k+1\,j^*}\right) - c_{k\,j^*}}{C_k\gamma_{j^*}sgn\left(\beta_{k+1\,j^*}\right) - a_{k\,j^*}}$$

The sign of the coefficient to be computed is not known a priori, but this expression implies that the step sizes needed to enter a coefficient in the active set with positive or negative values will be different:

$$\alpha_k^+ = \frac{C_k\gamma_j - c_{k\,j}}{C_k\gamma_j - a_{k\,j}}$$
$$\alpha_k^- = \frac{-C_k\gamma_j - c_{k\,j}}{-C_k\gamma_j - a_{k\,j}} = \frac{C_k\gamma_j + c_{k\,j}}{C_k\gamma_j + a_{k\,j}}$$

Therefore, as long as step sizes are valid (i.e., they are positive), this means that -in any iteration k- if the minimum step size for a predictor to join the active set corresponded to α_k^+ (α_k^-), then the corresponding coefficient will take a positive (negative) value when included in the active set. This is a very useful result, as it allows us to have a natural way to impose nonnegative or nonpositive constraints. For instance, if we want to find the best positive (negative) solution we can just ignore the step sizes that will lead to negative (positive) coefficients and include in the active set only coefficients that will enter with a positive (negative) value. As we are also ensuring that all coefficients in the active set will keep the same sign or removed from the active set if they tend to change sign, it turns out that all coefficients in the final solution will have the same sign and we will get properly sign-constrained solutions.

Finally, recall that the fundamental property of the algorithm is that the absolute gradient of the objective function $|\mathbf{X}_{\mathcal{A}}^T\mathbf{r}_k|$ monotonically decreases along the selected step direction with a positive step size. As a result, there will always be a step size ensuring that a new predictor, not included in the active set, joins the active set. This step size should be taken to be the minimum among all predictors, no matter if it is for including a negative or a positive coefficient. However, another important point is to keep the signs of the coefficients already in the active set unchanged, therefore, the step size for which one of these coefficients becomes zero α_k^0 should be used if it is smaller than α_k^+ and α_k^-. With these considerations, the final estimate for the step size becomes $\alpha_k = \min^+\left\{\alpha_k^0, \alpha_{k\,j}^+, \alpha_{k\,j}^-: j \in \mathcal{A}^C\right\}$. ■

The two Propositions presented in this section show that the proposed AMNR algorithm fulfills the optimality conditions for the Adaptive LASSO model (Theorem 1), ensuring than in every iteration the solution is optimal and the gradient or the objective function is decreased. In essence, the convergence properties of the algorithm are similar to those of the MNR (MM algorithm, [16]) in each iteration. Therefore, convergence of the AMNR algorithm is ensured, since the gradient of the cost function is decrease in each iteration. Similarly, the order of the algorithm is the same as an OLS solution in every iteration, but in this case not all iterations have the same computational complexity as they have different sizes of the effective parameters to be estimated (active set). Assuming that there are no many removals of coefficients from the active set, i.e., every iteration the active set only increase by one, for a sparse solution (number of iterations equal or less than n) the order of all iterations together is $O(n^3)$, while for the worst case in which the algorithm needs to compute the whole set of coefficients, the complexity would be at most $O(p^2)$. Although we explore the computational time empirically with simulations in the next section, this property makes it a faster alternative to deal with high dimensional problems and multiple penalized models. The computational time can be shortened by selecting other stopping criterion or in case of a prior knowledge about the number of nonzero coefficients of the final solution. Since the algorithm controls the number of nonzero values (changing in just one every iteration), this allows to control or establish the exact sparsity of the final solution.

6. PRELIMINARY VALIDATION USING SIMULATED DATA

We used a simple simulation study to assess the performance of the AMNR algorithm in comparison with the use of the MNR algorithm (non active-set, equivalent to minorization-maximization in many models) and the LARS algorithm (active-set but updating in the OLS direction). One hundred independent repetitions of data were simulated using the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with p = 200 predictors and n = {10, 50, 100} observations, thus having three different n/p ratios (n/p = {0.05, 0.25, 0.5}), which reflects the level of ill-posedness (the lowest ratio, the highest ill-posedness). We simulated a "true" solution showing three nonzero regions: the 'bell', the 'square' and the 'point' sources. Mathematically, we set $\beta_j = 0$ except for:

$$\beta_j = \begin{cases} e^{-0.015(j-50)^2}, & \text{for } 30 < j < 70 \text{ (bell)} \\ 1, & \begin{cases} \text{for } 95 < j < 105 \text{ (square)} \\ \text{and } j = 150 \text{ (point)} \end{cases} \end{cases}$$

The predictors $\mathbf{X}_j$ and the noise $\boldsymbol{\varepsilon}$ are sampled from a standard normal distribution for each of the 100 simulations. The simulated true solution is not easy to recover with classical particular models based on L2 or L1 norm, as it is a sparse piece-wise combination of smooth, constant and isolated coefficients. We choose this construction in order to evaluate the flexibility of the MPLS models that uses combinations of penalty functions based on different norms. We selected three measures previously used in the literature for assessing the performance of the different algorithms: the average computational time of one solution, the reconstruction relative error ($RE = \sum_{i=1}^{n}(\beta_i - \hat{\beta}_i)^2 / \sum_{i=1}^{n} {\beta_i}^2$), and the area under the ROC curve (AUC) [21].

Figure 1 presents a boxplot of the computational time (in seconds) of all models and algorithms, showing that - besides the non-iterative Ridge solutions- the faster models are LASSO using LARS algorithm, as well as NN-SLASSO and NNG using AMNR for whatever reference estimator. As expected, the same models using AMNR were generally faster than when computed with the MNR algorithm.

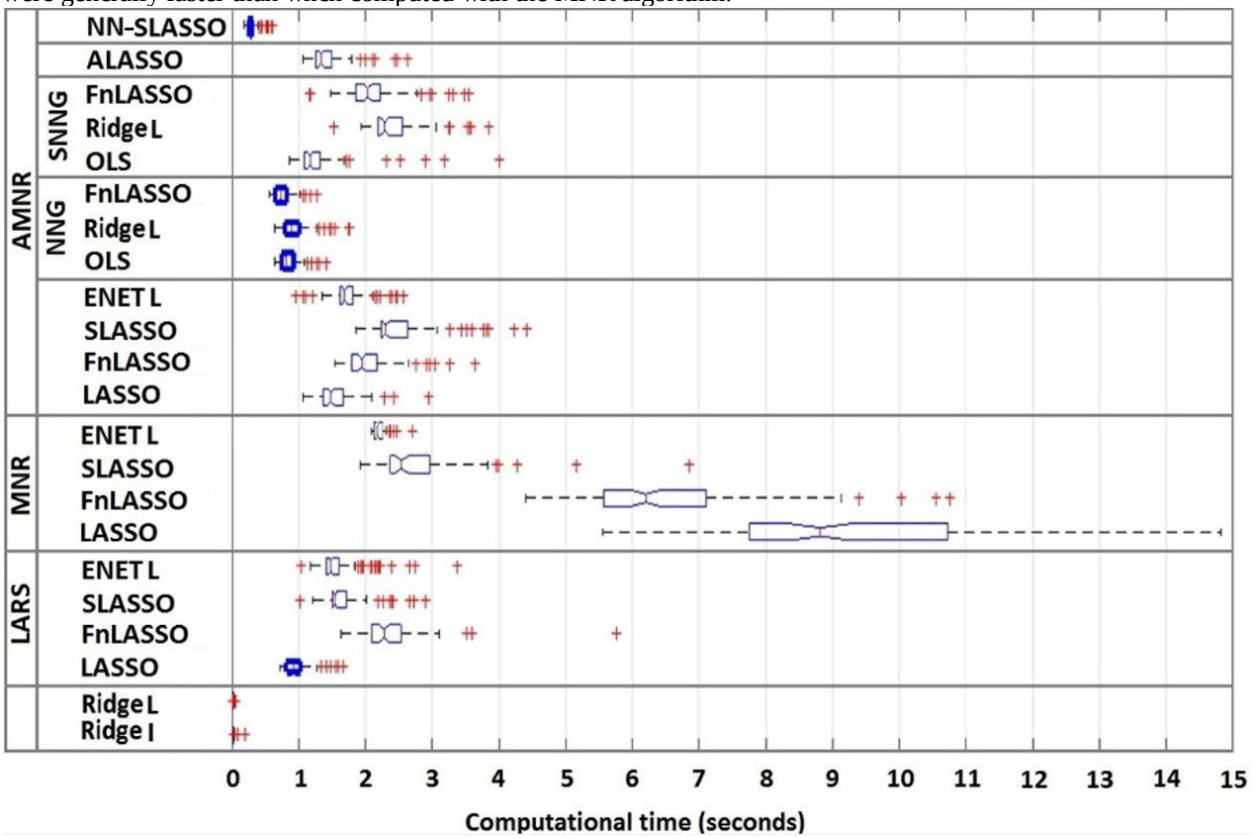

Figure 1 Boxplots of the time necessary for computing one solution in each combination of model and algorithm, from the simulations using p =200 and n = 100. The Ridge solutions were found using Tikhonov regularization. In the case of NNG and SNNG models, they are shown using three different reference solutions: OLS, Ridge I and FnLASSO.

Figure 2 shows the median of AUC, median Relative Error and median computational time across the 100 estimated solutions in the three cases of n/p ratios {0.05, 0.25, 0.5} and the mean across these three n/p ratios (black line). The methods with better behavior for any n/p relation were SLASSO (with LARS and AMNR algorithms), ENET L (with LARS and AMNR algorithms), SNNG (with FnLASSO as reference estimator) and the nonnegative version of SLASSO (NN-SLASSO). Together with a visual inspection of typical reconstructions, these results suggest that SLASSO, ENET L, SNNG and NN-SLASSO are the solutions that better reconstructed

the simulated bell and square regions. We found that all methods had more problems in reconstructing the isolated point source, although it is clear that this problem will not be largely reflected in the quantitative measures, as it is just one out of 200 estimated points.

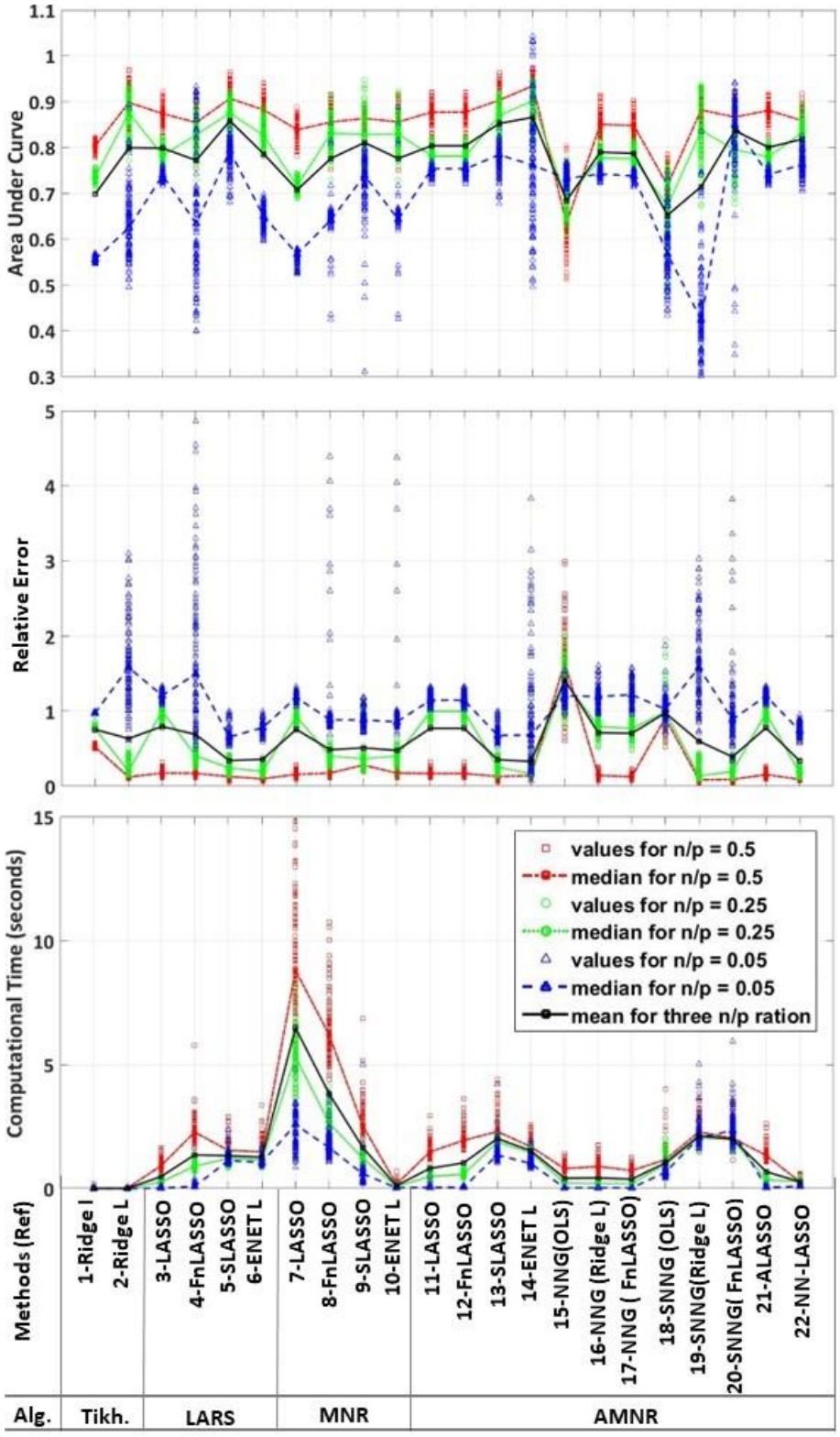

Figure 2: Medians of evaluation criteria (AUC, Relative Error and Computational Time) obtained from all 100 estimated solutions for the three different n/p ratios. The black line represents the mean value across all n/p ratios. The 22 methods in the x-axis correspond to combination of models and algorithms (Alg.) as named in the last row, where Tikh. refers to the Tikhonov algorithm

As a second validation study, we used a synthetic realistic EEG data, created from four different sets of simulated primary current density (PCD) distributions, all of them represented as three-dimensional Gaussian sources with amplitude of 10 nA/mm2 and width of 10 mm (spherical). Each dataset contains seven PCDs: a 'centroid' PCD with maximum located in a particular anatomical structure of a brain space of 3862 generators, and 6 others derived from this one by locating the maxima in each of the 6 closest neighbor generators (2 in each direction x, y, z). The maximum values of the simulated PCDs were located in 1) the cingulate region left (Cingulate), 2) occipital pole left (Occipital), 3) postcentral gyrus (Postcentral), and temporal gyrus right (Temporal) as shown in the first row of Figure 3. Talairach Coordinates [25] of the maximum value of each simulated PCD appear in Table 2. The EEG data was obtained by projecting these sources to the scalp using the Electric Lead Field computed for the brain space, using a three-spheres piece-wise homogeneous and isotropic model [23]. Finally, a low level of Gaussian noise was added in order to have an SNR of about 10 db.

	Coordinate				Coordinate				Coordinate				Coordinate		
Region: Cingulate	**x**	**y**	**z**	**Region: Postcentral**	**x**	**y**	**z**	**Region: Occipital**	**x**	**y**	**z**	**Region: Temporal**	**x**	**y**	**Z**
	-8	**48**	**5**		**20**	**-43**	**68**		**-22**	**-99**	**-2**		**41**	**-8**	**-37**
	6	48	5		13	-43	68		-29	-99	-2		34	-8	-37
	-1	48	5		27	-43	68		-15	-99	-2		34	13	-30
	-8	48	12		20	-43	75		-22	-99	5		41	-8	-30
	-8	48	-2		20	-43	61		-22	-99	-9		41	-8	-44
	-8	55	5		20	-36	68		-22	-92	-2		41	-1	-37
	-8	41	5		13	-43	61		-22	-92	-9		41	-15	-37

Table 2: Talairach coordinates of the maximum value of simulated four sets of solutions. The first row in each case shows the coordinates of the "centroid" simulated PCD (bold).

Figure 3 presents the estimated sources by the methods that showed the best reconstructions of the simulated 'centroid' PCDs in each region, both visually and according to the results in the previous simulation study. We also added the Ridge L solution, which is mathematically equivalent to a classical solution known as LORETA in the field of EEG source localization [22]. It can be seen that as expected, the Ridge L solutions are very smooth, while ENET L and SNNG methods (computed with AMNR) offered solutions that fluctuate between different degrees of sparsity/smoothness. Also, the use of sign constraints (allowed by AMNR) in the new inverse solutions SNNG and NN-SLASSO, led to sparser solutions than the unconstrained counterparts. SNNG solutions seem to be sparser versions of the reference solutions but without removing all ghost sources. The NN-SLASSO solutions are over-sparse but showing much less ghost sources as a convenient side effect. This solution also improves the localization of the main source with respect to ENET L, offering a very good localization even for the deepest simulated PCD (Temporal).

7. CONCLUSIONS

In this work, we gave a theoretical derivation of an algorithm based on the active-set strategy for solving Multiple Penalized Least Squares regression models. We showed that optimality conditions hold for the Adaptive LASSO model, which allows us to address many other different models based on multiple penalization that can be algebraically reduced to it. The main objective of proposing this type of algorithm is to offer an alternative to other classical approaches that critically depend on the correct estimation of one or more regularization parameters. We also carried out a preliminary exploration of the performance of the AMNR using simulated data from well-known ill-posed problems, including realistic simulations based on the EEG inverse problem.

Other active-set based algorithms have been proposed for an efficient estimation without losing accuracy, but usually targeting specific problems or models, such as the LARS [4] and the Fast Marginal Likelihood Maximization for Relevance Vector Machine (RVM, [29]). The former was derived mainly for the LASSO model, although subsequent developments have been able to adapt it to other similar models. The RVM is based in a Sparse Bayesian Model using a Gaussian prior with different weights (precisions) for each coefficient of the solution, which make it similar to an Adaptive Lasso. Both cases avoid the search for optimal values of regularization parameters using empirical information criteria, such as Akaike's or Generalized Cross Validation.

In the AMNR algorithm, similar to LARS, the selection of an optimal regularization parameter is replaced by iterating along a path of optimal solutions with respect to a bounding constraint which can be imposed or is fulfilled automatically by the stopping criteria based on the convergence of the solution. The step in which the algorithm converges define the level of sparsity of the solution attained. This is anyway an empirical solution to the problem and does not allow to make a deeper comparison and insight of the difference between solutions obtained with classical Newton-Raphson methods and their corresponding active set strategy. We believe that the AMNR strategy, which can be derived directly for more general MPLS models might help searching, in future studies, for an analytical relationship between the regularization parameter and the thresholding parameter in the equivalent constrained formulation of the problem.

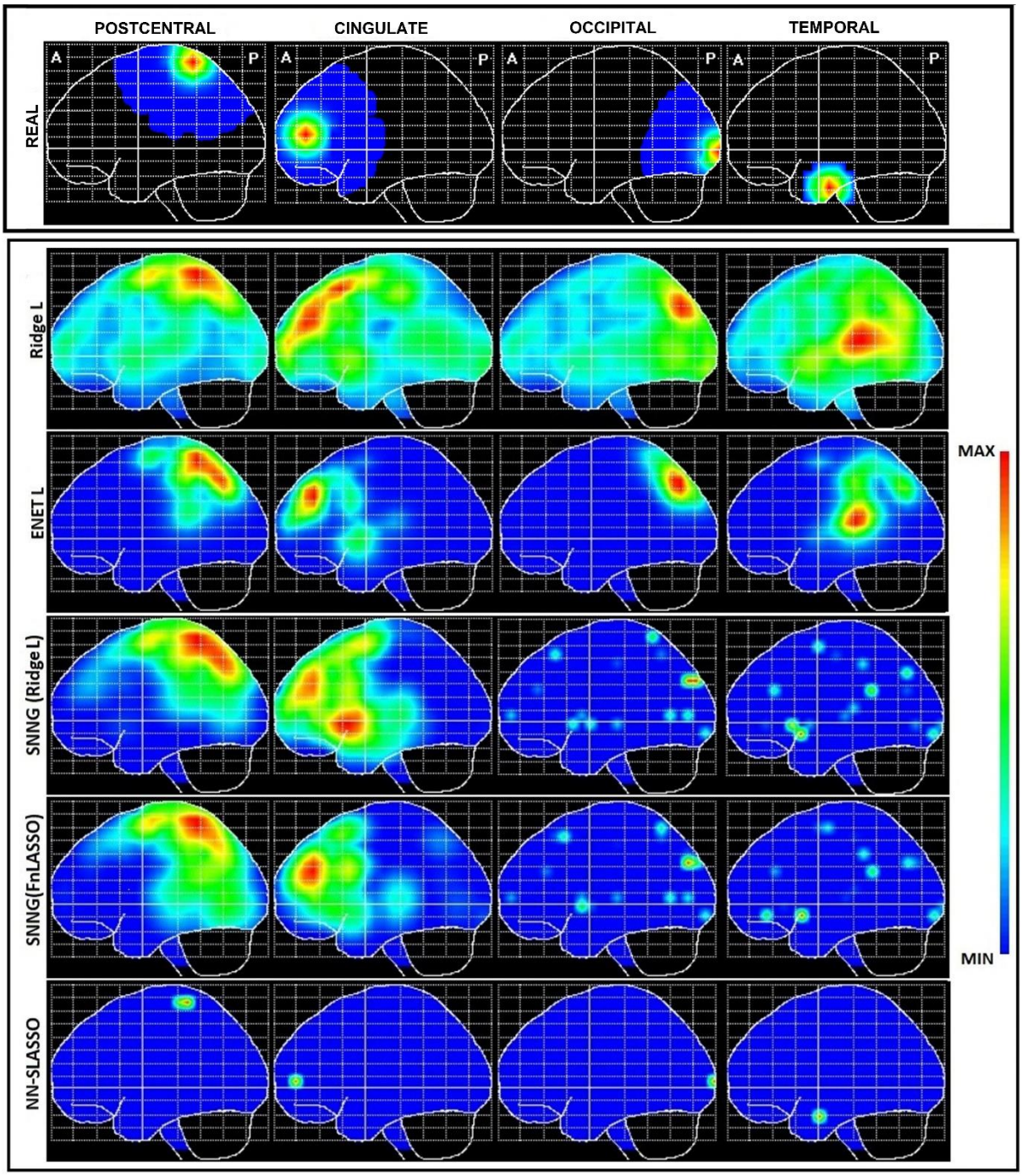

Figure 3: Maximum intensity projection in the sagittal plane of the four simulated 'centroid' PCDs (top row) and the corresponding estimated PCDs using five different methods (Ridge L; ENET L, SNNG with two different reference estimators and NN-SLASSO, the last four computed using the AMNR algorithm).

Given that the AMNR algorithm can be derived by following the same steps proposed by Efron in the derivation of the LARS algorithm, we can also expect that these algorithms are closely related. Analytically, we can say that the AMNR generalizes the idea of updating the solution by increments in the direction of the Newton-Raphson solution of the general penalized problem. This direction corresponds to the Ordinary Least Squares in LARS, for the LASSO type solution. In this sense, LARS can be said to be a particular case of the AMNR algorithm. Indeed, we found that in the case of low noise in the data, the AMNR solution for the LASSO model coincides with the LARS solution. However, for more general models, the Newton-Raphson direction takes into account not only the minimization of the likelihood, but the minimization of the whole cost function which includes the penalization terms. Therefore, what is claimed in LARS as following the direction that ensures equiangularity with respect to the residuals of all coefficients included in the active set, becomes in the AMNR in following the direction that ensures their equi-contribution to the fitting of both the data (residuals) and the penalization terms.

Our results also suggest that for popular MPLS models such as LASSO and ENET, the AMNR algorithm can reach the solutions faster than the Modified Newton-Raphson algorithms such as the classical Minorization-

Maximization and Local Quadratic which need to be done in the future. For instance, the influence of the initial estimates on the final solution must be studied, as well as the robustness of the solutions given by the algorithm when analyzing data with different levels of signal-to-noise ratio, which can be explored by varying the variance of the additive noise affecting the simulated data. In the context of the EEG inverse problem, it is also relevant to study how well the algorithm performs when estimating solutions with main activation at different depths in the brain, (i.e. at different distances from the sensors) and also for many different configurations of multiple sources in the brain. Our results also suggest that for popular MPLS models such as LASSO and ENET, the AMNR algorithm can reach the solutions faster than the Modified Newton-Raphson algorithms such as the classical Minorization-Maximization and Local Quadratic Approximation.

This makes it a promising alternative for handling data with very high dimensionality and the associated highly ill-posed problems. However, this study did not cover all experiments necessary to make a thorough characterization of the advantages and disadvantages of the AMNR as compared with other methods,

Finally, we think that a more complete theoretical study of the properties of this algorithm, as well as the derivation of versions to handle more complex models with several penalization terms, should be carried out to evaluate its usefulness in real-world applications of penalized regression to solve highly ill-posed inverse problems. This is particularly important for those problems where it is not possible to have a ground truth to compare the solution with, so different flexible penalization models that can adapt to the data at hand should be adopted. This is precisely the case of the Electroencephalography Inverse Problem.

RECEIVED: OCTOBER, 2022.
REVISED: AUGUST, 2023.

REFERENCES

[1] BERTSEKAS, D.P. (1995): **Nonlinear programming**, Athena Sci., no. January 1995, pp. x–646, 1997, doi: 10.1057/palgrave.jors.26800425.

[2] BERTSEKAS, D.P. (2003): Convex analysis and optimization, **Athena Sci.,** no. February. 2003.

[3] BREIMAN, L. (1995): Better Subset Regression Using the Nonnegative Garrote, **Technometrics,** 37,. 373–384, 1

[4] EFRON, B. T. HASTIE, I. JOHNSTONE, and R. TIBSHIRANI, "LEAST ANGLE REGRESSION (2004): **Ann. Stat.,** 32, 407–499.

[5] FAN, J. AND R. LI, (2001): Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties, **J. Am. Stat. Assoc.,** 96, 1348–1360,

[6] FRIEDMAN, J. T. HASTIE, H. HÖFLING, AND R. TIBSHIRANI, (2007): Pathwise coordinate optimization, **Ann. Appl. Stat.,** 1, 302–332,

[7] FU, W. J., (1998): Penalized Regressions: The Bridge Versus the Lasso," **J. Comput. Graph. Stat.,** 7, 397–416.

[8] GENÇ, M. (2021): **A new double-regularized regression using Liu and lasso regularization.** Springer Berlin Heidelberg.

[9] GREENWOOD, C. J. *et al.*, (2020): A comparison of penalised regression methods for informing the selection of predictive markers, **PLoS One,** 15, 1–14.

[10] HANSEN, C. (1998): **Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion,** Rank-Defic. Society for Industrial and Applied Mathematics, Philadelphia

[11] HEBIRI M. and S. VAN DE GEER, (2011): The smooth-lasso and other l1 + l2-penalized methods," **Electron. J. Stat.,** 5, 1184–1226.

[12] HOERL, A. E. and R. W. KENNARD, (1970): Ridge Regression: Applications to Nonorthogonal Problems,**Technometrics**, 12, 69–82.

[13] HUNTER, D.R. and R. LI, (2005): Variable selection using MM algorithms, **Ann. Stat.,** 33, 1617–1642.

[14] KIM, B., D. YU, and J.-H. WON, (2018): Comparative study of computational algorithms for the Lasso with high-dimensional , highly correlated data., **Appl Intell,** 48, 1933–1952.

[15] LAND, S. R. and J. H. FRIEDMAN, (1996): Variable Fusion: A new adaptive signal regression method, Stanford Pub, Standford.,

[16] LANGE, K. (2016): Convexity and Inequalities," in **MM Optimization Algorithms,** SIAM, Ed. Philadelphia, 2016, 21–48.

[17] LANGE, K. (2016): MM Optimization Algorithms, in **MM Optimization Algorithms,** Society fo., Philadelphia, 2016.

[19] LIN, C. J. (2007): Projected gradient methods for nonnegative matrix factorization, **Neural Comput.**, 19, 2756–2779.

[18] MARTÍNEZ-MONTES, E. J. M. SÁNCHEZ-BORNOT, and P. A. VALDÉS-SOSA, (2008): Penalized

PARAFAC analysis of spontaneous EEG recordings, **Stat. Sin.,** 18, 1449–1464.
[20] MØRUP, M. K. H. MADSEN, and L. K. HANSEN, (2008): Approximate L0 constrained non-negative matrix and tensor factorization," *Proc. - IEEE* **Int. Symp. Circuits Syst.,** 1328–1331,
[21] OBUCHOWSKI, N. A. And J. A. BULLEN, (2018): Receiver operating characteristic (ROC) curves: Review of methods with applications in diagnostic medicine, **Phys. Med. Biol.,** 63, 2018, doi: 10.1088/1361-6560/aab4b1.
[22] PASCUAL-MARQUI, R.D., C. M. MICHEL, and D. LEHMANN, (1994): Low resolution electromagnetic tomography: a new method for localizing electrical activity in the brain," **Int. J. Psychophysiol.,** vol. 18, no. 1, pp. 49–65, 1994, doi: 10.1016/0167-8760(84)90014-X.
[23] RIERA, J.J. AND M. E. FUENTES, (1998): Electric Lead Field for a Piecewise Homogeneous Volume Conductor Model of the Head, **IEEE Trans. Biomed. Eng.,** 45, 746–753.
[24] SÁNCHEZ-BORNOT, J. M. , E. MARTÍNEZ-MONTES, A. LAGE-CASTELLANOS, M. VEGA-HERNÁNDEZ, and P. A. VALDÉS-SOSA, (2008): Uncovering sparse brain effective connectivity: A voxel-based approach using penalized regression, **Stat. Sin.,** 18, 1501–1518. ,
[25] TALAIRACH, J. M. RAYPORT, and P. TOURNOUX, (1988): **Co-planar stereotaxic atlas of the human brain 3-dimensional.** Theme Medical Publishers, New York.
[26] TIBSHIRANI, R. (1996): Regression Shrinkage and Selection via the Lasso," **Journal of the Royal Statistical Society B,** 58, . 267–288.
[27] TIBSHIRANI, R., M. SAUNDERS, S. ROSSET, J. ZHU, and K. KNIGHT, (2005): Sparsity and smoothness via the fused lasso, **J. R. Stat. Soc. Ser. B Stat. Methodol.,** 67, . 91–108.
[28] TIKHONOV A. N., A. V. GONCHARSKY, V. V. STEPANOV, and A. G. YAGOLA, (1995): **Numerical Methods for the Solution of Ill-Posed Problems.** doi: 10.1007/978-94-015-8480-7.
[29] TIPPING M. E. and A. C. FAUL, "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models," **Proc. Ninth Int. Work. Artif. Intell. Stat.**, no. August, 2003, [Online]. Available: http://www.miketipping.com/papers.htm
[30] VALDÉS-SOSA, P. A., J. M. BORNOT-SÁNCHEZ, M. VEGA-HERNÁNDEZ, L. MELIE-GARCÍA, A. LAGE-CASTELLANOS, and E. CANALES-RODRÍGUEZ, (2006): **Granger Causality on Spatial Manifolds: Applications to Neuroimaging.** doi: 10.1002/9783527609970.ch18.
[31] VEGA-HERNÁNDEZ, M., E. MARTÍNEZ-MONTES, J. M. SANCHEZ-BORNOT, A. LAGE-CASTELLANOS, and P. A. VALDÉS-SOSA, (2008): Penalized least squares methods for solving the EEG inverse problem," **Stat. Sin.,** 18, . 4, . 1535–1551,
[32] VORONIN, S. (2012): **Regularization of Linear Systems with Sparsity Constraints with Applications to Large Scale Inverse Problems**.
[33] WANG, Z. (2022) MM for penalized estimation, **Test,** 31,. 54–75.
[34] ZOU H. and T. HASTIE, (2005): Regularization and variable selection via the elastic-net, **J. R. Stat. Soc.,** 67, . 301–320,
[35] ZOU, H. (2006): The adaptive lasso and its oracle properties, **J. Am. Stat. Assoc.,** 101, . 1418–1429.