A Review on the Adaptive-Ridge Algorithm with several extensions

Rémy Abergel, Olivier Bouaziz, Grégory Nuel

▶ **To cite this version:**

Rémy Abergel, Olivier Bouaziz, Grégory Nuel. A Review on the Adaptive-Ridge Algorithm with several extensions. 2023. hal-04051929

HAL Id: hal-04051929
https://hal.science/hal-04051929

Preprint submitted on 30 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Review on the Adaptive-Ridge Algorithm with several extensions

Rémy Abergel and Olivier Bouaziz

Université Paris Cité, CNRS, MAP5, F75006 Paris, France
e-mail: Remy.Abergel@u-paris.fr; Olivier.Bouaziz@u-paris.fr

Grégory Nuel

LPSM, CNRS 8001, Sorbonne University,
Campus Pierre et Marie Curie, 4 place Jussieu, 75005 Paris - France
e-mail: Nuel@math.cnrs.fr

Abstract: The Adaptive Ridge Algorithm is an iterative algorithm designed for variable selection. It is also known under the denomination of Iteratively Reweighted Least-Squares Algorithm in the communities of Compressed Sensing and Sparse Signals Recovery. Besides, it can also be interpreted as an optimization algorithm dedicated to the minimization of possibly nonconvex ℓ^q penalized energies (with $0 < q < 2$). In the literature, this algorithm can be derived using various mathematical approaches, namely Half Quadratic Minimization, Majorization-Minimization, Alternating Minimization or Local Approximations. In this work, we will show how the Adaptive Ridge Algorithm can be simply derived and analyzed from a single equation, corresponding to a variational reformulation of the ℓ^q penalty. We will describe in detail how the Adaptive Ridge Algorithm can be numerically implemented and we will perform a thorough experimental study of its parameters. We will also show how the variational formulation of the ℓ^q penalty combined with modern duality principles can be used to design an interesting variant of the Adaptive Ridge Algorithm dedicated to the minimization of quadratic functions over (nonconvex) ℓ^q balls.

Keywords and phrases: adaptive ridge, iteratively reweighted algorithms, majorize-minimize algorithms, variable selection, sparse minimization.

Contents

1 Introduction 2
2 Iteratively reweighted algorithms for ℓ^q penalized selection 5
2.1 Variational formulation of the ℓ^q penalty 5
2.2 The majorize-minimize strategy 6
2.3 The alternating minimization strategy 10
2.4 The local approximation based strategy 11
2.5 Links with other algorithms 14
3 Adaptive-Ridge and square-log penalized selection 15
4 Several implementations of the Adaptive-Ridge Algorithm 17
4.1 System inversion based implementation 17
4.2 Conjugate-Gradient based implementation 19
4.3 Numerical experiments 22
5 Extension to ℓ^q constrained selection 33
5.1 Principles and motivations 33
5.2 An alternating minimization scheme and its properties 34
5.3 Numerical implementation 37
5.4 Experiments 40

6 Conclusion 46
Appendix 48
References 52

1. Introduction

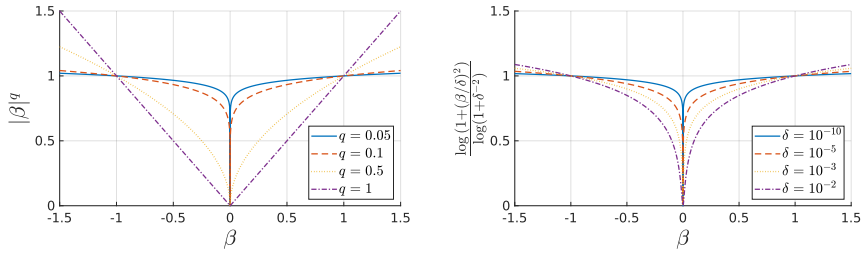
In statistical applications, variable selection is often a desirable goal. It allows to identify relevant predictors and at the same time may enhance the prediction performance of the model. There is a large literature on this topic that can be roughly separated in two main approaches: stepwise selection and penalized models. The first one encompasses the best subset selection method and the forward-backward stepwise selection methods. They either explore exhaustively all possible subsets (defined by all possible combinations of covariates) or find a good path through these subsets by sequentially adding or removing variables into the model. A criterion that can be based on a statistical test or on criterions such as AIC or BIC is used at each step to compare the models among themselves (see [26] for a review on these methods). On the other hand, penalized models estimate the regression coefficients by constraining their size. They were popularized with the ridge estimator [27], the LASSO estimator [46] and its extensions such as the adaptive lasso [54] or the elastic-net [55]. In this paper, we provide a review on one of these methods, the adaptive-ridge proposed in [20], which is an iterative weighted penalized algorithm. In the following, we start by rewriting the variable selection problem as a ℓ^0 penalized criterion. The aim of this work is to make the connection between the adaptive-ridge algorithm and this variable selection problem.

Given a positive integer p, a data-fidelity (usually a log-likelihood or a least-squares criterion) function $C : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ such as $\mathrm{dom}(C) := \{\beta \in \mathbb{R}^p,\ C(\beta) < +\infty\} \neq \emptyset$, and a regularity (or penalty) parameter $\lambda > 0$, most variable selection models boil down to computing

$$\overline{\beta} \in \underset{\beta \in \mathbb{R}^p}{\mathrm{argmin}} \ \ (\mathcal{E}_\lambda(\beta) := C(\beta) + \lambda\, \mathcal{L}_0(\beta)) \tag{1}$$

where $\mathcal{L}_0(\beta)$ denotes the number of non-zero entries of β, that is, $\mathcal{L}_0(\beta) := \#\{j \in \{1, 2, \ldots, p\},\ \beta_j \neq 0\}$, denoting by $\#$ the cardinality. Notice that the argmin in (1) can be restricted to $\mathrm{dom}(C)$ without changing the set of its minimizers. Allowing C to take the value $+\infty$ over $\mathbb{R}^p$ is a simple and common way in optimization to integrate constraints (for instance nonnegativity) in the optimization problem (1). In the general case, the nonconvexity and the nonsmoothness of the $\mathcal{L}_0$ penalty make the optimization problem (1) difficult to handle [35, 44]. A common alternative to (1) consists in replacing the $\mathcal{L}_0$ penalty by smoother approximations [5, 40, 17, 19, 54, 44]. For instance, given $q > 0$, one can consider the ℓ^q approximation of the $\mathcal{L}_0$ penalty (see e.g. [19] and Fig. 1 (a)),

$$\mathcal{L}_0(\beta) \approx \|\beta\|_q^q := \sum_{j=1}^{p} |\beta_j|^q\,, \tag{2}$$



(a) ℓ^q penalty (b) square-log penalty

FIG 1. ***Smooth approximations of the $\mathcal{L}_0$-penalty.*** *We display the graphs of several approximations of the $\mathcal{L}_0$ penalty in the monodimensional setting. The graph of the ℓ^q penalty, that is the function $\beta \mapsto |\beta|^q$, is displayed in (a) for several positive values of q. The graph of the log-square penalty, that is the function $\beta \mapsto \log\left(1+(\beta/\delta)^2\right)/\log\left(1+\delta^{-2}\right)$, is displayed in (b) for several positive values of δ. The $\mathcal{L}_0$ penalty (not represented here) satisfies $\mathcal{L}_0(0) = 0$ and $\mathcal{L}_0(\beta) = 1$ for any $\beta \neq 0$ and is therefore discontinuous at 0. Both the ℓ^q and the log-square penalties are continuous over $\mathbb{R}$ and can be viewed as smooth approxmimations of the $\mathcal{L}_0$ penalty that get more accurate as their parameters $q > 0$ and $\delta > 0$ decrease. However, taking $q = 0$ would yield $|\beta|^q = 1$ for any $\beta \in \mathbb{R}$, which is not a satisfactory approximation of $\mathcal{L}_0(\beta)$ anymore. Similarly, the setting $\delta = 0$ is not allowed for the square-log penalty. Therefore, both parameters must be kept positive in practice.*

leading to the ℓ^q-penalized problem of finding

$$\widetilde{\beta} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \quad \left(E_{\lambda,q}(\beta) := C(\beta) + \lambda \|\beta\|_q^q\right) . \tag{3}$$

Alternatively, given $\delta > 0$, one can consider the *square-log* approximation of the $\mathcal{L}_0$ penalty (see [24, 23] and Fig. 1 (b)),

$$\mathcal{L}_0(\beta) \approx \sum_{j=1}^{p} \frac{\log\left(1+(\beta_j/\delta)^2\right)}{\log\left(1+\delta^{-2}\right)} \tag{4}$$

and look for a solution of the square-log penalized problem

$$\widetilde{\beta} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \quad \left(F_{\lambda,\delta}(\beta) := C(\beta) + \lambda \sum_{j=1}^{p} \frac{\log\left(1+(\beta_j/\delta)^2\right)}{\log\left(1+\delta^{-2}\right)}\right) . \tag{5}$$

Both the ℓ^q penalized energy $E_{\lambda,q}$ and the square-log penalized energy $F_{\lambda,\delta}$ can be viewed as smooth approximations of the targeted $\mathcal{L}_0$ penalized energy $\mathcal{E}_\lambda$. As illustrated in Fig. 1, the smaller the parameters q and δ are, the better accurate those approximations are. Notice that, in general, $F_{\lambda,\delta}$ is nonconvex, and so is $E_{\lambda,q}$ when $q < 1$. Thus, handling (3) and (5) remains a difficult task. Several algorithms designed to approach some (at least local) minimizers of energies like $E_{\lambda,q}$ or $F_{\lambda,\delta}$ can be found in the literature. Those come from various communities, such as statistics [17, 28, 56, 42, 20], sparse signal recovery and compressed sensing [12, 5, 34, 41], optimization [36, 8, 39, 29] and image

processing [2, 9, 37, 3, 22, 21]. The *Adaptive-Ridge (AR) Algorithm* is one of those which was proposed in [20] and comes from a long line of optimization algorithms. Given $w^{(0)} \in (\mathbb{R}_+^*)^p$, $\delta > 0$, $q \in [0, 2)$ and $\gamma > 0$, the AR scheme, to which we may refer below as $\mathrm{AR}_{\lambda,q}^{\delta,\gamma}$, consists in iterating, for $k \geq 0$,

$$\begin{cases} \beta^{(k+1)} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; C(\beta) + \dfrac{\lambda}{2} \displaystyle\sum_{j=1}^{p} w_j^{(k)} \, \beta_j^2 & \text{(6a)} \\ w^{(k+1)} = \left(|\beta^{(k+1)}|^\gamma + \delta^\gamma \right)^{\frac{q-2}{\gamma}}, & \text{(6b)} \end{cases}$$

having set $|z|^a = (|z_1|^a, |z_2|^a, \ldots, |z_p|^a)$ for all $a \in \mathbb{R}$ and all $z \in \mathbb{R}^p$.

Prior to [20], many numerical schemes closely related to (6) have been proposed in the literature. They can be described using different mathematical approaches, including locally quadratic or linear approximations [17, 56, 42], majorize-minimize schemes [28, 56], iteratively reweighted algorithms [10, 12, 39], half-quadratic regularization [37, 36, 8]. Those scheme and their properties remain a topical and fruitful research subject [49, 31, 41, 23] covering a wide range of applications beyond the single problem of variable selection.

In this work, we will perform a review of the Adaptive-Ridge Algorithm, from its mathematical design to its numerical implementation, including a thorough experimental analysis of its parameters and its general behavior. We will enlighten the links between this algorithm and other closely related schemes and we will propose several extensions. This work is organized as follows. In Section 2, we will firstly provide a variational formulation of the ℓ^q penalty (2) and use it to build up some iteratively reweighting schemes designed for solving (3). We will show how the Adaptive-Ridge Algorithm can be identified to a particular instance of those schemes and we will demonstrate, using noticeably simple arguments, its ability to decrease the ℓ^q penalized cost function $E_{\lambda,q}$ involved in (3) along the scheme iterations. We will also show how other mathematical constructions of the scheme based on majorize-minimize or alternating minimization strategies can be derived naturally from the above mentioned variational formulation of the ℓ^q penalty. Links with other existing instances or closely related variant of the Adaptive-Ridge that can be found in the literature will be also presented. Section 3 will be dedicated to the particular setting $q = 0$ in (6) where the Adaptive-Ridge Algorithm will be interpreted as a locally linear approximation scheme dedicated to the minimization of the square-log penalized energy involved in (5). In Section 4, we will focus on two numerical implementations of the Adaptive-Ridge Algorithm, a particular care will be taken to ensure robustness of the proposed implementations with respect to numerical errors. Then, we will present a thorough experimental study of the Adaptive-Ridge Algorithm over synthetic datasets. Finally, in Section 5, we will propose and study a variant of the Adaptive-Ridge Algorithm dedicated to the minimization of a quadratic cost over a ℓ^q ball constraint set.

2. Iteratively reweighted algorithms for ℓ^q penalized selection

2.1. Variational formulation of the ℓ^q penalty

The mathematical construction of iteratively reweighted algorithms dedicated to ℓ^q penalized optimization problems like (3) can be explained in various ways, leading to different interpretations of very closely related numerical schemes. We found that a quite simple and elegant description of such numerical scheme could be derived using a variational reformulation of the ℓ^q penalty described in Lemma 1 in dimension one and extended to $\mathbb{R}^p$ in Proposition 1.

Lemma 1. *For all $\beta \in \mathbb{R}$, for all $q > 0$ and for all $\nu > q$, we have*

$$|\beta|^q = \inf_{\eta \in \mathbb{R}_+^*} \left(\ell_q^\nu(\beta, \eta) := \frac{q}{\nu} \cdot \frac{|\beta|^\nu}{\eta} + \frac{\nu - q}{\nu} \cdot \eta^{\frac{q}{\nu - q}} \right), \tag{7}$$

where $\mathbb{R}_+^*$ denotes the set $(0, +\infty)$. Besides, when $\beta \neq 0$, the infimum (7) is attained at $\eta = |\beta|^{\nu - q}$.

Proof. Let $\beta \in \mathbb{R}$, $q > 0$ and $\nu > q$. When $\beta = 0$, both sides of (7) are equal to 0 (note that the infimum is not attained). When $\beta \neq 0$, noting $f = \eta \mapsto \ell_q^\nu(\beta, \eta)$, one easily checks that f assumes $\eta^* = |\beta|^{\nu - q}$ as unique critical point over $\mathbb{R}_+^*$ and that $f'(\eta) \geq 0$ if and only if $\eta \geq \eta^*$. Therefore, f achieves its minimum over $\mathbb{R}_+^*$ at η^* and we can finally check that $f(\eta^*) = |\beta|^q (\frac{q}{\nu} + \frac{\nu - q}{\nu}) = |\beta|^q$. □

Proposition 1 (variational reformulation of the ℓ^q penalty term)**.** *For all $\beta = (\beta_1, \beta_2, \ldots, \beta_p) \in \mathbb{R}^p$, for all $q > 0$ and for all $\nu > q$, we have*

$$\|\beta\|^q = \inf_{\eta = (\eta_1, \eta_2, \ldots, \eta_p) \in \left(\mathbb{R}_+^*\right)^p} \left(\mathcal{L}_q^\nu(\beta, \eta) := \sum_{j=1}^p \frac{q}{\nu} \cdot \frac{|\beta_j|^\nu}{\eta_j} + \frac{\nu - q}{\nu} \cdot \eta_j^{\frac{q}{\nu - q}} \right), \tag{8}$$

and, when $\beta \in (\mathbb{R}^*)^p$, the infimum is attained at $\eta = |\beta|^{\nu - q}$.

Proof. Let $\beta = (\beta_1, \beta_2, \ldots, \beta_p) \in \mathbb{R}^p$, $q > 0$, and $\nu > q$. By additive separability of $\mathcal{L}_q^\nu(\beta, \eta) = \sum_{j=1}^p \ell_q^\nu(\beta_j, \eta_j)$ with respect to $(\eta_1, \eta_2, \ldots, \eta_p)$, we can interchange infimum and sum in the right-hand side of (8). Thus, using Lemma 1, we get

$$\inf_{\eta = (\eta_1, \eta_2, \ldots, \eta_p) \in \left(\mathbb{R}_+^*\right)^p} \mathcal{L}_q^\nu(\beta, \eta) = \sum_{j=1}^p \inf_{\eta_j \in \mathbb{R}_+^*} \ell_q^\nu(\beta_j, \eta_j) = \sum_{j=1}^p |\beta_j|^q = \|\beta\|_q^q,$$

as announced. □

The variational formulation (7) with the setting $\nu = 2$ and $q \in (0, 2)$ can be found in [8] where ℓ_q^2 is referred as *half-quadratic* since it is quadratic with respect to β but not with respect to η. The relation (7) can also be found in [33] for the more particular setting $q = 1$ and $\nu = 2$ but in a slightly disputable form since the infimum (7) is taken over $\mathbb{R}_+$ instead of $\mathbb{R}_+^*$ in [33, Section 5.4] although the function $\eta \mapsto \ell_q^\nu(\beta, \eta)$ involved in (7) is not defined for $\eta = 0$.

A closely related (but non equivalent) result as (8) with the setting $\nu = 2$ and $q \in (0, 2)$ can be found in [30, Lemma 3.1] and seems also, as formulated, slightly contestable due to an infimum taken over $\mathbb{R}_+^p$ instead of $(\mathbb{R}_+^*)^p$. Notice that the methodology described in [36, Section II], which was initiated in [21] and applies to more general penalty terms than the ℓ^q penalty considered here, can be used to derive (7) and (8) in the case $\nu = 2$ and $q \in (0, 2)$. Besides, a very interesting link between the variational formulation (8) and the Legendre-Fenchel conjugate was pointed out in [36]. Another interesting methodology relying on the notion of *Legendre pairs* for obtaining similar (but non equivalent) variational (in fact half-quadratic) formulations of a large class of penalty terms was initiated in [22] and further developed for image processing applications in [37, 9, 29, 2]. We could not find explicit references to (7) and (8) for $\nu \neq 2$ (in particular, for $\nu = 1$) in the literature but we believe that those relations are probably well known as they can be used to derive some classical iteratively reweighted algorithms in a very natural way, as we shall see now.

2.2. The majorize-minimize strategy

In all the following and unless explicitly mentioned otherwise, ν and q denote two real numbers such as $\nu > q > 0$. A straightforward consequence of Proposition 1 is that, for any $\eta \in \left(\mathbb{R}_+^*\right)^p$, we have the upper-bound

$$\forall \beta \in \mathbb{R}^p\,, \quad E_{\lambda,q}(\beta) \leq \mathcal{S}^{\nu}_{\lambda,q}(\beta, \eta) := C(\beta) + \lambda \mathcal{L}^{\nu}_q(\beta, \eta)\,. \tag{9}$$

Besides, when $\beta \in (\mathbb{R}^*)^p$ (and only in this case), this large inequality becomes an equality for $\eta = |\beta|^{\nu-q}$, i.e.,

$$\forall \beta \in (\mathbb{R}^*)^p\,, \quad E_{\lambda,q}(\beta) = \mathcal{S}^{\nu}_{\lambda,q}(\beta, |\beta|^{\nu-q})\,. \tag{10}$$

Therefore, if we consider an initial guess $\beta^{(0)} \in (\mathbb{R}^*)^p$ for the optimization problem (3), then $\mathcal{S}_0 : \beta \mapsto \mathcal{S}^{\nu}_{\lambda,q}(\beta, |\beta^{(0)}|^{\nu-q})$ bounds $E^{\nu}_{\lambda,q}$ from above over $\mathbb{R}^p$ and both functions coincide at point $\beta = \beta^{(0)}$. More precisely, we have

$$\forall \beta \in \mathbb{R}^p\,, \quad E_{\lambda,q}(\beta) \leq \mathcal{S}_0(\beta) \quad \text{and} \quad E_{\lambda,q}(\beta^{(0)}) = \mathcal{S}_0(\beta^{(0)})\,. \tag{11}$$

Now, assuming that we are able to compute a minimizer of $\mathcal{S}_0$, noting $\beta^{(1)}$ such a minimizer, we obtain

$$E_{\lambda,q}(\beta^{(1)}) \leq \mathcal{S}_0(\beta^{(1)}) \leq \mathcal{S}_0(\beta^{(0)}) = E_{\lambda,q}(\beta^{(0)})\,. \tag{12}$$

Assuming that $\beta_j^{(1)} \neq 0$ for all j, we can repeat the process replacing $\mathcal{S}_0$ by $\mathcal{S}_1 : \beta \mapsto \mathcal{S}^{\nu}_{\lambda,q}(\beta, |\beta^{(1)}|^{\nu-q})$, look for a minimizer $\beta^{(2)}$ of $\mathcal{S}_1$, and so on. This yields the so-called *majorize-minimize* iterations, which amounts to computing

$$\forall k \in \mathbb{N}\,, \quad \beta^{(k+1)} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \mathcal{S}^{\nu}_{\lambda,q}(\beta, |\beta^{(k)}|^{\nu-q})\,, \tag{13}$$

or, equivalently, to setting $\eta^{(0)} = |\beta^{(0)}|^{\nu-q}$ and iterating

$$\begin{cases} \beta^{(k+1)} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}\ \mathcal{S}^{\nu}_{\lambda,q}(\beta, \eta^{(k)}) & \text{(14a)} \\ \eta^{(k+1)} = |\beta^{(k+1)}|^{\nu-q}\,. & \text{(14b)} \end{cases}$$

Finally, removing from $\mathcal{S}^{\nu}_{\lambda,q}$ the constant terms that do not change the value of the argmin (14a), we obtain

$$\begin{cases} \beta^{(k+1)} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}\ C(\beta) + \dfrac{\lambda q}{\nu}\displaystyle\sum_{j=1}^{p} \frac{|\beta_j|^{\nu}}{\eta_j^{(k)}} & \text{(15a)} \\ \eta^{(k+1)} = |\beta^{(k+1)}|^{\nu-q}\,. & \text{(15b)} \end{cases}$$

It is important to note that the majorize-minimize scheme (15) is valid as long as its iterates $(\eta^{(k)})_{k\geq 0}$ remain in $(\mathbb{R}_+^*)^p$, or equivalently, as long as its iterates $(\beta^{(k)})_{k\geq 0}$ remain in $(\mathbb{R}^*)^p$. Under this assumption, the scheme (15) guarantees the monotonic decrease of the energy sequence $(E_{\lambda,q}(\beta^{(k)}))_{k\geq 0}$, as stated in Proposition 2.

Proposition 2. *Assuming that $\eta^{(0)} = |\beta^{(0)}|^{\nu-q} \in (\mathbb{R}_+^*)^p$ and that the iterates $(\beta^{(k)})_{k\geq 0}$ generated using (15) lie in $(\mathbb{R}^*)^p$, we have*

$$\forall k \in \mathbb{N}\,, \quad E_{\lambda,q}(\beta^{(k+1)}) \leq E_{\lambda,q}(\beta^{(k)})\,. \tag{16}$$

Consequently, if C is bounded from below, then, the sequence $\left(E_{\lambda,q}(\beta^{(k)})\right)_{k\geq 0}$ is also bounded from below and, thus, converges.

Proof. Let $k \in \mathbb{N}$. By construction in (15), we have $\eta^{(k)} = |\beta^{(k)}|^{\nu-q}$ and, by assumption, we have $\eta^{(k)} \in (\mathbb{R}_+^*)^p$. Then, thanks to (9) and (15a), we have $E_{\lambda,q}(\beta^{(k+1)}) \leq \mathcal{S}^{\nu}_{\lambda,q}(\beta^{(k+1)}, \eta^{(k)}) \leq \mathcal{S}^{\nu}_{\lambda,q}(\beta^{(k)}, \eta^{(k)}) = E_{\lambda,q}(\beta^{(k)})$, the right-hand side equality coming from (10). □

Interestingly enough, one can see that, when $\nu = 2$, the scheme (15) is closely related to the Adaptive-Ridge scheme (6), as we detail in Remark 1.

Remark 1 (link between (15) and the $\mathrm{AR}^{\delta,\gamma}_{\lambda,q}$ scheme (6))**.** *Taking $\nu = 2$ in (15) and noting $w_j^{(k)} = \frac{1}{\eta_j^{(k)}}$, we obtain*

$$\begin{cases} \beta^{(k+1)} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}\ C(\beta) + \dfrac{\lambda q}{2}\displaystyle\sum_{j=1}^{p} w_j^{(k)}\beta_j^2 & \text{(17a)} \\ w_j^{(k+1)} = |\beta_j^{(k+1)}|^{q-2} \quad \textit{for all } j \in \{1, 2, \ldots, p\}\,, & \text{(17b)} \end{cases}$$

which corresponds to the $AR^{\delta,\gamma}_{\lambda',q}$ scheme (6) provided that we set $\delta = 0$, $\gamma \neq 0$ and $\lambda' = \lambda q$. Since we explained that (15) (and thus (17)) is suited to the minimization of $E_{\lambda,q}$, we can see that the $AR^{\delta,\gamma}_{\lambda,q}$ scheme described in [20] is in fact

suited to the minimization of $E_{\frac{\lambda}{q},q}$ (instead of that of $E_{\lambda,q}$, as was announced in [20]).

Despite the nice energy decrease property (16) provided by Scheme (15), this scheme is only valid when all the iterates $(\beta^{(k)})_{k\geq 0}$ remain in $(\mathbb{R}^*)^p$, which is not guaranteed. In fact, if we keep in mind that the ℓ^q penalty is used in (3) to promote sparsity, we precisely expect a minimizer of $E_{\lambda,q}$ to assume zero entries. Consequently, it is likely that, at some point k_0 of the iteration process, we will obtain an iterate $\beta^{(k_0+1)}$ with zero entries so that (15b) yields $\eta^{(k_0+1)} \notin (\mathbb{R}_+^*)^p$ and Scheme (15) cannot be iterated for $k \geq k_0$ due to the presence of indefinite terms $\frac{|\beta_j|^\nu}{0}$ in (15a). A way to tackle this issue and keep iterating Scheme (15) when $\beta_j^{(k_0)} = 0$ consists in imposing $\beta_j^{(k)} = 0$ for the next iterations $k \geq k_0$. This modification of the scheme can be formally stated by replacing the terms $\frac{|\beta_j|^\nu}{\eta_j}$ in (15a) by $r(|\beta_j|^\nu, \eta_j)$, noting $r : \mathbb{R}^2 \to \mathbb{R} \cup \{+\infty\}$ the function defined by

$$\forall (x,y) \in \mathbb{R}^2\,, \quad r(x,y) = \begin{cases} 0 & \text{if } x = y = 0 \\ +\infty & \text{if } x \neq 0 \text{ and } y = 0 \\ \frac{x}{y} & \text{otherwise.} \end{cases} \tag{18}$$

This yields the iterations

$$\begin{cases} \beta^{(k+1)} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}\ C(\beta) + \dfrac{\lambda q}{\nu} \displaystyle\sum_{j=1}^{p} r(|\beta_j|^\nu, \eta_j^{(k)}) & \text{(19a)} \\ \eta^{(k+1)} = |\beta^{(k+1)}|^{\nu-q}\,. & \text{(19b)} \end{cases}$$

One can see that, when $\beta_j^{(k)} = 0$, we have $\eta_j^{(k)} = 0$ and the only way to avoid the sum taking the value $+\infty$ in (19a) is to impose the constraint $\beta_j = 0$ in (19a). Therefore, when (19) generates an iterate with vanishing entries, the same entries will remain equal to zero in all later iterations. As pointed out in [12], the persistence of vanishing entries may be problematic. For instance, assuming that (3) admits a unique solution β^*, if one iteration of (19) produces $\beta_j^{(k)} = 0$ while $\beta_j^* \neq 0$, then (19) has no longer chance to converge towards β^*. Despite this limitation, (19) remains interesting because, of Proposition 3.

Proposition 3. *When C is finite and additively separable over $\mathbb{R}^p$, then, the iterates $(\beta^{(k)})_{k\geq 0}$ generated using (19) satisfy the energy decrease property (16).*

Proof. The proof is given in Appendix A. □

Another way to tackle the issue of vanishing iterates in the majorize-minimize scheme (15) consists in imposing nonzero entries for the $\eta^{(k)}$ iterates. This can be done by introducing a numerical parameter $\delta > 0$, and replacing for instance (15b) by

$$\eta^{(k+1)} = \left(|\beta^{(k)}|^\nu + \delta^\nu\right)^{\frac{\nu-q}{\nu}}\,, \tag{20}$$

leading to the iterations

$$\begin{cases} \beta^{(k+1)} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}\ C(\beta) + \dfrac{\lambda q}{\nu}\displaystyle\sum_{j=1}^{p} \dfrac{|\beta_j|^\nu}{\eta_j^{(k)}} & \text{(21a)} \\ \eta^{(k+1)} = \left(|\beta^{(k+1)}|^\nu + \delta^\nu\right)^{\frac{\nu-q}{\nu}} . & \text{(21b)} \end{cases}$$

Thanks to the presence of the positive parameter δ in (21b), the entries of the $\eta^{(k)}$ iterates cannot vanish anymore, so that we can safely perform the iterations (21). However, the energy decrease property (16) is not guaranteed anymore. Indeed, looking at (21) and noting $\mathcal{S}_k : \beta \mapsto \mathcal{S}^\nu_{\lambda,q}(\beta, \eta^{(k)})$, from (9) and (21a), we have $E_{\lambda,q}(\beta^{(k+1)}) \leq \mathcal{S}_k(\beta^{(k+1)}) \leq \mathcal{S}_k(\beta^{(k)})$ but $\mathcal{S}_k(\beta^{(k)}) \neq E_{\lambda,q}(\beta^{(k)})$ since, for $\beta = \beta^{(k)}$ and $\eta = \eta^{(k)} \neq |\beta^{(k)}|^{\nu-q}$, the large inequality (9) is in fact a strict inequality. For that reason, in the general case, we won't have $E_{\lambda,q}(\beta^{(k+1)}) \leq E_{\lambda,q}(\beta^{(k)})$ using (21). Nevertheless, we show in Proposition 4 that, instead of $E_{\lambda,q}$, a slightly modified energy decreases along the iterations of (21). Besides, Scheme (21) corresponds to an instance of the Adaptive-Ridge Algorithm, as pointed out in Remark 2.

Lemma 2. *Let $\delta > 0$, $\nu > q$ and $E^{\nu,\delta}_{\lambda,q} : \mathbb{R}^p \to \mathbb{R}$ the energy defined by*

$$\forall \beta \in \mathbb{R}^p\,, \quad E^{\nu,\delta}_{\lambda,q}(\beta) = C(\beta) + \lambda \left\| |\beta|^\nu + \delta^\nu \right\|^{q/\nu}_{q/\nu}. \tag{22}$$

Then, we have

$$\forall \beta \in \mathbb{R}^p\,, \quad E^{\nu,\delta}_{\lambda,q}(\beta) = \min_{\eta\in(\mathbb{R}_+^*)^p} \left(\mathcal{S}^{\nu,\delta}_{\lambda,q}(\beta,\eta) := C(\beta) + \lambda \mathcal{L}^{\nu,\delta}_q(\beta,\eta)\right), \tag{23}$$

noting $\mathcal{L}^{\nu,\delta}_q(\beta,\eta) = \sum_{j=1}^p \frac{q}{\nu}\cdot\frac{|\beta_j|^\nu+\delta^\nu}{\eta_j} + \frac{\nu-q}{\nu}\cdot\eta_j^{\frac{q}{\nu-q}}$. Besides, the minimum in (23) is attained at $\eta = (|\beta|^\nu + \delta^\nu)^{\frac{\nu-q}{\nu}}$.

Proof. This result is a simple consequence of Proposition 1 after remarking that $\mathcal{L}^{\nu,\delta}_q(\beta,\eta) = \mathcal{L}^1_{\frac{q}{\nu}}(|\beta|^\nu + \delta^\nu, \eta)$. The proof is detailed in Appendix B. □

Proposition 4. *Let $\delta > 0$, $\nu > q$, $\beta^{(0)} \in \mathbb{R}^p$ and $\eta^{(0)} = \left(|\beta^{(0)}|^\nu + \delta^\nu\right)^{\frac{\nu-q}{\nu}}$. Then, the sequence of iterates $(\beta^{(k)})_{k\geq 0}$ generated using (21) satisfies*

$$\forall k \in \mathbb{N}\,, \quad E^{\nu,\delta}_{\lambda,q}(\beta^{(k+1)}) \leq E^{\nu,\delta}_{\lambda,q}(\beta^{(k)}). \tag{24}$$

Consequently, if C is bounded from below, then, the sequence $\left(E^{\nu,\delta}_{\lambda,q}(\beta^{(k)})\right)_{k\geq 0}$ is also bounded from below and, thus, converges.

Proof. Let $k \in \mathbb{N}$. By construction, we have $\eta^{(k)} = (|\beta^{(k)}|^\nu + \delta^\nu)^{\frac{\nu-q}{\nu}} \in (\mathbb{R}_+^*)^p$. Thus, from (23), we have $E^{\nu,\delta}_{\lambda,q}(\beta^{(k+1)}) \leq \mathcal{S}^{\nu,\delta}_{\lambda,q}(\beta^{(k+1)}, \eta^{(k)})$. Besides, since the

iterate $\beta^{(k+1)}$ computed using (21a) is a minimizer of $\beta \mapsto \mathcal{S}^{\nu,\delta}_{\lambda,q}(\beta, \eta^{(k)})$, we get $\mathcal{S}^{\nu,\delta}_{\lambda,q}(\beta^{(k+1)}, \eta^{(k)}) \leq \mathcal{S}^{\nu,\delta}_{\lambda,q}(\beta^{(k)}, \eta^{(k)})$ and, from Lemma 2, we have $\mathcal{S}^{\nu,\delta}_{\lambda,q}(\beta^{(k)}, \eta^{(k)}) = E^{\nu,\delta}_{\lambda,q}(\beta^{(k)})$. Finally, we obtain $E^{\nu,\delta}_{\lambda,q}(\beta^{(k+1)}) \leq E^{\nu,\delta}_{\lambda,q}(\beta^{(k)})$ as announced. □

Remark 2 (link between (21) and the $\mathrm{AR}^{\delta,\gamma}_{\lambda,q}$ scheme (6))**.** *Using again the variable change* $w_j^{(k)} = 1/\eta_j^{(k)}$ *and setting* $\nu = 2$ *in* (21) *yields exactly the* $AR^{\delta,\gamma}_{\lambda',q}$ *scheme with with* $\gamma = 2$ *(which is the setting recommended by the authors of [20]) and* $\lambda' = \lambda q$.

2.3. The alternating minimization strategy

In Section 2.2, we used Proposition 1 to derive numerical schemes for the minimization of $E_{\lambda,q}$ using the majorize-minimize framework. Interestingly enough, Proposition 1 can also be used to interpret those schemes in terms of alternating minimization framework. Indeed, from Proposition 1 we have

$$\forall \beta \in \mathbb{R}^p\,, \quad E_{\lambda,q}(\beta) = \inf_{\eta \in (\mathbb{R}_+^*)^p} \mathcal{S}^{\nu}_{\lambda,q}(\beta, \eta) \tag{25}$$

were $\mathcal{S}^{\nu}_{\lambda,q} : (\beta, \eta) \mapsto C(\beta) + \lambda\, \mathcal{L}^{\nu}_{q}(\beta, \eta)$ is the function defined in (9). Therefore, the optimization problem (3) is equivalent to

$$\operatorname*{argmin}_{\beta \in \mathbb{R}^p} \inf_{\eta \in (\mathbb{R}_+^*)^p} \mathcal{S}^{\nu}_{\lambda,q}(\beta, \eta)\,. \tag{26}$$

If the infimum in (26) was systematically attained, it could be replaced by a minimum, and we could implement the alternating minimization scheme

$$\begin{cases} \beta^{(k+1)} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \mathcal{S}^{\nu}_{\lambda,q}(\beta, \eta^{(k)}) & \text{(27a)} \\ \eta^{(k+1)} \in \operatorname*{argmin}_{\eta \in (\mathbb{R}_+^*)^p} \mathcal{S}^{\nu}_{\lambda,q}(\beta^{(k+1)}, \eta)\,. & \text{(27b)} \end{cases}$$

Assuming that $\beta^{(k+1)} \in (\mathbb{R}^*)^p$, then, problem (27b) admits $\eta^{(k+1)} = |\beta^{(k+1)}|^{\nu-q}$ as unique solution and (27) is exactly the same as (14) or (15). Of course, (27) exhibits the same limitation as (14) and (15), that is, it cannot be iterated anymore when one of the iterates $\beta^{(k+1)}$ admits vanishing entries. An alternative to (26) can be obtained using function r defined in (18) to reformulate (25) as a minimum, thanks to Proposition 5.

Proposition 5. *For any* $\nu > q$ *and for any* $\beta \in \mathbb{R}^p$, *we have*

$$E_{\lambda,q}(\beta) = \min_{\eta \in \mathbb{R}^p_+} \widetilde{\mathcal{S}}^{\nu}_{\lambda,q}(\beta, \eta)\,, \tag{28}$$

noting $\widetilde{\mathcal{S}}^{\nu}_{\lambda,q}(\beta, \eta) = C(\beta) + \sum_{j=1}^p \frac{\lambda q}{\nu} \cdot r(|\beta_j|^\nu, \eta) + \lambda \frac{\nu - q}{\nu} \cdot \eta_j^{\frac{q}{\nu-q}}$. *Besides, the minimum in* (28) *is attained at* $\eta = |\beta|^{\nu-q}$, *whatever the value of* β.

Proof. The proof is given in Appendix C. □

From Proposition 5, we can reformulate the optimization problem (3) as

$$\operatorname*{argmin}_{\beta\in\mathbb{R}^p} \min_{\eta\in\mathbb{R}_+^p} \widetilde{\mathcal{S}}^{\nu}_{\lambda,q}(\beta,\eta)\,, \tag{29}$$

so that an alternating minimization scheme for (29) can be obtained by replacing $\mathcal{S}^{\nu}_{\lambda,q}$ by $\widetilde{\mathcal{S}}^{\nu}_{\lambda,q}$ in (27), yielding exactly (19). As mentioned before, a limitation of such scheme is that any vanishing entry that it generates won't change anymore in all later iterations.

Instead of the minimization of the energy $E_{\lambda,q}$, one can consider the minimization of the modified energy $E^{\nu,\delta}_{\lambda,q}$ defined in (22). From Lemma 2, finding a minimizer of $E^{\nu,\delta}_{\lambda,q}$ is equivalent to computing

$$\operatorname*{argmin}_{\beta\in\mathbb{R}^p} \min_{\eta\in(\mathbb{R}_+^*)^p} \mathcal{S}^{\nu,\delta}_{\lambda,q}(\beta,\eta)\,. \tag{30}$$

An alternating minimization scheme for (30) can be obtained by replacing $\mathcal{S}^{\nu}_{\lambda,q}$ by $\mathcal{S}^{\nu,\delta}_{\lambda,q}$ in (27) yielding exactly (21).

2.4. The local approximation based strategy

Another very interesting way to address optimization problems like (3) is based on local approximations of the energy to minimize. Such approach was proposed in [17] in a slightly different framework than (3) since the authors focus on a variable selection model that boils down to the minimization of an energy $J_\lambda : \mathbb{R}^p \to \mathbb{R}$ of the type

$$\forall \beta \in \mathbb{R}^p\,, \quad J_\lambda(\beta) = C(\beta) + \lambda \sum_{j=1}^{p} p_j(|\beta_j|)\,, \tag{31}$$

where $\lambda > 0$ and where each p_j is a coordinate-wise penalty function from $\mathbb{R}_+$ to $\mathbb{R}$. The iterative local approximation framework proposed in [17] is the following. Assuming that we are able to design a function $G_\lambda : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ such that, for all $(\beta,\beta') \in \mathbb{R}^p \times \mathbb{R}^p$,

$$\beta \approx \beta' \Rightarrow J_\lambda(\beta) \approx G_\lambda(\beta,\beta') \quad \text{and} \quad J_\lambda(\beta') = G_\lambda(\beta',\beta') \tag{32}$$

and such that $G_\lambda(\beta,\beta')$ can be easily minimized over $\mathbb{R}^p$ with respect to β, then, from an initial guess $\beta^{(0)} \in \mathbb{R}^p$, one can iterate

$$\beta^{(k+1)} \in \operatorname*{argmin}_{\beta\in\mathbb{R}^p} G_\lambda(\beta,\beta^{(k)})\,. \tag{33}$$

Remark 3. *We can notice that, if $\beta \mapsto G_\lambda(\beta, \beta^{(k)})$ bounds J_λ from above, then (33) is a majorize-minimize scheme. Otherwise, we must hope that, at each step of (33), $\beta^{(k+1)}$ is close to $\beta^{(k)}$, so that the approximation (32) is accurate and the minimization of $\beta \mapsto G_\lambda(\beta, \beta^{(k)})$ is roughly related to that of J_λ in the vicinity of $\beta^{(k)}$.*

The authors of [17] designed G_λ by combining a second order Taylor approximation of C (assuming C twice continuously differentiable) with a local quadratic approximation (LQA) for each coordinate-wise penalty term. For all $j \in \{1, 2, \dots, p\}$, given $\beta_j^{(k)} \in \mathbb{R}^*$ and assuming p_j differentiable overs $\mathbb{R}_+^*$, the proposed LQA is

$$\forall \beta_j \in \mathbb{R}\,,\ \beta_j \approx \beta_j^{(k)} \Rightarrow p_j(|\beta_j|) \approx \underbrace{p_j(|\beta_j^{(k)}|) + \frac{1}{2}\frac{p_j'(|\beta_j^{(k)}|)}{|\beta_j^{(k)}|}\left(\beta_j^2 - {\beta_j^{(k)}}^2\right)}_{:=\mathrm{lqa}_j(\beta_j, \beta_j^{(k)})}, \quad (34)$$

and can be obtained by integrating over $[\beta_j^{(k)}, \beta_j]$ both sides of the linear approximation $g_j'(s) \approx \frac{p_j'(\beta_j^{(k)})}{|\beta_j^{(k)}|}\, s$ noting $g_j(s) = p_j(|s|)$ for all $s \in \mathbb{R}^*$ (see also [23]). The authors of [17] handle the computation of (33) using the Newton-Raphson Algorithm. Besides, when (33) generates an iterate $\beta^{(k+1)}$ with vanishing entry (or an entry smaller than a given threshold), they propose to keep this entry vanishing in all later iterations, not without mentioning the limitations of such strategy as we did in Sections 2.2 and 2.3.

In terms of optimization, unless when $\beta \mapsto G_\lambda(\beta, \beta^{(k)})$ bounds J_λ from above, a relation like (32) does not ensure the relevance of (33) regarding the minimization of J_λ since, as stated in Remark 3, we must have $\beta^{(k+1)} \approx \beta^{(k)}$ at each iteration to keep the approximation of J_λ by $\beta \mapsto G_\lambda(\beta, \beta^{(k)})$ accurate. In fact, the local approximation scheme (33) is probably efficient mostly in situations where it boils down to a majorize-minimize scheme. In [28, Proposition 3.1], some sufficient conditions on the coordinate-wise penalty functions $(p_j)_{1 \le j \le p}$ are given to ensure that each function $\beta_j \mapsto \mathrm{lqa}_j(\beta_j, \beta_j^{(k)})$ bounds $\beta_j \mapsto p_j(|\beta_j|)$ from above. One of those conditions is that each p_j is piecewise differentiable over $\mathbb{R}_+^*$ and that $p_j'(|\beta_j|)$ admits a (finite) limit as $\beta_j \to 0$. We can remark that if we set $p_j(|\beta_j|) = |\beta_j|^q$ in order to have $J_\lambda = E_{\lambda,q}$, then the limit of $p_j'(|\beta_j|)$ as $\beta_j \to 0$ is infinite as soon as $q < 1$ and [28, Proposition 3.1] cannot be applied in this situation.

The local approximation framework was further studied and extended in [56] where the authors propose, as an alternative to the LQA (34), to consider a locally linear approximation (LLA). Assuming each p_j differentiable over $\mathbb{R}_+$ and given $\beta_j^{(k)} \in \mathbb{R}$, the proposed LLA is given by

$$\forall \beta_j \in \mathbb{R}\,,\ \beta_j \approx \beta_j^{(k)} \Rightarrow p_j(|\beta_j|) \approx \underbrace{p_j(|\beta_j^{(k)}|) + p_j'(|\beta_j^{(k)}|)\left(|\beta_j| - |\beta_j^{(k)}|\right)}_{:=\mathrm{lla}_j(\beta_j, \beta_j^{(k)})}, \quad (35)$$

and can be derived from a simple first order Taylor approximation of p_j at point $|\beta_j^{(k)}|$. In [56, Theorem 1], the authors provide sufficient conditions to ensure that $\beta_j \mapsto \mathrm{lla}_j(\beta_j, |\beta_j^{(k)}|)$ bounds $\beta_j \mapsto p_j(|\beta_j|)$ from above and they derive further majorize-minimize schemes. However, the differentiability of each p_j at zero is required to apply [56, Theorem 1] and also to compute (35). Therefore, the choice $p_j(|\beta_j|) = |\beta_j|^q$ with $q < 1$ seems again problematic.

Let us focus back on the problem (3) and link the schemes presented in sections 2.2 and 2.3 to those local approximation based algorithms. First, remark that both the LQA and the LLA can be unified using a single formula, using, for $\beta_j \approx \beta_j^{(k)}$,

$$p_j(|\beta_j|) \approx g_j^\nu(\beta_j, \beta_j^{(k)}) := p_j(|\beta_j^{(k)}|) + \frac{1}{\nu} \frac{p_j'(|\beta_j^{(k)}|)}{|\beta_j^{(k)}|^{\nu-1}} \left(|\beta_j|^\nu - |\beta_j^{(k)}|^\nu \right), \tag{36}$$

which boils down to the LQA (34) for $\nu = 2$ (and $\beta_j^{(k)} \neq 0$), and to the LLA (35) for $\nu = 1$. Thus, defining

$$\forall \beta \in \mathbb{R}^p, \quad G_\lambda(\beta, \beta') = C(\beta) + \lambda \sum_{j=1}^p g_j^\nu(\beta_j, \beta_j') \tag{37}$$

yields a LQA (for $\nu = 2$ and $\beta' \in (\mathbb{R}^*)^p$) or a LLA (for $\nu = 1$ and $\beta' \in \mathbb{R}^p$) of J_λ that satisfies (32). Now, let us set $p_j(|\beta_j|) = |\beta_j|^q$ and $\nu \in \{1, 2\}$. Assuming $\beta_j^{(k)} \neq 0$, (36) yields

$$\begin{aligned} \forall \beta_j \in \mathbb{R}, \quad g_j^\nu(\beta_j, \beta_j^{(k)}) &= |\beta_j^{(k)}|^q + \frac{q}{\nu} |\beta_j^{(k)}|^{q-\nu} \left(|\beta_j|^\nu - |\beta_j^{(k)}|^\nu \right) \\ &= \frac{q}{\nu} |\beta_j^{(k)}|^{q-\nu} |\beta_j|^\nu + \frac{\nu - q}{\nu} |\beta_j^{(k)}|^q \\ &= \ell_q^\nu(\beta_j, \eta_j^{(k)}) \end{aligned} \tag{38}$$

noting $\eta_j^{(k)} = |\beta_j^{(k)}|^{\nu-q}$ and recognizing the function ℓ_q^ν defined in (7). Thus, when $q < \nu$, using Lemma 1, we get

$$\forall \beta_j \in \mathbb{R}, \quad g_j^\nu(\beta_j, \beta_j^{(k)}) = \ell_q^\nu(\beta_j, |\beta_j^{(k)}|^{\nu-q}) \geq |\beta_j|^q = p_j(|\beta_j|),$$

so that the local approximation $\beta_j \mapsto g_j^\nu(\beta_j, \beta_j^{(k)})$ does indeed bound $\beta_j \mapsto |\beta_j|^q$ from above. Besides, using $g_j^\nu(\beta_j, \beta_j^{(k)}) = \ell_q^\nu(\beta_j, |\beta_j^{(k)}|^{\nu-q})$ in (37) yields $G_\lambda(\beta, \beta^{(k)}) = \mathcal{S}_{\lambda,q}^\nu(\beta, |\beta^{(k)}|^{\nu-q})$ and Scheme (33) is indeed a majorize-minimize scheme which is nothing more than (13), although the sufficient conditions given in [28, Proposition 3.1] and [56, Theorem 1] are not fulfilled by $(p_j)_{1 \leq j \leq p}$. Imposing, as done in [17], vanishing entries $\beta_j^{(k)} = 0$ to be kept unchanged in later iterations yields exactly Scheme (19).

Instead of addressing the minimization of $E_{\lambda,q}$, one can focus on that of $E_{\lambda,q}^{\nu,\delta}$ for a given $\delta > 0$ by setting $p_j(|\beta_j|) = (|\beta_j|^\nu + \delta^\nu)^{\frac{q}{\nu}}$. In this case, both [28,

Proposition 3.1] and [56, Theorem 1] can be applied to show that g_j^ν bounds $\beta_j \mapsto p_j(|\beta_j|)$ from below. However, as we shall see now, this result can be proven very simply using the variational formulation of the ℓ^q penalty presented in Section 2.1. Indeed (36) yields

$$g_j^\nu(\beta_j, \beta_j^{(k)}) = \left(|\beta_j^{(k)}|^\nu + \delta^\nu\right)^{\frac{q}{\nu}} + \frac{q}{\nu}\left(|\beta_j^{(k)}|^\nu + \delta^\nu\right)^{\frac{q-\nu}{\nu}}\left(|\beta_j|^\nu - |\beta_j^{(k)}|^\nu\right)$$

and is well defined whatever the values of $\beta_j \in \mathbb{R}$ and $\beta_j^{(k)} \in \mathbb{R}$. Using

$$|\beta_j|^\nu - |\beta_j^{(k)}|^\nu = \left(|\beta_j|^\nu + \delta^\nu\right) - \left(|\beta_j^{(k)}|^\nu + \delta^\nu\right),$$

and setting $\eta_j^{(k)} = \left(|\beta_j^{(k)}|^\nu + \delta^\nu\right)^{\frac{\nu-q}{\nu}}$, we obtain

$$\begin{aligned} g_j^\nu(\beta_j, \beta_j^{(k)}) &= \frac{q}{\nu}\frac{|\beta_j|^\nu + \delta^\nu}{\eta_j^{(k)}} + \left(\frac{\nu - q}{\nu}\right)\left(\eta_j^{(k)}\right)^{\frac{q}{\nu-q}} \\ &= \ell_r^1(|\beta_j|^\nu + \delta^\nu, \eta_j^{(k)}) \quad \text{where} \quad r = \frac{q}{\nu}. \end{aligned} \tag{39}$$

Thus, from Lemma 1, we get $g_j^\nu(\beta_j, \beta_j^{(k)}) \leq \left(|\beta_j|^\nu + \delta^\nu\right)^r = p_j(|\beta_j|)$ as announced. Besides, using (39) in (37) yields $G_\lambda(\beta, \beta^{(k)}) = \mathcal{S}_{\lambda,q}^{\nu,\delta}(\beta, \eta^{(k)})$ where $\mathcal{S}_{\lambda,q}^{\nu,\delta}$ was defined in (23). Consequently, (33) is the same as (21) in this situation.

2.5. Links with other algorithms

In addition to the Adaptive-Ridge Algorithm, the numerical schemes derived above can be linked to many other algorithms. One of those is the famous Iteratively Reweighted Least Squares (IRLS) Algorithm proposed in [12] to address, given a matrix $X \in \mathbb{R}^{n\times p}$, a vector $y \in \mathbb{R}^n$, and for $q \in (0, 1]$, the constrained minimization problem

$$\underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \ \|\beta\|_q^q \quad \text{subject to} \quad X\beta = y\,. \tag{40}$$

Problem (40) is equivalent to (3) provided that we set $\mathscr{C} = \{\beta \in \mathbb{R}^p\,,\ X\beta = y\}$ and

$$\forall \beta \in \mathbb{R}^p, \quad C(\beta) = \delta_{\mathscr{C}}(\beta) := \begin{cases} 0 & \text{if } \beta \in \mathscr{C} \\ +\infty & \text{otherwise.} \end{cases} \tag{41}$$

The IRLS Algorithm described in [12] corresponds to (21) with $\nu = 2$ and with, in addition to (21a) and (21b), a clever update of the δ parameter (which is not fixed anymore and decreases along the iterations) depending on the value of the coordinate residuals $(|y_j - (X\beta)_j|)_{1\leq j\leq p}$ and the assumed number of nonzero

elements in the search for solution of (40). We refer to [12] (see also [41]) for more details about the IRLS Algorithm.

A variant of IRLS, usually referred as the IRL1 Algorithm, was proposed in [5] and further studied in [34]. This algorithm is equivalent to iterating Scheme (21) with the setting $C = \delta_{\mathscr{C}}$, $\nu = 1$ and provided that each iteration of (21) is done asymptotically for $q \to 0_+$. As for the IRLS Algorithm, an iterative update of the δ parameter is proposed in [5] although the use of a fixed value for δ is also considered. Notice that the authors of [20] remarked that the weight update of IRL1, that is, step (21b) with $\nu = 1$ and $q = 0$, is similar to step (6b) of their algorithm with the setting $\gamma = 1$ and $q = 1$. However, the comparison cannot be led any further because, under these settings, the steps (21a) and (6a) remain sensibly different and both algorithms probably do not address the minimization of the same energy. In particular, we are not able to identify the underlying energy minimization addressed by the $\mathrm{AR}^{\delta,\gamma}_{\lambda,q}$ scheme for $\gamma \neq 2$.

Another algorithm, namely the *Adaptive Lasso Algorithm*, was proposed in [54] for the purpose of variable selection. The Adaptive-Lasso Algorithm is a two steps algorithm. Given $\lambda' > 0$, the first step consists in computing a solution $\widehat{\beta}$ of the LASSO problem, i.e.,

$$\widehat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda' \|\beta\|_1 \,. \tag{42}$$

Then, the second step consists in finding

$$\overline{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda' \sum_{j=1}^{p} r(|\beta_j|, |\widehat{\beta}_j|^\gamma) \,, \tag{43}$$

where $\gamma > 0$ is an hyperparameter that must be set by the user. Assuming the uniqueness of the minimizer $\overline{\beta}$ of (43) to simplify the discussion, we can remark that $\overline{\beta}$ corresponds to the first iterate $\beta^{(1)}$ generated by Scheme (19) with the setting $\nu = 1$, $q < \nu$, $\lambda = \lambda'/q$, $C : \beta \mapsto \|y - X\beta\|_2^2$ and the initialization $\eta^{(0)} = |\widehat{\beta}|^\gamma$. Assuming a Gaussian distribution for the $(y_i)_{1 \leq i \leq n}$, the authors of [54] provide sufficient conditions for λ' and γ to ensure the *consistency of selection* and the *normality* of the Adaptive-Lasso estimate $\overline{\beta}$ as $n \to +\infty$ (see [54, Theorem 2] for more details).

3. Adaptive-Ridge and square-log penalized selection

As explained in Section 1, the ℓ^q penalized energy $E_{\lambda,q}$ (for $q > 0$) involved in (3) can be viewed as an approximation of the sparsity promoting $\mathcal{L}_0$-penalized energy $\mathcal{E}_\lambda$ defined in (1). In Section 2, we showed how the minimization of $E_{\lambda,q}$ or its approximation $E^{\delta,\nu}_{\lambda,q}$ (for $\nu = 2$), defined in (22), could be handled using the $\mathrm{AR}^{\delta,\gamma}_{\lambda_q,q}$ Algorithm (see Remark 1 and Remark 2). However, other smooth approximations of $\mathcal{E}_\lambda$, such as the square-log penalized energy $F_{\lambda,\delta}$ (for $\delta > 0$) defined in (5), can also be considered. The LQA methodology that we recalled

in Section 2.4 can be used to address the minimization of $F_{\lambda,q}$. Indeed, given $\delta > 0$ and letting

$$\forall \rho \in \mathbb{R}_+\,, \quad p_\delta(\rho) = \frac{\log\left(1 + (\rho/\delta)^2\right)}{\log\left(1+\delta^{-2}\right)}\,,$$

the LQA of p_δ given by (34) yields, for all $(\rho, \widetilde{\rho}) \in \mathbb{R}_+^2$,

$$\rho \approx \widetilde{\rho} \quad \Rightarrow \quad p_\delta(\rho) \approx \underbrace{\frac{\log\left(1 + (\widetilde{\rho}/\delta)^2\right)}{\log\left(1+\delta^{-2}\right)} + \frac{1}{\log\left(1+\delta^{-2}\right)} \cdot \frac{\rho^2 - \widetilde{\rho}^2}{\delta^2 + \widetilde{\rho}^2}}_{:=\mathrm{lqa}(\rho,\widetilde{\rho})}\,. \tag{44}$$

Since p_δ is continuously differentiable, nondecreasing and concave over $\mathbb{R}_+$, from [28, Proposition 3.1], we have, for all $\beta \in \mathbb{R}^p$, for all $\widetilde{\beta} \in (\mathbb{R}^*)^p$ and for all $j \in \{1, 2, \ldots, p\}$,

$$p_\delta\left(|\beta_j|\right) \leq \mathrm{lqa}\left(|\beta_j|, |\widetilde{\beta}_j|\right) \quad \text{and} \quad p_\delta\left(|\widetilde{\beta}_j|\right) = \mathrm{lqa}\left(|\widetilde{\beta}_j|, |\widetilde{\beta}_j|\right)\,. \tag{45}$$

One can easily check that (45) is in fact also valid for any $\widetilde{\beta} \in \mathbb{R}^p$ (i.e., without the restriction $\widetilde{\beta} \in (\mathbb{R}^*)^p$). Thus, letting

$$\forall (\beta, \widetilde{\beta}) \in \mathbb{R}^p \times \mathbb{R}^p\,, \quad G_\lambda(\beta, \widetilde{\beta}) = C(\beta) + \lambda \sum_{j=1}^p \mathrm{lqa}\left(|\beta_j|, |\widetilde{\beta}_j|\right)\,,$$

we obtain a majorize-minimize scheme for $F_{\lambda,\delta}$ by choosing $\beta^{(0)} \in \mathbb{R}^p$ and iterating for $k \geq 0$,

$$\beta^{(k+1)} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} G_\lambda(\beta, \beta^{(k)})\,. \tag{46}$$

Using (44) and removing from G_λ the terms that do not depend on β (and thus do not change the argmin in (46)), the majorize-minimize scheme (46) is equivalent to setting $\eta^{(0)} = {\beta^{(0)}}^2 + \delta^2$ and iterating for $k \geq 0$

$$\begin{cases} \beta^{(k+1)} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} C(\beta) + \dfrac{\lambda}{\log\left(1+\delta^{-2}\right)} \displaystyle\sum_{j=1}^p \frac{\beta_j^2}{\eta_j^{(k)}} & \text{(47a)} \\ \eta^{(k+1)} = \left(\beta^{(k+1)}\right)^2 + \delta^2\,. & \text{(47b)} \end{cases}$$

We can see that (47) corresponds exactly to the $\mathrm{AR}^{\delta,\gamma}_{\lambda',q}$ Algorithm with the setting $\gamma = 2$, $q = 0$ and $\lambda' = 2\,\lambda/\log\left(1+\delta^{-2}\right)$. Notice that the ability of the Adaptive-Ridge Algorithm to handle the minimization of the square-log penalized energy $F_{\lambda,\delta}$ using the majorize-minimize strategy was already pointed out in [24]. Note also that the LLA methodology could also be used to handle the minimization of $F_{\lambda,\delta}$ and would also yield a majorize-minimize scheme since p_δ satisfies the conditions of [56, Theorem 1].

4. Several implementations of the Adaptive-Ridge Algorithm

In this Section, given a matrix $X \in \mathbb{R}^{n\times p}$ and an element $y \in \mathbb{R}^n$, we focus on practical implementations of the AR scheme in the case where $C(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$. We will restrict our study to the case $\gamma = 2$, since, as mentionned in Section 2.5, we could not derive a satisfactory interpretation of the AR scheme in terms of energy minimization for other settings of γ. Given $q \in [0, 2)$, $\delta > 0$ and $\lambda' > 0$, the $\mathrm{AR}^{\gamma,\delta}_{\lambda',q}$ scheme (with $\gamma = 2$) boils down to setting $w^{(0)} \in (\mathbb{R}_+^*)^p$ and to iterating, for $k \geq 0$,

$$\begin{cases} \beta^{(k+1)} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \ \frac{1}{2}\|y - X\beta\|_2^2 + \frac{\lambda'}{2}\sum_{j=1}^{p} w_j^{(k)}\,\beta_j^2 & \text{(48a)} \\ w^{(k+1)} = \left(|\beta^{(k+1)}|^2 + \delta^2\right)^{\frac{q-2}{2}} . & \text{(48b)} \end{cases}$$

One can easily check that, when $\delta > 0$, problem (48a) admits a unique solution (by continuity, strict convexity and coercivity of the function to minimize). However, we will also extend the study to the case $\delta = 0$ and comment on our implementations in this situation.

4.1. System inversion based implementation

Given any vector $z \in \mathbb{R}^p$, we set $\mathscr{D}(z) = \mathrm{diag}(z_1, z_2, \ldots, z_p)$. When $\delta > 0$, we can easily check that all iterates $(w^{(k)})_{k\geq 1}$ generated using (48b) lie in $(\mathbb{R}_+^*)^p$. In this situation, (48a) can be written

$$\beta^{(k+1)} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \ \frac{1}{2}\|y - X\beta\|_2^2 + \frac{\lambda'}{2}\|\mathscr{D}((w^{(k)})^{1/2})\beta\|_2^2 \tag{49}$$

and admits as unique solution that of the linear system

$$\left(X^tX + \lambda'\mathscr{D}(w^{(k)})\right)\beta^{(k+1)} = X^ty\,. \tag{50}$$

Solving (50) is formally possible but can be numerically difficult because the matrix $X^tX + \lambda'\mathscr{D}(w^{(k)})$ can be badly conditioned, especially when δ is small. For instance, if we assume that the iterate $\beta^{(k)}$ generated using (48a) satisfies $\beta_j^{(k)} = 0$ and $\beta_{j'}^{(k)} = \alpha \neq 0$ for two indexes $j \neq j'$, then, we have $w_j^{(k)} = \delta^{q-2}$ and $w_{j'}^{(k)} = (|\alpha|^2+\delta^2)^{\frac{q-2}{2}}$. It follows that the condition number in ℓ^2 norm of $\mathscr{D}(w^{(k)})$ is larger than $(\frac{\delta^2}{|\alpha|^2+\delta^2})^{\frac{q-2}{2}}$, and thus, becomes arbitrary large as $\delta \to 0^+$. Although, in the general case, the bad conditioning of the matrix $\mathscr{D}(w^{(k)})$ does not necessarily imply a bad conditioning for the matrix $X^tX + \lambda'\mathscr{D}(w^{(k)})$, we observed in practice that such situation does make (50) ill-conditioned. To tackle this numerical difficulty we suggest to consider, instead of (50), the preconditioned system

$$\mathscr{D}(r^{(k)})\left(X^tX + \lambda'\mathscr{D}(w^{(k)})\right)\beta^{(k+1)} = \mathscr{D}(r^{(k)})X^ty \tag{51}$$

where $r^{(k)}$ corresponds to the vector made of the inverse of the diagonal elements of the matrix $X^tX + \lambda' \mathscr{D}(w^{(k)})$, i.e.,

$$\forall j \in \{1, 2, \dots, p\}\,, \quad r_j^{(k)} = \frac{1}{v_j + \lambda' w_j^{(k)}} \quad \text{noting } v_j = (X^tX)_{jj} = \sum_{i=1}^{n} X_{ij}^2\,. \tag{52}$$

Notice that using $v_j \geq 0$ and $w_j^{(k)} > 0$ in (52) yields $r_j^{(k)} > 0$. Thus, the diagonal matrix $\mathscr{D}(r^{(k)})$ involved in (51) is invertible and, therefore, (51) is equivalent to (50). The inverse of $\mathscr{D}(r^{(k)})$, that is the diagonal matrix made with the diagonal elements of $X^tX + \lambda' \mathscr{D}(w^{(k)})$, corresponds to the so-called *Jacobi preconditioner* of the system (50). The advantage of considering (51) instead of (50) in terms of system conditioning will be experimentally illustrated in Section 4.3.7.

Another practical advantage of (51) compared to (50) is that the matrix of system (51) remains finite as $\delta \to 0^+$, which is not necessarily true for that of system (50). Indeed from (48b), for all $j \in \{1, 2, \dots, p\}$, we have

$$\lim_{\substack{\delta \to 0 \\ \delta \geq 0}} w_j^{(k)} = \lim_{\substack{\delta \to 0 \\ \delta \geq 0}} \left(|\beta_j^{(k)}|^2 + \delta^2 \right)^{\frac{q-2}{2}} = \begin{cases} |\beta_j^{(k)}|^{q-2} & \text{if } \beta_j^{(k)} \neq 0 \\ +\infty & \text{otherwise.} \end{cases} \tag{53}$$

Therefore, as long as $\beta_j^{(k)} = 0$, the j-th diagonal element of the matrix of system (50) diverges to infinity as $\delta \to 0^+$. As we shall see now, this is not the case for system (51). Setting $z_j^{(k)} = r_j^{(k)} w_j^{(k)}$ for all $j \in \{1, 2, \dots, p\}$, we can write (51) as

$$\left(\mathscr{D}(r^{(k)}) X^tX + \lambda' \mathscr{D}(z^{(k)}) \right) \beta^{(k+1)} = \mathscr{D}(r^{(k)}) X^t y\,. \tag{54}$$

Besides, when $\beta_j^{(k)} = 0$, from (52) and (53), we have

$$\lim_{\substack{\delta \to 0 \\ \delta \geq 0}} r_j^{(k)} = 0 \quad \text{and} \quad \lim_{\substack{\delta \to 0 \\ \delta \geq 0}} z_j^{(k)} = \frac{1}{\lambda'} \tag{55}$$

so that the j-th diagonal element of the matrix of sytem (54) (or (51)) remains finite as $\delta \to 0^+$. Practically speaking, the setting $\delta = 0$ can be allowed in numerical implementations provided that we evaluate $r^{(k)}$ and $z^{(k)}$ using

$$\forall j \in \{1, 2 \dots, p\}, \quad r_j^{(k)} = \frac{\eta_j^{(k)}}{v_j \eta_j^{(k)} + \lambda'} \quad \text{and} \quad z_j^{(k)} = \frac{1}{v_j\, \eta_j^{(k)} + \lambda'}\,, \tag{56}$$

noting $\eta^{(k)} = (|\beta^{(k)}|^2 + \delta^2)^{\frac{2-q}{2}}$ (as we did in Section 2). Thus, assuming that $\beta_j^{(k)} = 0$ and $\delta = 0$, we obtain $\eta_j^{(k)} = 0$, $r_j^{(k)} = 0$ and $z_j^{(k)} = \frac{1}{\lambda'}$ in (56). Notice that, in such situation, $\mathscr{D}(r^{(k)})$ is not invertible. Therefore, (51) (or (54)) and (50) are not equivalent anymore and we can wonder how the solution of (54)

can be linked to (48a). In fact, in such situation, the j-th row of (54) simply yields $\beta_j^{(k+1)} = 0$ and we can easily show that the iterates $\beta^{(k+1)}$ generated by inversion of (54) are exactly the same as those generated using Scheme (19) with $\lambda = \lambda'/q$ and $\nu = 2$. Finally, the system inversion based implementation of the AR algorithm is summarized in Algorithm 1. Notice that an energy based criterion is used to stop the iterations in Algorithm 1. Other stopping criteria may be considered, for instance in situations where we are interested in the convergence of the iterates $(\beta^{(k)})_{k\geq 0}$ themselves rather than that of the targeted energy.

Algorithm 1: system inversion based implementation of the Adaptive-Ridge (`aridge_sysinv` module).

Inputs : a number $q \in [0, 2)$, a penalty parameter $\lambda' > 0$, a smoothing parameter $\delta \geq 0$ ($\delta = 0$ is allowed only when $q > 0$), a matrix $X \in \mathcal{M}_{n,p}(\mathbb{R})$, a vector $y \in \mathbb{R}^p$, an initial guess $\beta^{(0)} \in \mathbb{R}^p$, a tolerance parameter ε.

Output: an estimate of a local minimizer of the targeted energy, that is, of

$$\mathtt{E} : \beta \mapsto \frac{1}{2}\|y - X\beta\|_2^2 + \begin{cases} \frac{\lambda'}{q}\sum_{j=1}^{p}(\beta_j^2 + \delta^2)^{\frac{q}{2}} & \text{if } q \neq 0 \\ \frac{\lambda'}{2}\sum_{j=1}^{p}\log\left(1 + (\beta_j/\delta)^2\right) & \text{if } q = 0 \text{ (and } \delta \neq 0). \end{cases}$$

1 $v \leftarrow$ vector made of the diagonal elements of $X^t X$
2 $k \leftarrow 0$
3 **repeat**
4 $\quad \eta^{(k)} \leftarrow \left(|\beta^{(k)}|^2 + \delta^2\right)^{\frac{2-q}{2}}$
5 $\quad z \leftarrow \left(v \cdot \eta^{(k)} + \lambda'\right)^{-1}$
6 $\quad r \leftarrow \eta^{(k)} \cdot z$
7 $\quad \beta^{(k+1)} \leftarrow$ solution of $\left(\mathscr{D}(r)X^tX + \lambda'\mathscr{D}(z)\right)\beta = \mathscr{D}(r)X^t y$
8 $\quad k \leftarrow k + 1$
9 **until** $E(\beta^{(k-1)}) - E(\beta^{(k)}) \leq \varepsilon \cdot E(\beta^{(k-1)})$ `// energy-based stopping criterion`
10 **return** $\beta^{(k)}$

Lines 5 and 6: the dot product refers to the coordinatewise mutliplication between two vectors. Line 7: it is recommended to solve the linear sysem in the least-squares sense rather than by direct matrix inversion, this can be done using standard librairies such as LAPACK [1] (the latter being natively used in popular scientific programming langages such as R, Matlab or Octave).

4.2. Conjugate-Gradient based implementation

The system inversion based implementation of the Adaptive-Ridge Algorithm described in Section 4.1 relies on the inversion of a $p \times p$ sized linear system like (54) at each iteration (see Algorithm 1 line 7). Even though optimized solvers are available through standard libraries, the latter usually involve matricial manipulations (e.g., singular values decompositions, orthogonal factorization, QR decomposition, ...) which are computationally expensive, both in terms of time and memory (their typical complexity is $\Theta(p^3)$ of $\Theta(p^4)$ depending on the method). The Conjugate Gradient (CG) Algorithm is a famous iterative algorithm for solving symmetric and positive definite systems. The latter only requires matrix-vector multiplications (and thus $\Theta(p^2)$ operations) at each it-

eration. The convergence rate of the CG Algorithm depends on the spectral properties of the matrix of the system [48, 45, 38]. In many situations, it reveals faster and computationally less expensive than the previously mentioned methods. We shall describe now how the CG Algorithm can be used to implement the Adaptive-Ridge scheme (48). Notice that the idea of using the CG Algorithm within IRLS procedures is not new and was proposed in [18] (see also [41, Chapter 4]) in the framework of compressed sensing and sparse reconstruction.

Let us assume that $\delta > 0$. As we discussed in Section 4.1, solving (48a) is equivalent to solving the symmetric and positive definite linear system (50). To avoid bad conditioning issues with (50), we considered (54) in Section 4.1 as an equivalent preconditioned system, but the matrix of this system is not symmetric so that it cannot be addressed using the CG Algorithm. Another similar way to precondition (50) is the following. Let $k \geq 0$, let us keep the notations $r^{(k)}$ and $z^{(k)}$ defined in (56) and let us set $s^{(k)} = (r^{(k)})^{\frac{1}{2}}$. The assumption $\delta > 0$ ensures that the entries of $r^{(k)}$ are positive (see Section 4.1), and so are those of $s^{(k)}$. Thus, we can consider the bijective change of variables

$$\beta^{(k+1)} = \mathscr{D}(s^{(k)})\,\widetilde{\beta}^{(k+1)}\,, \tag{57}$$

and reformulate (50) into the equivalent form

$$\left(X^tX + \lambda'\mathscr{D}(w^{(k)})\right)\mathscr{D}(s^{(k)})\,\widetilde{\beta}^{(k+1)} = X^ty\,. \tag{58}$$

Left-multiplying (58) by $\mathscr{D}(s^{(k)})$ and using $\mathscr{D}(s^{(k)})\mathscr{D}(w^{(k)})\mathscr{D}(s^{(k)}) = \mathscr{D}(z^{(k)})$, we reformulate (58) into the equivalent linear system

$$\left(\mathscr{D}(s^{(k)})X^tX\mathscr{D}(s^{(k)}) + \lambda'\mathscr{D}(z^{(k)})\right)\widetilde{\beta}^{(k+1)} = \mathscr{D}(s^{(k)})X^ty\,, \tag{59}$$

which is a symmetric and positive definite system. Thus, the CG Algorithm can be used to approach the solution $\widetilde{\beta}^{(k+1)}$ of (59). Afterwards, one can retrieve the solution of (50) simply using (57). The CG Algorithm is summarized in Algorithm 2, and the CG based implementation of the Adaptive-Ridge Algorithm is summarized in Algorithm 3.

Remark 4. *One can easily check that the matrix of system (59) remains finite when $\delta = 0$. However, when $\delta = 0$, we have, for all $j \in \{1, 2, \ldots, p\}$, $\beta_j^{(k)} = 0 \Leftrightarrow r_j^{(k)} = 0 \Leftrightarrow s_j^{(k)} = 0$. When $s^{(k)}$ admits vanishing entries, the change of variables (57) is not bijective anymore so that we loose the equivalence between system (50) and the preconditioned system (59). Note also that $s_j^{(k)} = 0$ yields $\widetilde{\beta}_j^{(k+1)} = 0$ in (59) and $\beta_j^{(k+1)} = 0$ in (57). One can easily check that the iterates $(\beta^{(k)})_{k\geq 0}$ generated by Algorithm 3 with $\delta = 0$ are exactly the same as those generated using Scheme (19) with $\lambda = \lambda'/q$ and $\nu = 2$.*

Algorithm 2: Conjugate Gradient Algorithm (`cg` module)

Inputs : a symmetric and positive definite matrix A with order p, a vector $b \in \mathbb{R}^p$, an initial guess $x_0 \in \mathbb{R}^p$, a maximal number of iterations N_{iter} and a tolerance parameter ε_{CG}.

Output: an estimate of the solution of the linear system $Ax = b$

1 $r_0 \leftarrow b - Ax_0$
2 $\pi_0 \leftarrow r_0$
3 $k \leftarrow 0$
4 **while** $\|r_k\|_2 > \varepsilon_{\text{CG}} \cdot \|b\|_2$ *and* $k < N_{iter}$ **do**
5 $\quad \alpha \leftarrow \frac{\|r_k\|_2^2}{\pi_k^t A \pi_k}$
6 $\quad x_{k+1} \leftarrow x_k + \alpha \cdot \pi_k$
7 $\quad r_{k+1} \leftarrow r_k - \alpha \cdot A\pi_k$
8 $\quad \pi_{k+1} \leftarrow r_{k+1} + \frac{\|r_{k+1}\|_2^2}{\|r_k\|_2^2} \cdot \pi_k$
9 $\quad k \leftarrow k + 1$
10 **end**
11 **return** x_k

Convergence of the Conjugate Gradient Algorithm is theoretically ensured after at most p iterations. In practice, more than p iterations may be needed due to numerical error propagation (mainly round-off and cancellation errors). Line 7: in the absence of numerical errors, we would have $r_{k+1} = b - Ax_{k+1}$ at each iteration $k \geq 0$.

Algorithm 3: Conjugate Gradient based implementation of the Adaptive-Ridge (`aridge_cg` module).

Inputs : a number $q \in [0, 2)$, a penalty parameter $\lambda' > 0$, a smoothing parameter $\delta \geq 0$ ($\delta = 0$ is allowed only when $q > 0$), a matrix $X \in \mathcal{M}_{n,p}(\mathbb{R})$, a vector $y \in \mathbb{R}^p$, an initial guess $\beta^{(0)} \in \mathbb{R}^p$, two tolerance parameters ε and ε_{CG}.

Output: an estimate of a local minimizer of the targeted energy, that is, of

$$\mathtt{E} : \beta \mapsto \frac{1}{2}\|y - X\beta\|_2^2 + \begin{cases} \frac{\lambda'}{q}\sum_{j=1}^p (\beta_j^2 + \delta^2)^{\frac{q}{2}} & \text{if } q \neq 0 \\ \frac{\lambda'}{2}\sum_{j=1}^p \log\left(1 + (\beta_j/\delta)^2\right) & \text{if } q = 0 \text{ (and } \delta \neq 0). \end{cases}$$

1 $v \leftarrow$ vector made of the diagonal elements of $X^t X$
2 $k \leftarrow 0$
3 **repeat**
4 $\quad \eta^{(k)} \leftarrow \left(|\beta^{(k)}|^2 + \delta^2\right)^{\frac{2-q}{2}}$
5 $\quad z \leftarrow \left(v \cdot \eta^{(k)} + \lambda'\right)^{-1}$
6 $\quad s \leftarrow \left(\eta^{(k)} \cdot z\right)^{\frac{1}{2}}$
7 $\quad \widetilde{\beta}^{(k+1)} \leftarrow \mathtt{cg}\left(\mathscr{D}(s)X^tX\mathscr{D}(s) + \lambda'\mathscr{D}(z), \mathscr{D}(s)X^ty, \mathscr{D}(s)^\dagger\beta^{(k)}, p, \varepsilon_{\text{CG}}\right)$
8 $\quad \beta^{(k+1)} \leftarrow \mathscr{D}(s)\widetilde{\beta}^{(k+1)}$
9 $\quad k \leftarrow k + 1$
10 **until** $E(\beta^{(k-1)}) - E(\beta^{(k)}) \leq \varepsilon \cdot E(\beta^{(k-1)})$
11 **return** $\beta^{(k)}$

Lines 5 and 6: the dot product refers to the coordinatewise mutliplication between two vectors. Line 7 : $\mathscr{D}(s)^\dagger$ denotes the Moore-Penrose pseudoinverse of the diagonal matrix $\mathscr{D}(s)$, that is, the diagonal matrix obtained by inverting each nonzero diagonal term of $\mathscr{D}(s)$ and keeping all zero-values in place. One can check that, at each iteration $k \geq 1$, we have $\mathscr{D}(s)^\dagger\beta^{(k)} = \widetilde{\beta}^{(k)}$. At iteration $k = 0$, the initial guess $\mathscr{D}(s)^\dagger\beta^{(0)}$ used in the `cg` module is a rescaling of $\beta^{(0)}$ sharing the same support as $\beta^{(0)}$.

4.3. Numerical experiments

This section aims to illustrate the behavior of the Adaptive-Ridge Algorithm as well as the influence of its parameters (λ', q, δ) on the computed output. To facilitate the study, we will focus on synthetic datasets generated using a simple and standard simulation scheme that we shall now describe.

4.3.1. Simulation scheme

Given $(n, p) \in \mathbb{N}^2$, a synthetic dataset $(X, \beta^\star, y) \in \mathbb{R}^{n\times p} \times \mathbb{R}^p \times \mathbb{R}^n$ is obtained as follows.

(i) We compute a random matrix $\widetilde{X} \in \mathbb{R}^{n\times p}$ made of independant and uniformly distributed random entries $\widetilde{X}_{ij} \sim \mathcal{U}_{[0,1]}$. Then, we apply a columnwise normalization (to impose zero-mean and unitary ℓ^2 norm of the columns) using

$$\forall (i,j) \in \{1,2,\dots,n\} \times \{1,2,\dots,p\}, \quad X_{ij} = \frac{\widetilde{X}_{ij} - \mu_j}{n_j},$$

where $\mu_j = \sum_{i=1}^n \widetilde{X}_{ij}$ and $n_j^2 = \sum_{i=1}^n (\widetilde{X}_{ij} - \mu_j)^2$.

(ii) We compute a random sparse reference signal $\beta^\star \in \mathbb{R}^p$ (referred as *ground-truth* in the following) using

$$\forall j \in \{1,2,\dots,p\}, \quad \beta_j^\star = \begin{cases} 1 & \text{if } \delta_j > 0.95 \\ 0 & \text{otherwise} \end{cases}$$

where $\{\delta_j\}_{1\le j\le p}$ are independant and uniformly distributed random variables. By construction, in average 95% of the entries of $\beta^\star$ are zero, the 5% remaining are equal to one.

(iii) We compute a random observation $y \in \mathbb{R}^n$ using

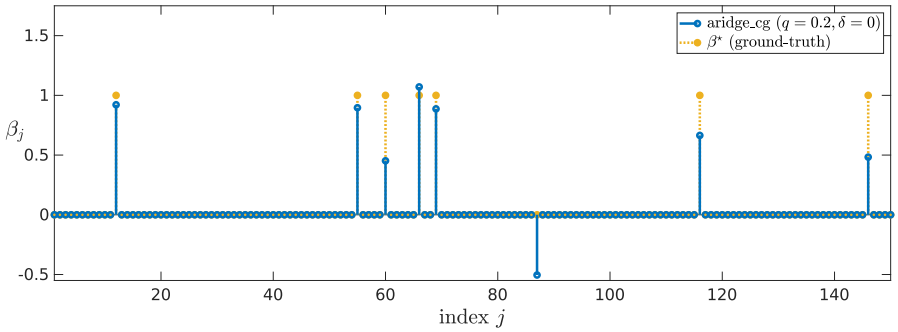
$$\forall i \in \{1,2,\dots,n\}, \quad y_i = (X\beta^\star)_i + \sigma\varepsilon_i \tag{60}$$

where $\sigma = 0.2$ and $\{\varepsilon_i\}_{1\le i\le n}$ are independant Gaussian random variables following a normal distribution and $(X\beta^\star)_i$ denotes the i-th entry of $X\beta^\star$.

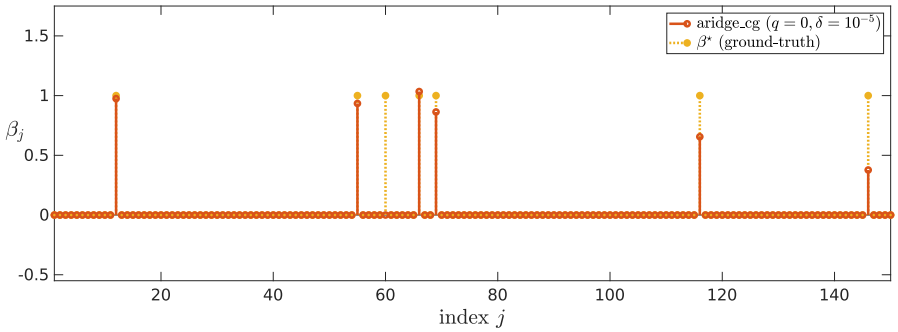
Unless explicitly mentioned, we will consider the setting $n = 300$ and $p = 150$ in our simulations. We experimentally checked that both Algorithm 1 and Algorithm 3 returned the same output (up to machine precision) when applied to the same dataset with the same setting for their parameters $(q, \lambda', \delta, \varepsilon)$ and taking $\varepsilon_{\text{CG}} = 10^{-16}$ in Algorithm 3. This is not surprising since both algorithms implement the same numerical scheme (48). However, they both differ in terms of execution time, as it will be illustrated in Section 4.3.5.

4.3.2. First examples of AR estimates

Examples of estimates produced by the Adaptive-Ridge Algorithm applied to a synthetical dataset $(X, \beta^\star, y)$ with the two settings $(q = 0.2, \delta = 0, \lambda' = 0.1)$ and $(q = 0, \delta = 10^{-5}, \lambda' = 0.1)$ are displayed in Fig 2. The two settings of (q, δ) yield two different estimates, since the AR scheme (48) addresses the minimization of two different energies: a ℓ^q penalized energy (when $q > 0$ and $\delta \geq 0$), or a log-square penalized energy (when $q = 0$ and $\delta > 0$).



(a) estimated (local) minimizer of the ℓ^q penalized energy

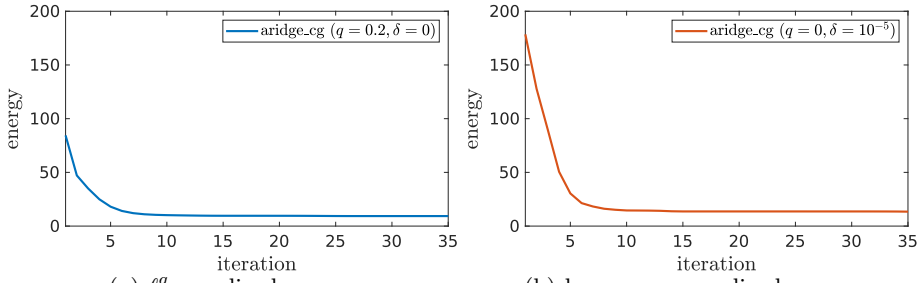


(b) estimated (local) minimizer of the log-square penalized energy

FIG 2. ***Example of estimates produced by the AR algorithm.*** *A simulated dataset $(X, \beta^\star, y)$ was processed using Algorithm 3 with the setting $\lambda' = 0.1$, $\varepsilon = \varepsilon_{\text{CG}} = 10^{-16}$, a random initial guess $\beta^{(0)}$ (with i.i.d. entries following a uniform distribution), and the two settings $(q = 0.2, \delta = 0)$ and $(q = 0, \delta = 10^{-5})$. We display with blue plain stems in (a) the estimate obtained using $(q = 0.2, \delta = 0)$ and with orange plain stems in (b) the estimate obtained using $(q = 0, \delta = 10^{-5})$. The ground-truth $\beta^\star$ is displayed in both graphs using yellow dotted stems. Those two settings yield two different estimates that are both close to $\beta^\star$, which gives first comforting insights on the ability of those optimization models to be useful for variable selection. In fact, with $q = 0.2 \neq 0$, Algorithm 3 addresses the minimization of a ℓ^q penalized energy, while with $q = 0$, a log-square penalized energy minimization is performed. The reason why the estimates displayed in (a) and (b) are different from $\beta^\star$ is simply that the ground truth $\beta^\star$ is not the minimizer of the considered energies (at least for the considered settings of q, δ and λ'). The ability of Algorithm 3 to efficiently address the minimization of those energy (independantly from the quality of the produced estimate) is illustrated in Fig. 3.*

4.3.3. Energy decrease

In Remark 4, we pointed out that the iterates $(\beta^{(k)})_{k\geq 0}$ generated using Algorithm 3 with $q > 0$ and $\delta = 0$ are the same as those generated using Scheme (19) (with $\lambda = \lambda'/q$ and $\nu = 2$). Since $C : \beta \mapsto \sum_{i=1}^{n}(y_i - (X\beta)_i)^2$ is additively separable, from Proposition 3, the sequence of ℓ^q penalized energies $\left(E_{\lambda,q}(\beta^{(k)})\right)_{k\geq 0}$ decreases along the scheme iterations. This result was experimentally highlighted in Fig. 3 (a). Similarly, when $q = 0$ and $\delta > 0$, Algorithm 3 ensures the decrease of the log-square penalized energy $F_{\lambda,\delta}$ (with $\lambda = \lambda' \log(1 + \delta^{-2})/2$), as explained in Section 3 and experimentally illustrated in Fig. 3 (b).

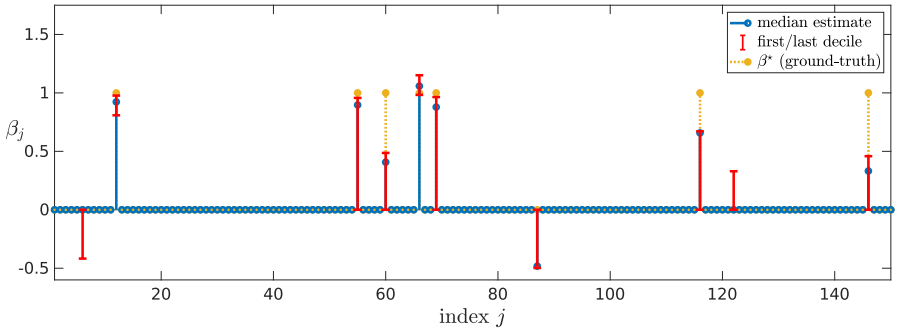


(a) ℓ^q penalized energy (b) log-square penalized energy

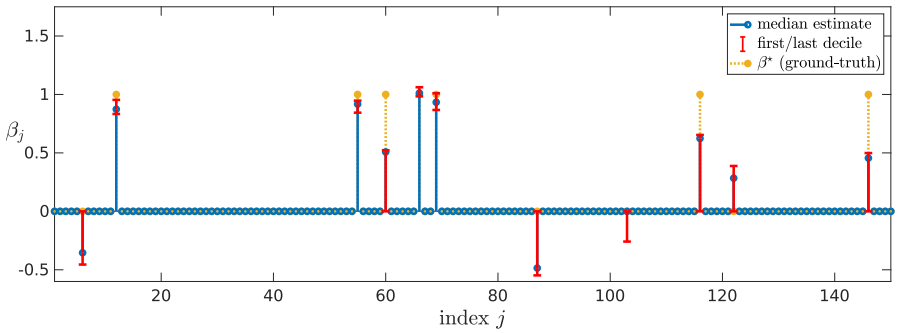
FIG 3. ***Energy decrease along with the AR iterations.*** *We consider exactly the same dataset and experiments as those presented in Fig. 2. We display in (a) the evolution of the ℓ^q penalized energy $k \mapsto E_{\lambda,q}(\beta^{(k)})$ (with $q = 0.2$, $\lambda = \lambda'/q = 0.5$ and $1 \leq k \leq 35$), denoting $\beta^{(k)}$ the k-th iterate computed using Algorithm 3 with the setting ($q = 0.2, \delta = 0$). We display in (b) the evolution of the log-square penalized energy $k \mapsto F_{\lambda,\delta}(\beta^{(k)})$ (with $\delta = 10^{-5}$ and $\lambda = \lambda' \log(1 + \delta^{-2})/2 \approx 1.1513$), where $\beta^{(k)}$ denotes this time the k-th iterate computed using Algorithm 3 with the setting ($q = 0, \delta = 10^{-5}$). In both cases, we can see that the considered energy decreases along with the AR iterations, as showed in Proposition 3 or in Section 3.*

4.3.4. Sensitivity to the choice of the initializer

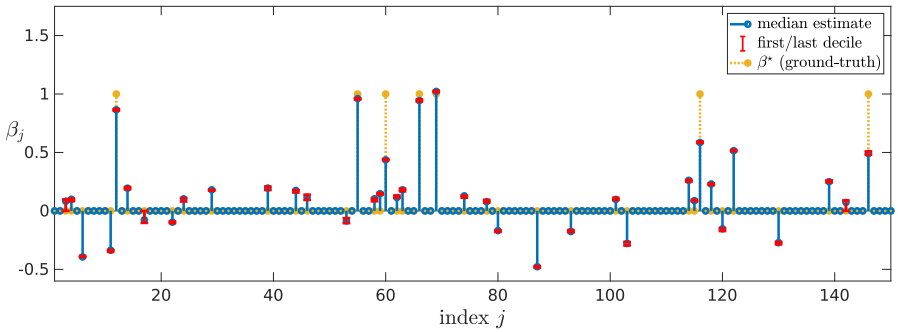
Addressing the minimization of non-convex energies is a difficult task, partly due to the presence of local extrema in the energy to minimize, which makes most non-convex optimization algorithms sensitive to the choice of the initializer. This is also the case for the Adaptive-Ridge scheme (48). For a given dataset $(X, \beta^\star, y)$, the output of algorithms 1 and 3, as well as the achieved value of the energy to minimize, may be strongly dependent on the choice of the initializer $\beta^{(0)}$. This is especially true for small values of q, as illustrated in Fig. 4. Interestingly enough, the sensitivity of the Adaptive-Ridge scheme to the choice of the initializer seems to decrease as q increases from 0 to 1, with no meaningful variability observed when $q = 1$. In fact, when $q = 1$ the



(a) module aridge_cg with $q = 0.1$ and $\delta = 0$



(b) module aridge_cg with $q = 0.3$ and $\delta = 0$



(c) module aridge_cg with $q = 0.8$ and $\delta = 0$

FIG 4. ***Variability of the estimated local minimizers with respect to the initialization.*** *We executed Algorithm 3 over a single dataset $(X, \beta^\star, y)$ using 10^4 random initial guesses $\beta^{(0)}$ (simulated using a uniform distribution), and setting $\lambda' = 0.1$, $\varepsilon = \varepsilon_{\text{CG}} = 10^{-16}$, $\delta = 0$ and $q \in \{0.1, 0.3, 0.8\}$. For each value of q, we display above using blue plain stems the median estimate computed among the 10^4 simulations (one simulation per initial guess $\beta^{(0)}$), and we indicate with red error bars the first and last decile estimate values of each entry except for entries where no variability was observed (in this case no error bar is displayed). We can see that most of the vanishing entries remained stable (i.e., remained vanishing whatever the initial guess) in this simulation. However, one can remark that some non vanishing entries exhibit a large variability with respect to the choice of the initializer. This is especially true for small values of q (see (a) and (b)) and this variability decreases as q gets close to 1 (see (c)). The sensitivity of the AR scheme to the choice of the initializer is a consequence of the nonconvexity (when $q < 1$) of the energy to minimize. However, when $q = 1$ the energy is convex and does not assume local minima. This experiments illustrates that the influence of local minima of the ℓ^q penalized energy on the estimate produced by the AR scheme decreases as q increases.*

underlying ℓ^1 penalized energy to minimize is convex and admits a unique minimizer (since X is of full rank in our simulation scheme). However, for $q < 1$, the number of local minima of the ℓ^q penalized energy probably increases as q decreases, and as a result, the scheme becomes more sensitive to the choice of the initializer. This can also be observed in Table 1 where we study the sensitivity of the energy of the final iterate produced by Algorithm 3 with respect to the choice of the initializer. Looking at the last column of Table 1, we can see that this sensitivity decreases as q increases from zero to one (with no quantitatively meaningful variability observed for $q = 1$).

TABLE 1

Sensitivity of the AR scheme to the initialization. As in Fig. 4. We considered one single $(X, \beta^\star, y)$ dataset and 10^4 random initial guesses $\beta^{(0)}$. We provide here some statistics on the energy of the estimate returned by Algorithm 3 with respect to the choice of the initial guess $\beta^{(0)}$ of the AR scheme (other parameters for Algorithm 3: $\lambda' = 0.1$, $\varepsilon = \varepsilon_{\text{CG}} = 10^{-16}$).

Setting		**energy of the Adaptive-Ridge output estimate**					
		min	**first decile**	**median**	**last decile**	**max**	$\frac{\textbf{max}-\textbf{min}}{\textbf{median}}$
$q = 0$	$\delta = 10^{-5}$	9.128	11.29	12.73	13.87	14.51	42 %
$q = 0.2$	$\delta = 0$	8.56	9.076	9.353	9.694	9.958	15 %
$q = 0.4$	$\delta = 0$	7.332	7.409	7.466	7.658	8.043	9.5 %
$q = 0.6$	$\delta = 0$	6.58	6.582	6.591	6.664	7.044	7 %
$q = 0.8$	$\delta = 0$	5.928	5.93	5.932	5.936	6.018	1.5 %
$q = 1$	$\delta = 0$	5.197	5.197	5.197	5.197	5.197	$5.2 \cdot 10^{-9}$

Another interesting point is that the variability of the output estimate with respect to the choice of the initializer seems to be restricted to some coordinates only. Indeed, we can observe in Fig. 4 that most vanishing coordinates of the estimates remained stable (i.e. they remained vanishing) in all our simulations. This means that the sensitivity of the scheme with respect to the initializer may not equally affect all entries of the estimator. However, this observation may also be a consequence of the restricted range of initializers considered in our simulations.

4.3.5. Execution time

As mentioned before, Algorithm 1 and Algorithm 3 both implement the same scheme, and thus, they both produce the same estimate (up to numerical rounding errors) when applied to the same dataset and using the same input parameter values. However, since those two algorithms are implemented in different ways, they exhibit different performances in terms of computation time. Algorithm 1 relies on the direct inversion of a linear system at each iteration, while, with Algorithm 3, the system is symmetrized and its inversion is specifically handled using the CG Algorithm (as explained in Section 4.2). The evolution of the computation time for both algorithms with respect to the dimension p of the problem is studied in Fig. 6. Unsuprisingly, we found that Algorithm 3 becomes significantly faster than Algorithm 1 for large values of p.

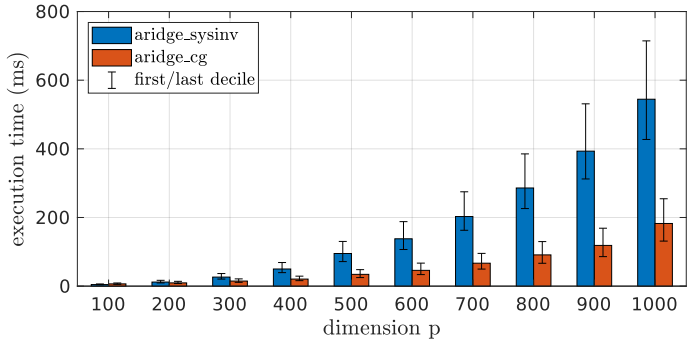
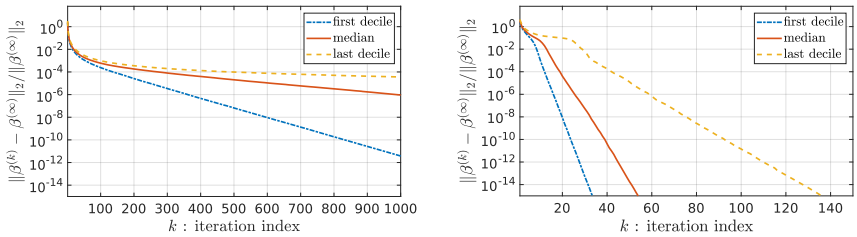


FIG 5. ***Execution time.*** *We measured the execution time of Algorithm 1 and Algorithm 3 applied to simulated datasets* $(X, \beta^\star, y)$ *with matrix* X *of size* $n \times p$ *for* $n = 500$ *and various values of* p *between* 100 *and* 1000. *We performed* 10^3 *simulations per tested value of* p *and executed both algorithms using* $q = 0.1$, $\delta = 0$, $\lambda' = 0.1$, $\varepsilon = 10^{-5}$ *(and* $\varepsilon_{\text{CG}} = 10^{-16}$ *for Algorithm 3). For each simulation, a given* $\beta^{(0)}$ *(randomly sampled according to a uniform distribution) was used as initial guess for both algorithms. For each value of* p, *we display using a blue bar the median execution time obtain with Algorithm 1 among the* 10^3 *simulations, and using an orange bar that obtained using Algorithm 3. Error bars indicate the first and last decile of the computation times measured among the* 10^3 *simulations. We can see that, the execution time of both algorithms is comparable for* $p < 300$. *However, we can see that as* p *increases, Algorithm 3 becomes significantly faster than Algorithm 3. This is not surprising since Algorithm 3 exhibits a* $\Theta(p^2)$ *temporal complexity while that of Algorithm 1 is in* $\Theta(p^3)$. *Notice that the same study was led for* $n = 100$ *and* $n = 1000$ *and yielded the same conclusion (results not shown).*

4.3.6. Convergence of the scheme iterates

Results of convergence for the iterates generated by the IRLS Algorithm in the constrained framework (40) can be found in [12] and references therein (we recall that the link between the IRLS and AR algorithms in this constrained framework was discussed in Section 2.5). In [12, Theorem 5.3], convergence with a linear rate (that is, exponential decay of the distance between the scheme iterates and their limit) of the sequence generated by the IRLS scheme toward a solution of (40) is shown for $q = 1$. In the setting $0 < q < 1$, convergence with a superlinear rate of the IRLS scheme toward a (local) solution of (40) is established in [12, Theorem 7.9]. However, the AR scheme (48) considered in our experimental section does not address the constrained problem (40) but the minimization of a ℓ^q penalized energy which can be viewed as a *relaxed* variant of the constrained problem (40). Some variants of the IRLS scheme in such relaxed framework are described in [31, 49, 18] and boil down to the same scheme as the AR scheme (48) provided that an appropriate update of the δ parameter is done at each iteration. Convergence results are also established in this relaxed framework but in a weaker sense (only the existence of accumulation points of the sequence of the IRLS iterates is shown, see [18, Theorem 5] and references therein). In this work, we performed an empirical study of the convergence of the AR scheme. It seems that a linear convergence rate is achieved for $0 < q \leq 1$ and that this rate improves as q decreases, as illustrated in Fig. 6.



(a) Algorithm 3 with $q = 1$ (b) Algorithm 3 with $q = 0.1$

FIG 6. ***Convergence of the AR scheme.*** *We simulated* 10^4 *synthetic datasets* $(X, \beta^\star, y)$. *Each dataset was processed using Algorithm 3 with* $\lambda' = 0.1$, $q \in \{0.1, 1\}$ *and a random initial guess. For each simulation, we comptuted the relative distance between the iterate* $\beta^{(k)}$ *and* $\beta^{(1000000)}$ *(denoted* $\beta^{(\infty)}$*). We display here the first, median and last deciles of this relative error as functions of* k. *For both* $q = 0.1$ *and* $q = 1$, *we can observe a relatively slow decrease rate of the relative error in* $\mathcal{O}(1/k)$ *during the first iterations, followed by a faster decrease rate in* $\mathcal{O}(e^{-\alpha\, k})$ *for a given* $\alpha > 0$. *This rate is much more faster for* $q = 0.1$ *than for* $q = 1$. *Interestingly enough, the same behavior was pointed out in [32] for another class of convex minimization algorithms (the so-called* Forward-Backward *splitting algorithms).*

4.3.7. Systems conditioning

In Section 4.1, we pointed out that some care should be taken when addressing the numerical computation of step (48a) of the AR scheme. Indeed, if computing (48a) is formally equivalent to solving the linear system (50), we explained that the latter may exhibit bad conditioning, especially for small values of δ. This phenomenon is indeed illustrated in Table 2, where we report some statistics on the condition number (in ℓ^2 norm) of all matrices (50) obtained when implementing the AR scheme (with $q = 0.1$, $\lambda' = 0.1$ and several values of δ) to process 10^4 simulated datasets $(X, \beta^\star, y)$. We can see in Table 2 that *large*

TABLE 2

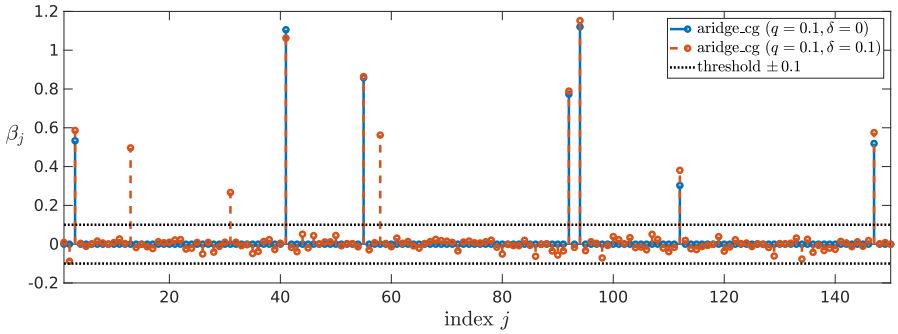
Setting	**Condition numbers for systems (50)**		
	first decile	**median**	**last decile**
$\delta = 10^{-1}$	11	12	14
$\delta = 10^{-5}$	$6.6 \cdot 10^7$	$3.3 \cdot 10^8$	$3.7 \cdot 10^8$
$\delta = 10^{-10}$	$4.9 \cdot 10^8$	$4.6 \cdot 10^{18}$	$5.2 \cdot 10^{20}$

enough values of δ (such as $\delta = 0.1$ in our experiments, but, as we will discuss in the next section, this setting is not scale invariant) yield well conditioned systems. However, as δ decreases, we rapidely observe extremely badly conditioned systems. This means that, for *small* values of δ, even small errors on the input observation y may be dramatically amplified during the inversion process. Notice that the authors of [20] recommend the setting $\delta = 10^{-5}$. However, as we will observe in the next section, in practical situations this setting turns out to be equivalent to the setting $\delta = 0$ (provided that bad conditioning issues are properly handled).

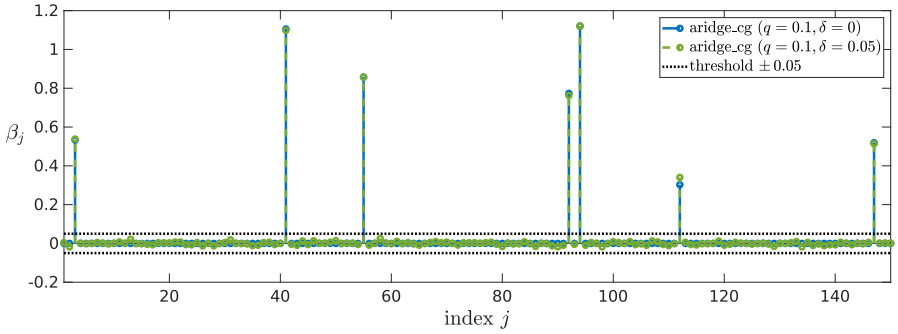
Repeating the same experiment as in Table 2 but considering the preconditioned linear systems (51) involved in Algorithm 1 or the preconditioned and symmetrized linear systems (54) involved in Algorithm 3, we observed condition numbers of at most 9 among all simulations, for $\delta \in \{10^{-10}, 10^{-5}, 10^{-1}\}$ and even for $\delta = 0$. This empirically confirms the benefit of those preconditioning operations regarding the numerical implementation of the AR scheme.

4.3.8. Influence of the δ parameter

In Fig. 7, we display a simple example of estimates produced by the AR scheme with $q = 0.1$, $\lambda' = 0.1$ and $\delta \in \{0, 0.05, 0.1\}$.



(a) module aridge_cg with $\delta = 0$ or $\delta = 0.1$



(b) module aridge_cg with $\delta = 0$ or $\delta = 0.05$

FIG 7. ***Influence of the δ parameter.*** *We processed a single synthetic dataset $(X, \beta^\star, y)$ using Algorithm 3 with $\lambda' = 0.1$, $q = 0.1$ and various settings of δ. The estimate obtained using $\delta = 0$ is displayed using blue plain stems in both (a) and (b). The estimate obtained using $\delta = 0.1$ is displayed using orange dashed stems in (a) and that obtained using $\delta = 0.05$ is displayed using green dashed stems in (b). We can see in (a) that the settings $\delta = 0$ and $\delta = 0.1$ yield sensibly different estimates. However, one can see in (b) that setting $\delta = 0.05$ yields an estimate closer to that obtained for $\delta = 0$. More comments on this experiments are given in the text below. Besides, a more exhaustive comparison between the estimates obtained using $\delta = 0$ and $\delta > 0$ is proposed in Fig. 8.*

In Fig. 7 (a), we can see that the estimate computed with $\delta = 0.1$ is sensibly different from that obtained using $\delta = 0$. Decreasing the value of δ yields closer estimates, as illustrated in Fig. 7 (b) with $\delta = 0.05$, and more exhaustively highlighted in Fig. 8.

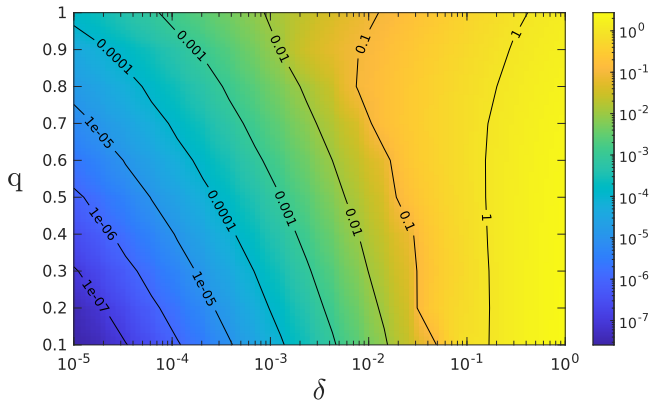


FIG 8. ***Distance between estimates obained with $\delta = 0$ and $\delta > 0$.*** *In this experiment, we simulated 10^4 synthetic datasets $(X, \beta^\star, y)$. Each dataset was processed by Algorithm 3 using a random inital guess $\beta^{(0)}$ and setting $\lambda' = 0.1$, $q \in [0.1, 1]$, $\varepsilon = \varepsilon_{\text{CG}} = 10^{-16}$ and $\delta \in [0, 1]$. After each experiment, we computed the Euclidean distance between the estimate produced using $\delta = 0$ and $\delta > 0$. For each setting of (q, δ), we computed the median distances observed over the 10^4 simulations and we display here, using false colors and level lines, the evolution of this distance as a function of (q, δ). We can see that, the estimates obtained using $\delta > 0$ gets rapidly significantly close to that obtained for $\delta = 0$ as δ decreases.*

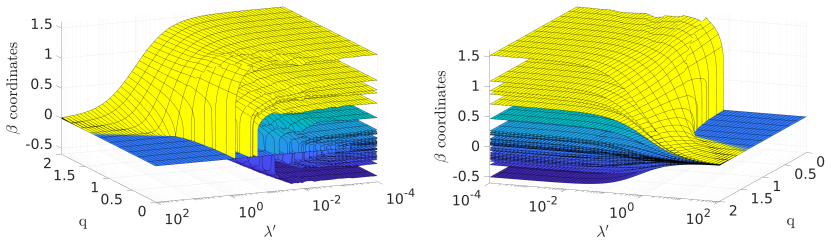
The proximity observed in Fig. 8 between the estimates computed using *small enough* values of δ and that computed using $\delta = 0$ was somehow expected since the underlying energies to be minimized, i.e. $E^{\nu,\delta}_{\lambda,q}$ (with $\nu = 2$ and $\lambda = \lambda'/q$) when setting $\delta > 0$ in Algorithm 3 and $E_{\lambda,q}$ when setting $\delta = 0$ in Algorithm 3, are asymptotically the same as $\delta \to 0$. However, we observed that a noticeable difference remains between the estimates produced using $\delta = 0$ and $\delta > 0$: those obtained using $\delta = 0$ are sparse, while those obtained using $\delta > 0$ exhibit no vanishing entries. Since producing sparse estimates is the initial motivation for addressing ℓ^q penalized minimization, when the setting $\delta > 0$ is considered, a natural idea is to *hard threshold* the produced estimate using δ as threshold parameter. This means that at the end of the execution of the AR scheme, the entries of the produced estimate with absolute values less than δ are replaced by 0. As illustrated in Fig. 7, we observe that, as long as δ is small enough, this thresholding operation produces estimates with the same support as those obtained using $\delta = 0$.

In [20], the authors recommend the setting $\delta = 10^{-5}$ claiming that $\delta = 0$ would be more attractive but would cause numerical instabilities. We have several objections about this recommendation. First, the setting $\delta = 10^{-5}$ is not scale invariant. If we multiply by a given scalar number the ground-truth $\beta^\star$ in

the simulation scheme, the δ parameter will also need to be rescaled accordingly. Second, thanks to preconditioning, the AR scheme does no longer suffer from numerical instabilities caused by the setting $\delta = 0$ (nor by *small* values of δ). Besides, as far as we could observe in our simulations, the setting $\delta \leq 10^{-5}$ yields estimates with no substantial differences compared to those computed using $\delta = 0$ (as long as the initial guess $\beta^{(0)}$ does not assume vanishing entries). From a practical viewpoint, the setting $\delta = 0$ avoids the tuning of δ, the scale invariance issues, and produces estimates that can assume vanishing entries so that no further thresholding operation is necessary. For sure, the choice $\delta = 0$ also comes with drawbacks that we already discussed before. More advanced schemes with adaptive setting of the δ parameter can be found in the field of compressed sensing with IRLS algorithms [12, 18, 41] although the uptade of δ also raises some numerical issues, as pointed out in [41, Section 2.4.1.2] and [18].

4.3.9. Regularization paths

The quality of the estimates produced by the AR Algorithm largely depends on the setting of the q and λ' parameters, as well as on the sparsity of the targeted $\beta^\star$ vector involved in our simulation scheme. In the statistics community, this aspect is usually studied through the so-called *regularization paths*. The latter correspond to the evolution of each entry of the AR estimate as a function of q and λ' (2D regularization paths), or as a function of λ' for a fixed value of q (1D regularization paths). An example of 2D regularization path is displayed and commented in Fig. 9. The 1D regularization paths for $q \in \{0.1, 1, 1.9\}$ extracted from this 2D regularization path are displayed in Fig. 10 (first column).



(a) view from one side (b) view from the other side

FIG 9. ***Two-dimensional regularization paths.*** *We synthetized a random dataset* $(X, \beta^\star, y)$ *with dimensions* ($n = 300$, $p = 50$) *and such as* $\|\beta^\star\|_0 = 5$. *We used Algorithm 3 to process this dataset for a large range of parameters* $(q, \lambda') \in [0.01, 1.99] \times [10^{-8}, 10^2]$, *using* $\delta = 0$ *and* $\varepsilon = \varepsilon_{\text{CG}} = 10^{-16}$. *Denoting* $\widehat{\beta}^{q,\lambda'}$ *the estimate produced using* (q, λ') *as input parameters, we display using two dimensional surfaces the evolution of each coordinate* $\widehat{\beta}^{q,\lambda'}_j$ *(for* $1 \leq j \leq p$*) as a function of* (q, λ'). *The five coordinates corresponding to* active *coordinates in the ground truth signal* $\beta^\star$, *(i.e., the coordinates* j *such as* $\beta^\star_j \neq 0$*) are drawn in yellow color, while we used blue color for the other coordinates. Correct support identification (that is non-zero values for the yellow surfaces and zero-values for the blue ones) seems difficult to obtain for* $1 < q < 2$ *but easier for* $q \leq 1$ *(see also the first column of Fig. 10).*

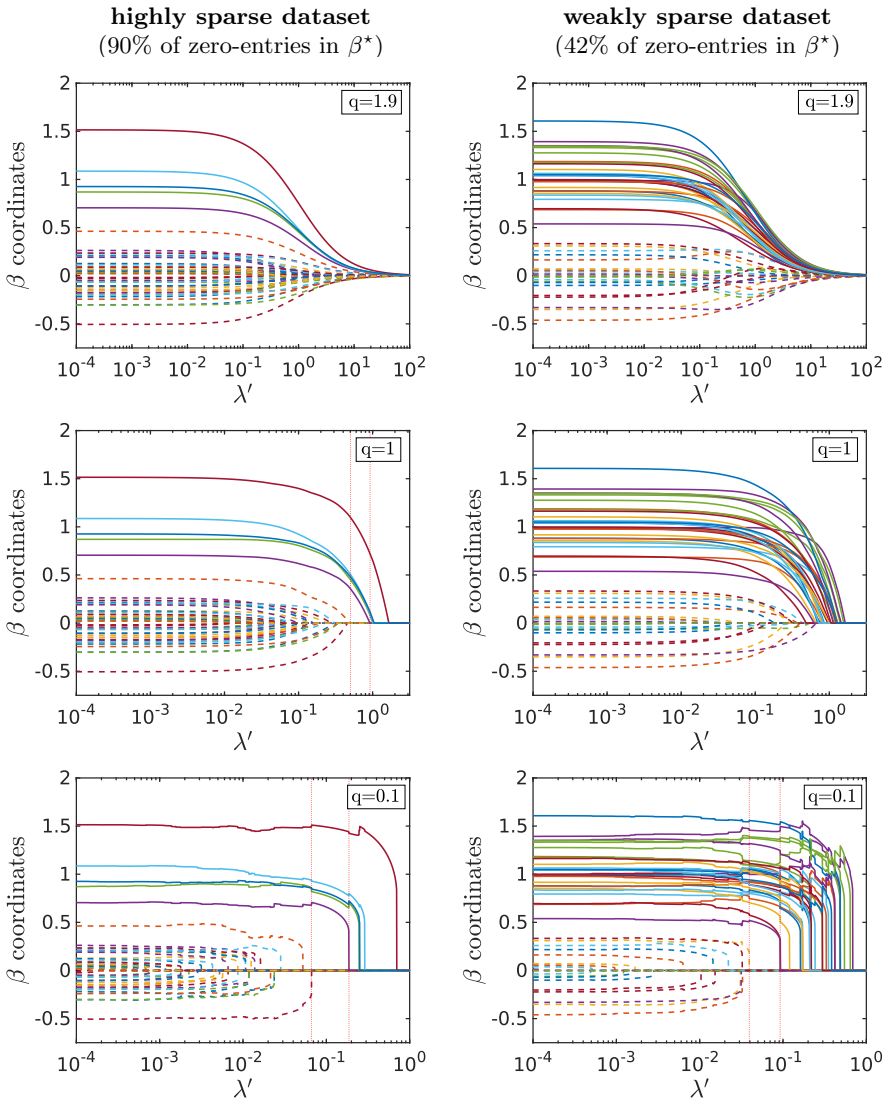


FIG 10. ***Mono-dimensional regularization paths.*** *First column : we display the 1D regularization paths extracted from the 2D regularization path of Fig. 9 for $q = 1.9$ (first row), $q = 1$ (second row) and $q = 0.1$ (last row). We used plain curves to display the five coordinates that are active in the ground truth signal $\beta^\star$, and dashed curves to display the others. We can see that, provided that λ' lies in an appropriate interval (delimited by dotted red lines in the second and last rows), perfect support identification is possible for $q = 1$ and $q = 0.1$, which is not the case for $q = 1.9$. Second column: we reproduced the same experiment using a dataset $(X, \beta^\star, y)$ generated with $\|\beta^\star\|_0 = 21$ (instead of $\|\beta^\star\|_0 = 5$). We can see that perfect support identification is no longer possible for this dataset when $q = 1$ but remains possible for $q = 0.1$.*

The first column of Fig. 10 illustrate how ℓ^q penalization, for $0 < q \leq 1$, can provide efficient support identification (i.e., provide an estimate with same support as the ground thruth signal $\beta^\star$) provided that the λ' parameter lies into an approriate range. This observation was confirmed by many simulations operated in the same conditions (that is, using $n = 300$, $p = 50$ and a ground truth $\beta^\star$ with an average of 90% of zero-valued entries). As surprising as it might seem, the ability of the convex ℓ^1 regularizer to perform as well as the nonconvex ℓ^0 regularizer in terms of support identification (and under sufficient conditions on X, $\beta^\star$ and y) is in fact a flagship result of compressed sensing [13, 25]. In this situation, convex ℓ^1 minimization can be preferred to nonconvex (and in general NP-hard) ℓ^0 minimization, or to its ℓ^q (for $0 < q < 1$) or log-square based approximations, because powerful algorithms can be used in the first case. Indeed, despite the nondifferentiability of the ℓ^1 norm, the minimization of ℓ^1 regularized energies can be efficiently handled using modern proximal algorithms based on Legendre-Fenchel duality [16, 43], such as the celebrated Chambolle-Pock Algorithm [6], or its recent generalization [7] which can efficiently benefit from the presence of terms with Lipschitz gradient (e.g., a quadratic least squares term) in the energy to be minimized (see also closely related algorithms in [11, 47, 14]). Those algorithms are fast and come with strong mathematical garantees regarding the convergence of the sequence of their iterates toward a minimizer of the targeted energy (which is not the case of the AR scheme). However, ℓ^1 minimization may also fail to provide efficient support identification. This typically occurs when dealing with datasets generated from weakly sparse signals $\beta^\star$, as illustrated in the second column of Fig. 10. In this situation, ℓ^q minimization with $0 < q < 1$ should be considered and the AR Algorithm (and more generally the IRLS algorithms) definitely provides an efficient scheme to address those nonconvex problems.

5. Extension to ℓ^q constrained selection

5.1. Principles and motivations

In previous sections, we mostly focused on the ℓ^q regularized problem (3) as an approximation of the $\mathcal{L}_0$ penalized problem (1) that we write again below for the reader convenience:

$$\underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; C(\beta) + \lambda \, \mathcal{L}_0(\beta) \, . \tag{61}$$

The parameter $\lambda > 0$ involved in (61) controls the relative importance of $\mathcal{L}_0$ with respect to C in the minimization process. Since the $\mathcal{L}_0$ regularizer promotes sparse signals, as λ increases, we expect the sparsity of the minimizers of (61) to increase as well. Nevertheless, predicting the sparsity level (that is, the number of nonzero entries) of the solutions of (61) depending on the calibration of λ is in general not possible.

In situations when we have insights about the sparsity level of the signal to compute, or when we need to precisely control the sparsity level of the signal to compute, we may prefer to address the constrained problem of computing

$$\widetilde{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; C(\beta) \quad \text{subject to} \quad \mathcal{L}_0(\beta) \leq t \tag{62}$$

for a given $t \geq 0$ which represents the maximal sparsity level allowed for the signal $\widetilde{\beta}$. In practice, this t parameter plays in (62) a similar role to that of parameter λ in (61) but with inverse variation (the larger t is, the less sparsity is promoted in (62), while the inverse phenomenon occurs with λ in (61)). In fact, (61) can be interpreted as a relaxed version of the constrained problem (62). Similarly, the cost function involved in (61) can be interpreted as the *Lagrangian* (see for instance [4, Chapter 5]) associated to the constrained problem (62). When the functionals involved in the minimization problems (cost functions and inequality constraints) are convex and differentiable, the so-called Karush-Kuhn-Tucker conditions can be used to characterize the solutions of the constrained problem and provide a form of equivalence between the constrained and relaxed problems (see [4, Section 5.5.3]). However, problems (61) and (62) involve nonconvex and nondifferentiable functions, so that formal equivalence between those two problems is not ensured here [35, 53]. Apart from those considerations, a practical advantage of the constrained formulation (62) is that the parameter t has a much more tangible interpretation than the parameter λ involved in (61). Consequently, the calibration of t may be noticeably easier than that of λ in practical applications.

For sure, the constrained problem (62) remains very challenging due to the presence of the nonconvex and nondifferentiable $\mathcal{L}_0$ term defining the constraint set. For that reason, we will use again the ℓ^q norm as an approximation of the $\mathcal{L}_0$ penalty, leading to the ℓ^q constrained problem of computing

$$\widetilde{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; C(\beta) \quad \text{subject to} \quad \|\beta\|_q^q \leq t \, . \tag{63}$$

In this framework, we are going to show how the variational formulation of the ℓ^q norm presented in Proposition 1 can be used to design an alternating minimization scheme for (63). As we shall see, the structure and some properties of this scheme will be similar to that of the AR scheme.

5.2. An alternating minimization scheme and its properties

From now, let us consider $q > 0$, a sparsity level $t \geq 0$, and let us focus on the ℓ^q constrained problem (63). Notice that, when the cost function C is continuous, since the constraint set

$$\mathscr{B}_q(t^{\frac{1}{q}}) = \left\{ \beta \in \mathbb{R}^p \, , \; \|\beta\|_q \leq t^{\frac{1}{q}} \right\} \tag{64}$$

is compact (as a closed and bounded subset of $\mathbb{R}^p$), the existence of solutions for (63) is ensured. The continuity of C is not a necessary condition to ensure the

existence of solutions for (63), but it happens to be satisfied in most practical applications (e.g., when $C : \beta \mapsto \|y - X\beta\|_2^2$, as in Section 4).

In order to figure out a numerical scheme for solving (63), let us establish a slight variant of Proposition 1 involving the generalized ratio function r defined in (18).

Proposition 6. *For all $\beta \in \mathbb{R}^p$, for all $q > 0$ and for all $\nu > q$, we have*

$$\|\beta\|_q^q = \min_{\eta \in \mathbb{R}_+^p} \left(\mathscr{L}_q^\nu(\beta, \eta) := \frac{q}{\nu} \cdot \sum_{j=1}^p r(|\beta_j|^\nu, \eta_j) + \frac{\nu - q}{\nu} \cdot \sum_{j=1}^p \eta_j^{\frac{q}{\nu - q}} \right) \tag{65}$$

and the minimum is attained at $\eta = |\beta|^{\nu - q}$.

Proof. Let us show this result in the unidimensional case ($p = 1$). Let $\beta \in \mathbb{R}$. If $\beta = 0$, for all $\eta > 0$, we have $\mathscr{L}_q^\nu(|\beta|^\nu, \eta) > \mathscr{L}_q^\nu(|\beta|^\nu, 0) = 0$. Therefore, (65) is true for $\beta = 0$ and the minimum is attained at $\eta = 0 = |\beta|^{\nu - q}$ as announced. Now, if $\beta \neq 0$, then, for all $\eta > 0$, we have $\mathscr{L}_q^\nu(|\beta|^\nu, \eta) < \mathscr{L}_q^\nu(|\beta|^\nu, 0) = +\infty$. Therefore, the minimum in (65) can be restricted to $\eta > 0$ so that (65) is nothing more than (8), and the minimum is indeed attained at $\eta = |\beta|^{\nu - q}$ thanks to Proposition 1. This ends the proof of Proposition 6 for $p = 1$. Finally, this result can be extended to higher dimension ($p \geq 1$) for any $\beta \in \mathbb{R}^p$ thanks to the additive separability of $\eta \mapsto \mathscr{L}_q^\nu(\beta, \eta)$ with respect to $(\eta_1, \eta_2, \ldots, \eta_p)$. □

Thanks to Proposition 6, the ℓ^q constrained problem (63) is equivalent to computing

$$\widetilde{\beta} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} C(\beta) \quad \text{subject to} \quad \min_{\eta \in \mathbb{R}_+^p} \mathscr{L}_q^\nu(\beta, \eta) \leq t\,. \tag{66}$$

An interesting alternating minimization strategy to address (66) consists in setting $\beta^{(0)} \in \mathscr{B}_q(t)$ and iterating for $k \geq 0$,

$$\begin{cases} \eta^{(k+1)} \in \operatorname*{argmin}_{\eta \in \mathbb{R}_+^p} \mathscr{L}_q^\nu(\beta^{(k)}, \eta) & \text{(67a)} \\ \beta^{(k+1)} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} C(\beta) \quad \text{subject to} \quad \mathscr{L}_q^\nu(\beta, \eta^{(k+1)}) \leq t\,. & \text{(67b)} \end{cases}$$

Thanks to Proposition 6, (67a) admits $\eta^{(k+1)} = |\beta^{(k)}|^{\nu - q}$ as unique solution. As we did before, we implicitly assume here that (67b) admits some solutions so that this scheme iteration is well defined. One can easily check that the constraint set in (67b) is closed and bounded, and thus, compact. Therefore, the existence of solutions for (67b) is ensured when C is continuous. In the following, we set

$$\forall \eta \in \mathbb{R}_+^p\,, \quad \mathscr{E}_q^\nu(\eta, t) = \left\{ \beta \in \mathbb{R}^p\,,\ \mathscr{L}_q^\nu(\beta, \eta) \leq t \right\} \tag{68}$$

so that the constraint $\mathscr{L}_q^\nu(\beta, \eta) \leq t$ is equivalent to $\beta \in \mathscr{E}_q^\nu(\eta, t)$ and (67) boils

down to setting $\beta^{(0)} \in \mathscr{B}_q(t^{\frac{1}{q}})$ and iterating, for $k \geq 0$,

$$\begin{cases} \eta^{(k+1)} = |\beta^{(k)}|^{\nu-q} & \text{(69a)} \\ \beta^{(k+1)} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}\ C(\beta) \quad \text{subject to} \quad \beta \in \mathscr{E}_q^{\nu}(\eta^{(k+1)}, t)\,. & \text{(69b)} \end{cases}$$

Now let us show that the iterates $\left(\beta^{(k)}\right)_{k\geq 0}$ generated using (69) satisfy the energy decrease property.

Proposition 7. *The iterates $(\beta^{(k)})_{k\geq 0}$ generated using (69) are all included in $\mathscr{B}_q(t)$ and the sequence $\left(C(\beta^{(k)})\right)_{k\geq 0}$ is decreasing, i.e.,*

$$\forall k \geq 0\,, \quad \beta^{(k)} \in \mathscr{B}_q(t^{\frac{1}{q}}) \quad \textit{and} \quad C(\beta^{(k+1)}) \leq C(\beta^{(k)})\,.$$

Besides, if C is bounded from below, then the sequence $\left(C(\beta^{(k)})\right)_{k\geq 0}$ converges.

In order to prove Proposition 7, we need the following Lemma.

Lemma 3. *For all $k \geq 0$, the set $\mathscr{E}_q^{\nu}(\eta^{(k+1)}, t)$ is a nonempty subset of $\mathscr{B}_q(t^{\frac{1}{q}})$ that contains $\beta^{(k)}$, i.e.,*

$$\forall k \geq 0\,, \quad \mathscr{E}_q^{\nu}(\eta^{(k+1)}, t) \subset \mathscr{B}_q(t^{\frac{1}{q}}) \quad \textit{and} \quad \beta^{(k)} \in \mathscr{E}_q^{\nu}(\eta^{(k+1)}, t)\,. \tag{70}$$

Proof. Let $k \geq 0$ and $\beta \in \mathscr{E}_q^{\nu}(\eta^{(k+1)}, t)$, thanks to Proposition 6, we have

$$\|\beta\|_q^q = \min_{\eta\in\mathbb{R}_+^p} \mathscr{L}_q^{\nu}(\beta, \eta) \leq \mathscr{L}_q^{\nu}(\beta, \eta^{(k+1)}) \leq t\,,$$

the right-hand side inequality $\mathscr{L}_q^{\nu}(\beta, \eta^{(k+1)}) \leq t$ coming from $\beta \in \mathscr{E}_q^{\nu}(\eta^{(k+1)}, t)$. It follows that $\beta \in \mathscr{B}_q(t^{\frac{1}{q}})$, showing the inclusion $\mathscr{E}_q^{\nu}(\eta^{(k+1)}, t) \subset \mathscr{B}_q(t^{\frac{1}{q}})$. Therefore, (69b) necessarily generates $\beta^{(k+1)} \in \mathscr{B}_q(t^{\frac{1}{q}})$. Since we initialized the scheme using $\beta^{(0)} \in \mathscr{B}_q(t^{\frac{1}{q}})$, all the sequence $(\beta^{(\ell)})_{\ell\geq 0}$ lies in $\mathscr{B}_q(t^{\frac{1}{q}})$. In particular, we have $\beta^{(k)} \in \mathscr{B}_q(t^{\frac{1}{q}})$, and thus, using again Proposition 6, we have

$$t \geq \|\beta^{(k)}\|_q^q = \min_{\eta\in\mathbb{R}_+^p} \mathscr{L}_q^{\nu}(\beta^{(k)}, \eta) = \mathscr{L}_q^{\nu}(\beta^{(k)}, |\beta^{(k)}|^{\nu-q}) = \mathscr{L}_q^{\nu}(\beta^{(k)}, \eta^{(k+1)})$$

showing that $\beta^{(k)} \in \mathscr{E}_q^{\nu}(\eta^{(k+1)}, t)$, which ends the proof. □

Proof of Proposition 7. The inclusion in $\mathscr{B}_q(t^{\frac{1}{q}})$ of all the iterates $\left(\beta^{(k)}\right)_{k\geq 0}$ was already established in the proof of Lemma 3. Let $k \geq 0$. From Lemma 3, we have $\beta^{(k)} \in \mathscr{E}_q^{\nu}(\eta^{(k+1)}, t)$. Therefore, (69b) necessarily generates an iterate $\beta^{(k+1)}$ such as $C(\beta^{(k+1)}) \leq C(\beta^{(k)})$, as announced. □

Remark 5. *Assuming that, for $k \geq 0$, we can find $j \in \{1, 2, \ldots, p\}$ such as $\beta_j^{(k)} = 0$, then (69a) yields $\eta_j^{(k+1)} = 0$ and one can easily check that we have*

$$\forall \beta \in \mathbb{R}^p\,, \quad \mathscr{L}_q^{\nu}(\beta, \eta^{(k+1)}) \leq t \ \Rightarrow\ \beta_j = 0$$

since $\mathscr{L}_q^\nu(\beta, \eta^{(k+1)}) = +\infty$ *when* $\beta_j \neq 0$. *Consequently, the constraint set* $\mathscr{E}_q^\nu(\eta^{(k+1)}, t)$ *involved in* (69b) *imposes* $\beta_j^{(k+1)} = 0$.

Remark 5 points out that, similarly to Scheme (19), the vanishing coordinates of an iterate $\beta^{(k)}$ generated by (69) will remain vanishing in all later iterations. As done in the AR and IRLS schemes, we can avoid the persitence of vanishing coordinates by introducing a parameter $\delta > 0$ within the ℓ^q penalty term. More precisely, instead of considering (63), one can consider the problem of finding

$$\widetilde{\beta} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} C(\beta) \quad \text{subject to} \quad \| |\beta|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} \leq t \,. \tag{71}$$

For the sake of completeness, we explain in Appendix D how (71) can be addressed by setting $\beta^{(0)}$ such that $\| |\beta^{(0)}|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} \leq t$, and iterating, for $k \geq 0$,

$$\begin{cases} \eta^{(k+1)} = \left(|\beta^{(k)}|^\nu + \delta^\nu \right)^{\frac{\nu - q}{\nu}} & \text{(72a)} \\ \beta^{(k+1)} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} C(\beta) \quad \text{subject to} \quad \beta \in \mathscr{E}_q^{\nu,\delta}(\eta^{(k+1)}, t) \,, & \text{(72b)} \end{cases}$$

where, for all $\eta \in (\mathbb{R}_+^*)^p$, we have set

$$\mathscr{E}_q^{\nu,\delta}(\eta, t) = \left\{ \beta \in \mathbb{R}^p \,,\ \frac{q}{\nu} \cdot \sum_{j=1}^p \frac{|\beta_j|^\nu + \delta^\nu}{\eta_j} + \frac{\nu - q}{\nu} \cdot \sum_{j=1}^p \eta_j^{\frac{q}{\nu - q}} \leq t \right\} . \tag{73}$$

This scheme can be viewed as another (constrained) variant of the AR or IRLS schemes. We easily get the following result.

Proposition 8. *Given* $\beta^{(0)}$ *such as* $\| |\beta^{(0)}|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} \leq t$, *the sequence* $(\beta^{(k)})_{k \geq 0}$ *generated using* (72) *satisfies*

$$\forall k \geq 0 \,, \quad \| |\beta^{(k)}|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} \leq t \quad \textit{and} \quad C(\beta^{(k+1)}) \leq C(\beta^{(k)}) \,.$$

Proof. The proof is given in Appendix D. □

We would like to point out that Scheme (72) has been proposed and studied very recently in [52] in the case where $C : \beta \mapsto \|y - \beta\|_2^2$ and using a dynamic updating strategy for parameter δ along the scheme iterations. Some other closely related schemes were also recently studied in [50, 51].

5.3. Numerical implementation

In this section, given a matrix $X \in \mathbb{R}^{n \times p}$ and a vector $y \in \mathbb{R}^n$, we focus on the numerical implementation of the scheme (69) in the case where $C : \beta \mapsto \|y - X\beta\|_2^2$. Since the η update step (69a) is explicit in this scheme, our main goal is to achieve the numerical computation of (69b), that is, for $k \geq 0$,

$$\beta^{(k+1)} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \quad \text{subject to} \quad \beta \in \mathscr{E}_q^\nu(\eta^{(k+1)}, t) \tag{74}$$

given $\eta^{(k+1)} = |\beta^{(k)}|^{\nu-q}$ computed from the previous iterate $\beta^{(k)}$. In order to compute $\beta^{(k+1)}$, we need the following Lemma.

Lemma 4. *Given $\widetilde{\beta} \in \mathscr{B}_q(t^{\frac{1}{q}})$ and setting $\widetilde{\eta} = |\widetilde{\beta}|^{\nu-q}$, we have*

$$\mathscr{E}_q^{\nu}(\widetilde{\eta}, t) = \mathscr{D}(\widetilde{\eta}^{\frac{1}{\nu}})\, \mathscr{B}_{\nu}(z_q^{\nu}(\widetilde{\eta}, t)) := \left\{ D(\widetilde{\eta}^{\frac{1}{\nu}})\, \widetilde{\xi}\,,\ \widetilde{\xi} \in \mathscr{B}_{\nu}(z_q^{\nu}(\widetilde{\eta}, t)) \right\} \tag{75}$$

where

$$z_q^{\nu}(\widetilde{\eta}, t) = \left[\frac{\nu}{q} \left(t - \frac{\nu - q}{\nu} \sum_{j=1}^{p} \widetilde{\eta}_j^{\frac{q}{\nu-q}} \right) \right]^{\frac{1}{\nu}} \tag{76}$$

and where we recall that $\mathscr{D}(\widetilde{\eta}^{\frac{1}{\nu}}) = \mathrm{diag}(\widetilde{\eta}_1^{\frac{1}{\nu}}, \widetilde{\eta}_2^{\frac{1}{\nu}}, \ldots, \widetilde{\eta}_p^{\frac{1}{\nu}})$ and that $\mathscr{B}_{\nu}(z)$ denotes the ℓ^{ν} ball with zero center and radius z, i.e., $\mathscr{B}_{\nu}(z) = \{\xi \in \mathbb{R}^p\,,\ \|\xi\|_{\nu} \leq z\}$.

Proof. The proof is given in Appendix E. □

Remark 6 (shape of the constraint set)**.** *From (75), we can see that $\mathscr{E}_q^{\nu}(\widetilde{\eta}, t)$ is nothing but a ℓ^{ν} ball (with radius $z_q^{\nu}(\widetilde{\eta}, t)$) dilated along each axis by the diagonal rescaling matrix $\mathscr{D}(\widetilde{\eta}^{\frac{1}{\nu}})$. In particular, $\mathscr{E}_q^{\nu}(\widetilde{\eta}, t)$ is an ellipsoid when $\nu = 2$ and a diamond when $\nu = 1$.*

Corollary 1. *For $k \geq 0$, denoting $\mathscr{D}_{k+1} = \mathscr{D}((\eta^{(k+1)})^{\frac{1}{\nu}})$, the solutions of (74) are the vectors $\beta^{(k+1)} = \mathscr{D}_{k+1}\, \xi^{(k+1)}$ such as*

$$\xi^{(k+1)} \in \underset{\xi \in \mathbb{R}^p}{\mathrm{argmin}}\ \|y - X\mathscr{D}_{k+1}\, \xi\|_2^2 \quad \textit{subject to} \quad \xi \in \mathscr{B}_{\nu}(z_q^{\nu}(\eta^{(k+1)}, t))\,. \tag{77}$$

This result is a straightforward consequence of Lemma 4 applied to $\widetilde{\beta} = \beta^{(k)}$ which is an element of $\mathscr{B}_q(t^{\frac{1}{q}})$ (see Proposition 7).

Remark 7. *When $\beta^{(k)}$ admits a vanishing coordinate, i.e., $\beta_j^{(k)} = 0$ for a given $j \in \{1, 2, \ldots, p\}$, we have $\eta_j^{(k+1)} = 0$. In this case, the matrix $X\mathscr{D}_{k+1}$ is rank deficient and (77) admits an infinite number of solutions as long as $t > 0$. However, whatever the considered solution $\xi^{(k+1)}$ of (77) in this situation, by setting afterwards $\beta^{(k+1)} = D_{k+1}\, \xi^{(k+1)}$, we necessarily end up with $\beta_j^{(k+1)} = 0$. Therefore, noting J_k the support of $\beta^{(k)}$, i.e.,*

$$J_k = \{j \in \{1, 2, \ldots, p\},\ \beta_j^{(k)} \neq 0\}\,,$$

one may prefer considering instead of (77) the constrained problem

$$\xi^{(k+1)} \in \underset{\xi \in \mathbb{R}^p}{\mathrm{argmin}}\ \|y - X\mathscr{D}_{k+1}\, \xi\|_2^2 \ \ \textit{subject to} \begin{cases} \xi \in \mathscr{B}_{\nu}(z_q^{\nu}(\eta^{(k+1)}, t)) \\ \forall j \notin J_k\,,\ \xi_j = 0 \end{cases} \tag{78}$$

and retrieve again the minimizers of (74) by computing $\beta^{(k+1)} = \mathscr{D}_{k+1}\, \xi^{(k+1)}$. In particular, we can show that the constrained problems (74) and (77) both admit a unique solution if and only if the matrix X_k obtained by removing from X the columns with index $j \notin J_k$ is of full rank.

Thanks to Corollary 1, the scheme (69) is equivalent to setting $\beta^{(0)} \in \mathscr{B}_q(t^{\frac{1}{q}})$ and iterating for $k \geq 0$,

$$\begin{cases} \eta^{(k+1)} = |\beta^{(k)}|^{\nu-q} & \text{(79a)} \\ \mathscr{D}_{k+1} = \mathscr{D}((\eta^{(k+1)})^{\frac{1}{\nu}}) & \text{(79b)} \\ z_{k+1} = \left[\dfrac{\nu}{q}\left(t - \dfrac{\nu-q}{\nu}\displaystyle\sum_{j=1}^{p}\left(\eta_j^{(k+1)}\right)^{\frac{q}{\nu-q}}\right)\right]^{\frac{1}{\nu}} & \text{(79c)} \\ \xi^{(k+1)} \in \underset{\xi\in\mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\mathscr{D}_{k+1}\,\xi\|_2^2 \quad \text{subject to} \quad \xi \in \mathscr{B}_\nu(z_{k+1}) & \text{(79d)} \\ \beta^{(k+1)} = \mathscr{D}_{k+1}\,\xi^{(k+1)}\,. & \text{(79e)} \end{cases}$$

In this scheme, steps (79a), (79b), (79c) and (79e) are explicit. It remains to compute a solution of the constrained problem (79d). Fortunately, this kind of problem can be easily addressed using proximal algorithms. In particular, for any $k \geq 0$, using the dual identity

$$\forall \xi \in \mathbb{R}^p\,, \quad \|y - X\mathscr{D}_{k+1}\,\xi\|_2^2 = \max_{\gamma\in\mathbb{R}^n} \ 2\,\langle X\mathscr{D}_{k+1}\,\xi, \gamma\rangle - \|\gamma + y\|_2^2 + \|y\|_2^2$$

in (79d) and removing the constant term $\|y\|_2^2$ that does not change the argmin, we can reformulate (79d) into the primal-dual problem of computing

$$\xi^{(k+1)} \in \underset{\xi\in\mathbb{R}^p}{\operatorname{argmin}} \max_{\gamma\in\mathbb{R}^n} \ 2\,\langle X\mathscr{D}_{k+1}\,\xi, \gamma\rangle - \|\gamma + y\|_2^2 \quad \text{s.t.} \quad \xi \in \mathscr{B}_\nu(z_{k+1})\,. \tag{80}$$

The primal-dual problem (80) can be efficiently handled using modern proximal algorithms, such as those proposed in [6, 7, 14, 11]. For instance, Chambolle-Pock Algorithm [6] applied to (80) boils down to setting $(\xi_0, \overline{\xi}_0, \gamma_0) \in \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^n$, $\tau > 0$, $\sigma > 0$, and to iterating for $\ell \geq 0$,

$$\begin{cases} \gamma_{\ell+1} = \dfrac{\gamma_\ell + 2\sigma X\mathscr{D}_{k+1}\overline{\xi}_\ell - 2\sigma y}{1 + 2\sigma} & \text{(81a)} \\ \xi_{\ell+1} = \Pi_{\mathscr{B}_\nu(z_{k+1})}\left(\xi_\ell - 2\tau\mathscr{D}_{k+1}X^t\gamma_{\ell+1}\right) & \text{(81b)} \\ \overline{\xi}_{\ell+1} = 2\,\xi_{\ell+1} - \xi_\ell & \text{(81c)} \end{cases}$$

denoting by $\Pi_{\mathscr{B}_\nu(z_{k+1})}$ the orthogonal projection onto the convex set $\mathscr{B}_\nu(z_{k+1})$. The convergence of (81) towards a solution of (80) (and thus, a solution of (79d)) is ensured as long as the primal and dual steps (τ, σ) satisfy $\tau\sigma < |||2X\mathscr{D}_{k+1}|||^2$, denoting by $|||\cdot|||$ the ℓ^2 induced norm (see [6, Theorem 1]). Therefore, from any upper bound $L_{k+1} \geq |||2X\mathscr{D}_{k+1}|||$, one can, for instance, set $\tau = \sigma = 0.99/L_{k+1}$ to ensure the convergence of (81) toward a solution of (79d). In practice, the closer to $|||2X\mathscr{D}_{k+1}|||$ the upper bound L_{k+1} is, the faster the convergence is. Notice that (81b) involves a projection onto the convex ball $\mathscr{B}_\nu(z_{k+1})$. When $\nu = 2$, this projection is explicit, and we simply have

$$\forall \xi \in \mathbb{R}^p\,, \ \forall z \geq 0\,, \quad \Pi_{\mathscr{B}_2(z)}(\xi) = \frac{z\,\xi}{\max\left(z, \|\xi\|_2\right)}\,. \tag{82}$$

When $\nu = 1$, the projection onto the ℓ^1 ball $\mathscr{B}_1(z_{k+1})$ can be efficiently computed in $\mathscr{O}(p \log p)$ operations using the method described in [15] and that we summarize in Algorithm 4 for the sake of completeness. The numerical evaluation of (79d) using Chambolle-Pock Algorithm is summarized in Algorithm 5. Finally, the computation of (63) with $C : \beta \mapsto \|y - X\beta\|_2^2$ using Scheme (79) is summarized in Algorithm 6.

Remark 8. *Scheme* (79) *can be easily generalized to address the approached (or* δ *regularized)* ℓ^q *constrained problem* (71) *through the alternating scheme* (72). *For any* $\delta \geq 0$, *one simply needs to replace* (79a) *by* (72a) *and* (79c) *by*

$$z_{k+1} = \left[\frac{\nu}{q} \left(t - \frac{q}{\nu} \sum_{j=1}^{p} r(\delta^\nu, \eta_j^{(k+1)}) - \frac{\nu - q}{\nu} \sum_{j=1}^{p} \left(\eta_j^{(k+1)} \right)^{\frac{q}{\nu - q}} \right) \right]^{\frac{1}{\nu}} . \tag{83}$$

When $\delta = 0$, (83) *is exactly the same as* (79c), *since* $r(0, y) = 0$ *for any* $y \in \mathbb{R}$. *Those modifications can be easily implemented in Algorithm 6 (line 5 and line 7) to expand its scope to* (71) *with* $\delta \geq 0$.

Algorithm 4: orthogonal projection onto $\mathscr{B}_1(z)$ [15].

Inputs : a vector $\xi \in \mathbb{R}^p$ and a scalar $z \geq 0$.

Output: $\Pi_{\mathscr{B}_1(z)}(\xi)$, the orthogonal projection of ξ onto the ℓ^1 ball with radius z.

1 **if** $\|\xi\|_1 > z$ **then**

2 $\quad \mu \leftarrow (|\xi_1|, |\xi_2|, \ldots, |\xi_p|)$

3 $\quad$ sort μ in descending order $(\mu_1 \geq \mu_2 \geq \cdots \geq \mu_p)$

4 $\quad \rho \leftarrow \max \left\{ j \in \{1, 2, \ldots, p\}, \ \mu_j - \frac{1}{j} \left(\sum_{r=1}^{j} \mu_r - z \right) > 0 \right\}$

5 $\quad \theta \leftarrow \frac{1}{\rho} \left(\sum_{j=1}^{\rho} \mu_j - z \right)$

6 $\quad w \leftarrow \max(|\xi| - \theta, 0) \cdot \operatorname{sign}(\xi)$ `// coordinate-wise operations (see below)`

7 **else** $w \leftarrow \xi$

8 **return** w

In line 6, for all $j \in \{1, 2, \ldots, p\}$, we set $w_j \leftarrow \max(|\xi_j| - \theta, 0) \cdot \operatorname{sign}(\xi_j)$, where $\operatorname{sign}(t) = t/|t|$ when $t \neq 0$ and $\operatorname{sign}(0) = 0$.

5.4. Experiments

In this section, we will illustrate the behavior of Algorithm 6 over synthetic datasets. First of all, we focus on the the two-dimensional framework ($p = 2$) since this simplified setting opens the door for easy understanding of the algorithmic procedure. In Fig. 11 we display and comment the first iterations of Algorithm 6 used with $\nu = 2$ over a synthetical dataset.

Algorithm 5: solver for Problem (79d) (`primaldual` module) [6].

Input : a matrix $M \in \mathcal{M}_{n,p}(\mathbb{R})$ (equal to the matrix product $X\mathscr{D}_{k+1}$ involved in (79d)), the vector $y \in \mathbb{R}^n$, the parameter $\nu > 0$ and the ℓ^ν ball radius $z_{k+1} \geq 0$ involved in (79d), some initial values for the primal and dual variables $(\xi_0, \overline{\xi}_0, \gamma_0) \in \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^n$ of the scheme (81), two steps $\tau > 0$ and $\sigma > 0$ such that $\tau\sigma < |||2M|||^2$ and a tolerance parameter $\varepsilon > 0$.

1 $\ell \leftarrow 0$
2 **repeat**
3 $\quad \gamma_{\ell+1} \leftarrow \left(\gamma_\ell + 2\sigma M\overline{\xi}_\ell - 2\sigma y\right)/(1+2\sigma)$
4 $\quad \xi_{\ell+1} \leftarrow \Pi_{\mathscr{B}_\nu(z_{k+1})}\left(\xi_\ell - 2\tau M^t\gamma_{\ell+1}\right)$ `// see note below.`
5 $\quad \overline{\xi}_{\ell+1} \leftarrow 2\,\xi_{\ell+1} - \xi_\ell$
6 $\quad \ell \leftarrow \ell + 1$
7 **until** $\|y - M\xi_\ell\|_2^2 - \|y - M\xi_{\ell-1}\|_2^2 \leq \varepsilon \cdot \|y - M\xi_{\ell-1}\|_2^2$
8 **return** (ξ_ℓ, γ_ℓ)

Line 4: the projection over $\mathscr{B}_\nu(z_{k+1})$ can be done using (82) when $\nu = 2$ or using Algorithm 4 when $\nu = 1$. Line 7: we used again an energy-based stopping criterion but other kind of criteria (e.g., based on the so-called duality gap [6]) can also be considered. Line 8: ξ_ℓ is the estimated minimizer of $\xi \mapsto \|y - M\xi\|_2^2$ over $\mathscr{B}_\nu(z_{k+1})$, the dual variable γ_ℓ is also returned to be used as a dual initializer in the next iteration of the scheme (79).

Algorithm 6: ℓ^q constrained minimization using Scheme (79)

Input : Some parameters $t \geq 0$, $q > 0$, $\nu > q$, a matrix $X \in \mathcal{M}_{n,p}(\mathbb{R})$, an initial guess $\beta^{(0)} \in \mathscr{B}_q(t^{\frac{1}{q}})$ and a tolerance parameter $\varepsilon > 0$.

Output: an estimate of a local minimizer of $\mathtt{E} : \beta \mapsto \|y - X\beta\|_2^2$ over $\mathscr{B}_q(t^{\frac{1}{q}})$

1 $k \leftarrow 0$
2 $\gamma^{(0)} \leftarrow 0$ `// zero vector in` $\mathbb{R}^n$
3 $\xi^{(0)} \leftarrow 0$ `// zero vector in` $\mathbb{R}^p$
4 **repeat**
5 $\quad \eta^{(k+1)} \leftarrow |\beta^{(k)}|^{\nu-q}$
6 $\quad \mathscr{D}_{k+1} \leftarrow \mathscr{D}((\eta^{(k+1)})^{\frac{1}{\nu}})$
7 $\quad z_{k+1} \leftarrow \left[\frac{\nu}{q}\left(t - \frac{\nu-q}{\nu}\sum_{j=1}^p \left(\eta_j^{(k+1)}\right)^{\frac{q}{\nu-q}}\right)\right]^{\frac{1}{\nu}}$
8 $\quad (\tau, \sigma) \leftarrow$ two positive parameters such as $\tau\sigma < |||2X\mathscr{D}_{k+1}|||^2$
9 $\quad (\xi^{(k+1)}, \gamma^{(k+1)}) \leftarrow \mathtt{primaldual}(X\mathscr{D}_{k+1}, y, \nu, z_{k+1}, (\xi^{(k)}, \xi^{(k)}, \gamma^{(k)}), \tau, \sigma, \varepsilon)$
10 $\quad \beta^{(k+1)} \leftarrow \mathscr{D}_{k+1}\,\xi^{(k+1)}$
11 $\quad k \leftarrow k + 1$
12 **until** $E(\beta^{(k)}) - E(\beta^{(k-1)}) \leq \varepsilon \cdot E(\beta^{(k-1)})$
13 **return** $\beta^{(k)}$

Line 8 : given an upper bound $L_X \geq |||X|||$, one can use $|||2X\mathscr{D}_{k+1}||| \leq L_{k+1} := 2 \cdot L_X \cdot \|\eta^{(k+1)}\|_\infty^{1/\nu}$. Thus, when $L_{k+1} > 0$, one can set $\tau = \sigma = 0.99/L_{k+1}$ to ensures that $\tau\sigma < |||2X\mathscr{D}_{k+1}|||^2$ is fulfilled. Otherwise, when $L_{k+1} = 0$ (and assuming that $X \neq 0$), we have $\eta^{(k+1)} = 0$, meanings that $\beta^{(k)} = 0$. In this case, the scheme iterations can be immediately stopped since we will necessarily get $\beta^{(\ell)} = 0$ in all later iterations $\ell > k$. Remark that in the simulation scheme described in Section 4.3.1, the unitary ℓ^2 norm of the columns of X ensures that $|||X||| \leq \sqrt{p}$, so that one can use $L_{k+1} = 2\sqrt{p}\|\eta^{(k+1)}\|_\infty^{1/\nu}$ in this case.

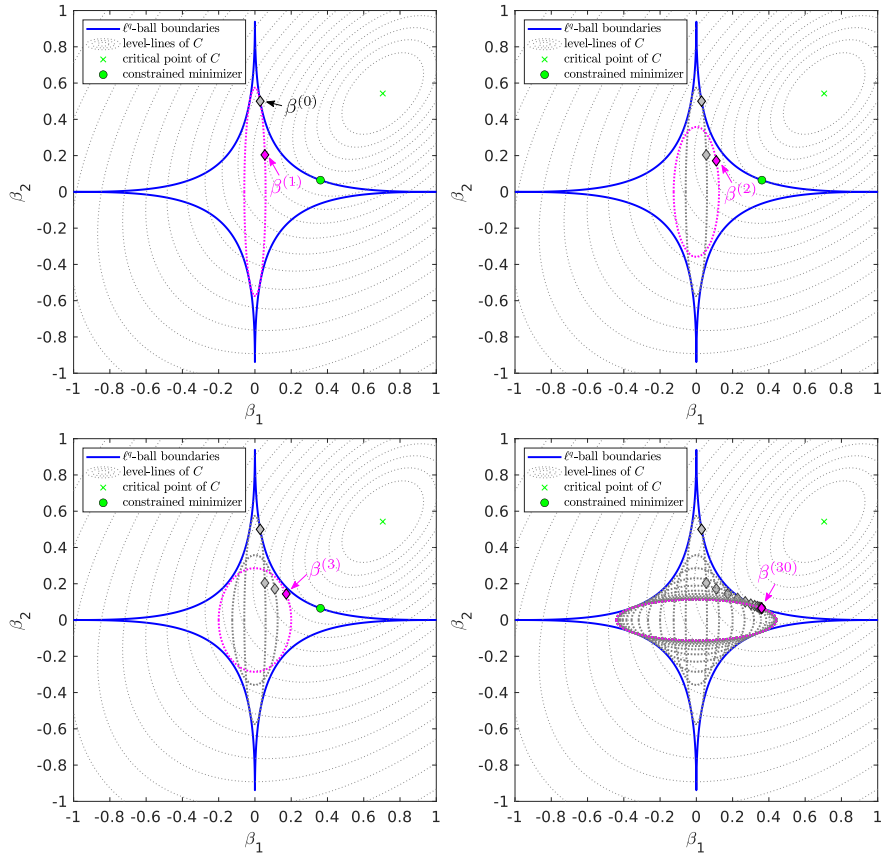


FIG 11. ***ℓ^q constrained minimization using Algorithm 6 with $\nu = 2$.*** *In this experiment, we consider a synthetic dataset $(X, \beta^\star, y)$ with dimensions $(n = 5, p = 2)$ generated using the simulation scheme described in Section 4.3.1 excepting that $\beta^\star$ was randomly sampled according to a uniform distribution in $[-1, 1]^2$, leading here to $\beta^\star \approx (0.94, 0.62)^t$. We used Algorithm 6 with $\nu = 2$, $q = 0.7$ and $t = 1$ to process this dataset and we display here its first 30 iterates. In all those graphs, the blue curve represents the boundaries of the constraint set involved in (63), that is, the ℓ^q ball with radius $t^{1/q} = 1$. Some level lines of the quadratic function $C : \beta \mapsto \|y - X\beta\|_2^2$ are displayed using lightgray dotted-lines, while the minimizer of C over $\mathscr{B}_q(t^{1/q})$ is represented using a green disk.* ***Top-left****: a gray diamond mark indicates the scheme initializer $\beta^{(0)}$ and a pink dotted line indicates the boundaries of the first constraint set $\mathscr{E}_q^\nu(\eta^{(1)}, t)$ appearing in (69b). The latter is an elipsoid because of the setting $\nu = 2$. The minimization of C over $\mathcal{E}_q^\nu(\eta^{(1)}, t)$ leads to the next iterate $\beta^{(1)}$, which is indicated using a pink diamond mark.* ***Top-right****: the next iteration yields $\eta^{(2)} = |\beta^{(1)}|^{\nu-q}$ and the pink dotted line represents the boundaries of the updated constraint set $\mathscr{E}_q^\nu(\eta^{(2)}, t)$. Minimizing C over this set yields the next iterate $\beta^{(2)}$ displayed with a pink diamond mark (previous iterates or constraint sets are displayed in gray).* ***Bottom-left*** *and* ***bottom-right****: third and thirtieth scheme iterations (we use the same displaying rules as described above). We can observe here the convergence of the scheme towards the green disk, that is, towards the (here unique) solution of (63).*

The same dataset as that used in Fig. 11 was processed using Algorithm 6 with the setting $\nu = 1$. The scheme iteration process is illustrated and commented in Fig. 12.

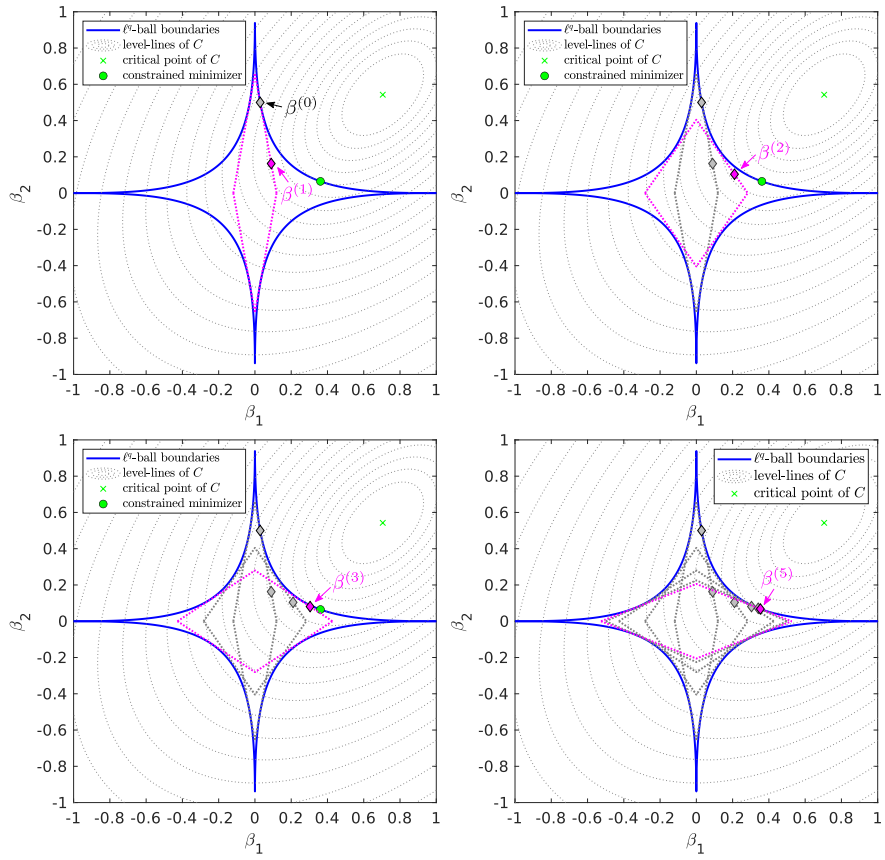


FIG 12. ***ℓ^q constrained minimization using Algorithm 6 with $\nu = 1$).*** *We display (with the same displaying rules as in Fig.11) the first five iterates generated by Algorithm 6 with $\nu = 1$ applied to the same dataset as that considered in Fig. 11. While in Fig. 11 the setting $\nu = 2$ yielded elipsoid shaped constraint sets $\mathcal{E}_q^\nu(\eta^{(k+1)}, t)$, the setting $\nu = 1$ yields diamond shaped constraint sets. We can see that, as in Fig. 11, the numerical scheme seems to sucessfully converges toward the solution of (63) (green disk). We can also notice that, over this dataset, the setting $\nu = 1$ yields a faster convergence than the setting $\nu = 2$ (see Fig. 11). Our experiments in larger dimensions (not shown here) tend to confirm that $\nu = 1$ yields a significantly higher convergence rate than the setting $\nu = 2$.*

In the experiments presented in Fig. 11 and Fig. 12, We were able to efficiently estimate the solution of the nonconvex problem (63) using Algorithm 6. However, the existence of local minima for problem (63) makes the latter sensitive to the choice of the initializer, as we already pointed out for the AR Algorithm. An example of convergence of Algorithm 6 towards a local minimum is presented in Fig. 13.

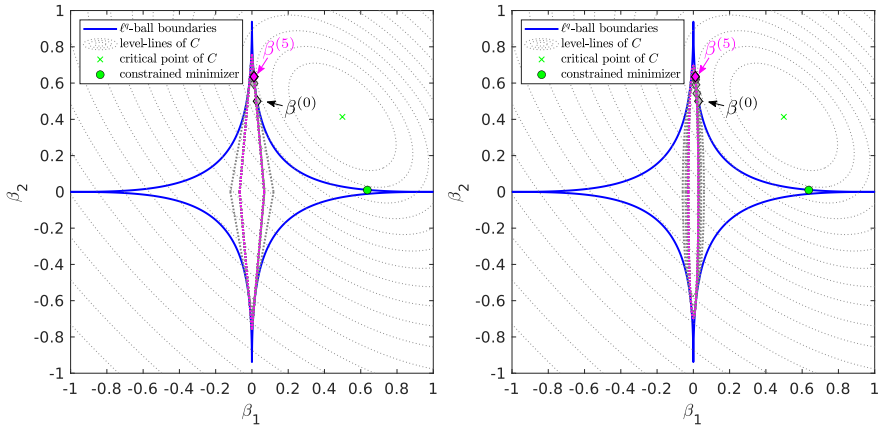
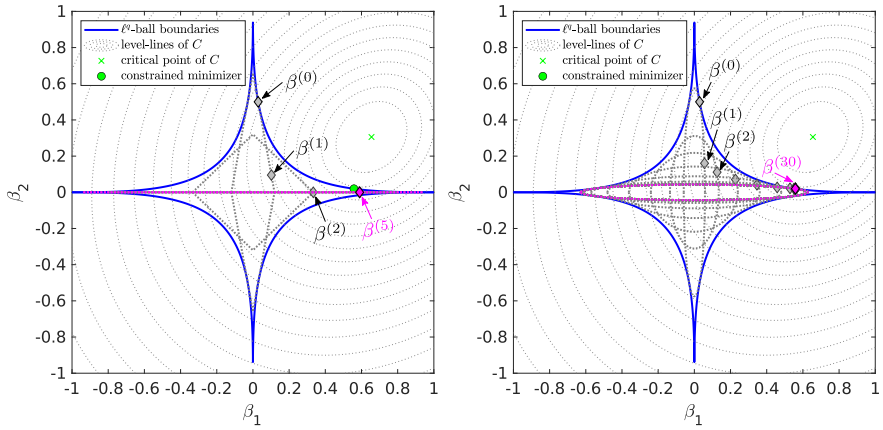


FIG 13. ***Convergence of Algorithm 6 towards a local minimum of*** (63). *A synthetic dataset* $(X, \beta^\star, y)$ *with dimensions* $(n = 5, p = 2)$ *was processed using Algorithm 6 with* $t = 1$ *and* $\nu = 1$ *(left) or* $\nu = 2$ *(right). The same initializer* $\beta^{(0)}$ *was used in both simulations (see the gray diamond marks indicated with a black arrow). The first five iterations of the algorithm are shown for both settings of* $\nu \in \{1, 2\}$. *We can see that, in both cases, the iterates* $\beta^{(k)}$ *seem to be attracted by a point different from the actual minimizer of* C *over* $\mathscr{B}_q^\nu(t^{1/q})$ *(indicated with a green disk). In fact, with the help of the displayed level lines of* C, *we can see that this point corresponds to a local minimizer of* C *over* $\mathscr{B}_q^\nu(t^{1/q})$. *In practice, Algorithm 6 can be, as the AR Algorihtm, very sensitive to the choice of the initializer* $\beta^{(0)}$.

In the experiments presented in Fig. 11, Fig. 12 and Fig. 13, the limit of the scheme iterations seemed to be the same whatever the setting of $\nu \in \{1, 2\}$. In Fig. 14, we show that the limit of the scheme may actually depend from ν. Indeed, we can see in Fig. 14 (a) that the iterate $\beta^{(2)}$ generated by Algorithm 6 with $\nu = 1$ is lying in the first orthan (i.e., $\beta_2^{(2)} = 0$), and so as the further iterates (because of Remark 5). However, we can see in Fig. 14 (b) that, when Algorithm 6 is used with $\nu = 2$ to process the same dataset, the coordinates of the generated iterates do not vanish. Actually, when $\nu = 2$, the computation of an iterate $\beta^{(k+1)}$ assuming vanishing coordinates is unlikely to occur (at least in our experimental setting) because of the ellipsoid shape of the constraint sets $\mathscr{E}_q^\nu(\eta^{(k+1)}, t)$ involved in (69b). However, when Algorithm 6 is used with $\nu = 1$, the diamond shape of the constraint sets $\mathscr{E}_q^\nu(\eta^{(k+1)}, t)$ promotes sparse iterates at each iteration, and vanishing coordinates are likely to be computed. Notice that, as we will shall illustrate at the end of this section, Algorithm 6 with $\nu = 2$ is in practice, exactly as the AR Algorithm, able to generate outputs with vanishing coordinates. In fact, during the scheme iterations, the amplitude of the iterate coordinates may become smaller than the smallest floating point number (so as the corresponding semi-axes of the ellipsoid constraint set), yielding numerically vanishing coordinate. However, when the algorithm is used with $\nu = 1$, vanishing coordinates $\beta_j^{(k+1)} = 0$ can be computed even when $\beta_j^{(k)}$ is far from zero, making the coordinates vanishing process much faster than for $\nu = 2$.



(a) Algorithm 6 with $\nu = 1$ (b) Algorithm 6 with $\nu = 2$

FIG 14. ***Noticeable difference between $\nu = 1$ and $\nu = 2$.*** *In this experiment, we processed a synthetic dataset using Algorithm 6 with $t = 1$, $q = 0.7$ and with $\nu = 1$ (a) or $\nu = 2$ (b). We can see in (a) that iterate $\beta^{(2)}$ is lying in the first orthan ($\beta_2^{(2)} = 0$), so as the next iterates (see Remark 5). We can see in (b) that with the setting $\nu = 2$, the coordinates of the iterates produced by Algorithm 6 never vanish. As for the AR Algorithm, the regularization of the ℓ^q norm by the mean of a parameter δ (see (71)) can be helpful to avoid the persistence of vanishing coordinates (see Fig. 15). Nevertheless, one can notice that the setting $\nu = 1$ yields here convergence towards the solution of the targeted $\mathcal{L}_0$ constrained problem (62), while the setting $\nu = 2$ yields convergence towards the solution its ℓ^q approximation, that is, the ℓ^q constrained problem (63).*

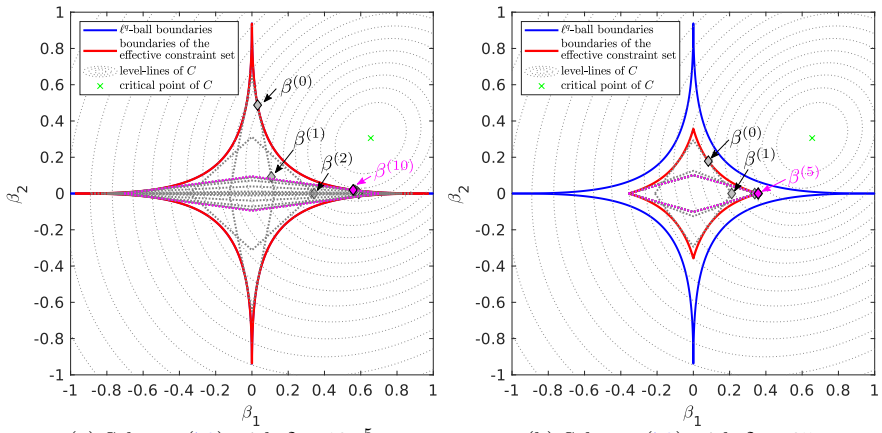
Interestingly enough, in the experiment presented in Fig. 14, the setting $\nu = 1$ yields convergence towards the solution of the $\mathcal{L}_0$ constrained problem (62), while the setting $\nu = 2$ yields convergence towards the solution of the ℓ^q constrained problem (63). On the one hand, interpreting problem (63) as an approximation of the targeted problem (62), the setting $\nu = 1$ is more efficient than the setting $\nu = 2$ in this experiment. On the other hand, the setting $\nu = 1$ also fails to compute (63) although Algorithm 6 was designed for achieving this task. Besides, the persistence of vanishing coordinates along the scheme iterations may be an even more important issue when $\nu = 1$ since this setting precisely promote vanishing coordinate at each iteration. As we explained at the end of Section 5.2, this issue can be easily addressed by considering the δ-regularized problem (71) (through Scheme (72)) instead of (63). According to Remark 8, Algorithm 6 can be easily modified to implement Scheme (72) for any $\delta \geq 0$, provided that we replace its line 5 by

$$\eta^{k+1} \leftarrow \left(|\beta^{(k)}|^\nu + \delta^\nu \right)^{\frac{\nu - q}{\nu}}$$

and its line 7 by

$$z_{k+1} \leftarrow \left[\frac{\nu}{q}\left(t - \frac{q}{\nu}\sum_{j=1}^{p} r(\delta^\nu, \eta_j^{(k+1)}) - \frac{\nu - q}{\nu}\sum_{j=1}^{p}\left(\eta_j^{(k+1)}\right)^{\frac{q}{\nu-q}}\right)\right]^{\frac{1}{\nu}}.$$

In Fig. 15 we display and comment the iterates generated by this modified Algorithm when applied to the same dataset as in Fig. 14.



(a) Scheme (72) with $\delta = 10^{-5}$ (b) Scheme (72) with $\delta = .05$

FIG 15. ***δ-regularized ℓ^q constrained minimization model*** (71). *We processed the same dataset as in Fig. 14 using Scheme* (72) *with* $\delta \in \{10^{-5}, 0.05\}$, $t = 1$, $q = 0.7$ *and* $\nu = 1$. *We display the boundaries of the constraint set* $\mathscr{B}_q^{\nu,\delta}(t^{\nu/q}) := \{\beta \in \mathbb{R}^p\,,\ \||\beta|^\nu + \delta^\nu\|_{q/\nu}^{q/\nu} \le t\}$ *involed in* (71) *using a red thick line while the dotted thick lines here represent the boundaries of the δ-regularized constraint sets* $\mathscr{E}_q^{\nu,\delta}(\eta^{(k+1)}, t)$ *involved in* (72b). *We can see in (a) that, for* $\delta = 10^{-5}$, *the constraint set* $\mathscr{B}_q^{\nu,\delta}(t^{\nu/q})$ *almost coincides with* $\mathscr{B}_q(t^{1/q})$. *We can see also that the first iterates of* (72) *are roughly the same as those in Fig. 14 (a). However, the use of* $\delta = 10^{-5}$ *allows here the iterates to escape from the first orthan after iteration* $k = 2$ *and converge toward to solution of* (71). *In (b), we increase the value of δ and we can see that the δ-regularized constraint set* $\mathscr{B}_q^{\nu,\delta}(t^{\nu/q})$ *(red thick line) is now substantially different from the* ℓ^q*-ball* $\mathscr{B}_q^{\nu}(t^{1/q})$ *(blue thick line). Finally, Scheme* (72) *sucessfully converges towards the solution of* (71), *although the latter is significantly different from that of* (63). *Last, we can remark in (b) that the use of* $\delta > 0$ *with* $\nu = 1$ *is not incompatible with convergence towards a sparse output. This contrasts with the AR Algorithm (and also with Scheme* (72) *with* $\nu = 2$*) since we showed that, with* $\delta > 0$, *the latter generally converges towards points with no vanishing coordinates.*

Finally, we display in Fig. 15 some examples of outputs returned by Algorithm 6 with $\nu \in \{1, 2\}$ applied to a synthetic dataset with larger dimensions $(n = 300, p = 150)$.

6. Conclusion

In this work, we performed a careful and thorough study of the Adaptive-Ridge Algorithm. It is worth noticing that all our mathematical descriptions of this

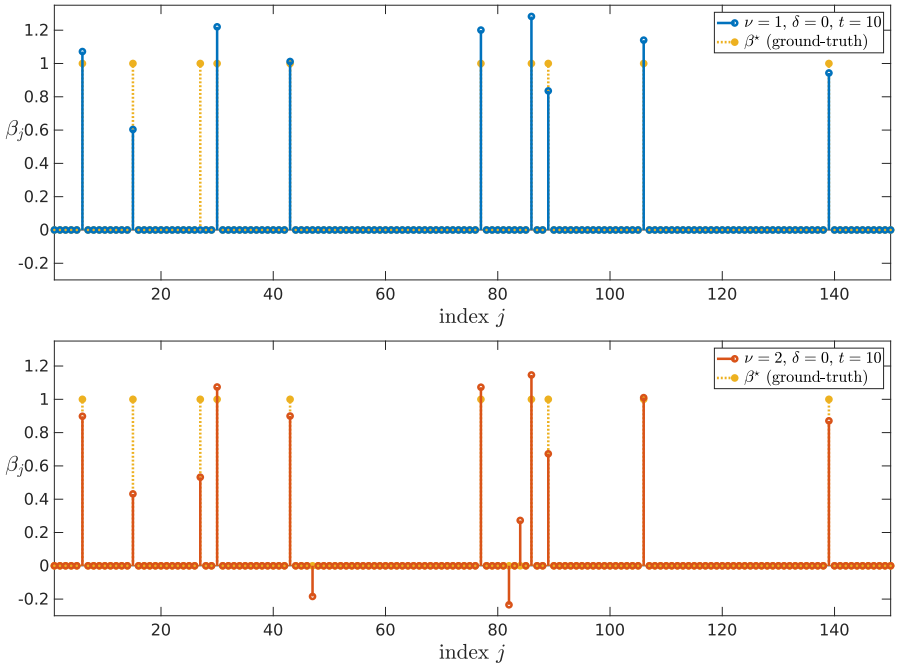


FIG 16. ***ℓ^q constrained minimization in high dimension ($n = 300, p = 150$).*** *We used Algorithm 6 to process a synthetical dataset $(X, \beta^\star, y)$ with dimensions $n = 300$ and $p = 150$. The ground-truth vector $\beta^\star$ is displayed above using yellow dotted stems. As can be seen above, $\beta^\star$ has ten nonzero entries ($\mathcal{L}_0(\beta^\star) = 10$) which encourages us to set $t = 10$ in problem (62). Thus, Algorithm 6 was run using $t = 10$, $q = 0.7$, $\delta = 0$ and $\nu = 1$ (top) or $\nu = 2$ (bottom). In this experiment, Algorithm 6 was initialized using a random initial guess $\beta^{(0)}$ lying in the constraint set $\mathscr{B}_q(t^{1/q})$. We can see that both settings of ν yielded an output with $\mathcal{L}_0$ norm close to $t = 10$ (that is, nine for $\nu = 1$ and thirteen for $\nu = 2$). The setting $\nu = 1$ yields here slightly sparser output (blue plain stems) than that obtained using $\nu = 2$ (orange plain stems), which is a general observation that we also made on many other simulations. As mentioned above, this behavior is probably due to the sparsity-promoting diamond shapes (when $\nu = 1$) of the constraint sets in (69b). Besides, we could observe that convergence of the iterates was roughly fifteen times faster with $\nu = 1$ than with $\nu = 2$ on this datasest. More generally significantly faster convergence was achieved with $\nu = 1$ than with $\nu = 2$ in all our the experiments that we made.*

algorithm and its properties were simply built from the variational formulation of the ℓ^q penalty term presented in Proposition 1. We found the latter very inspirational to understand, interpret and extend the AR scheme. In this review paper, we pointed out the existing links between this algorithm and many others coming from the literature. In particular, we showed that, when used with its recommended setting, the AR Algorithm corresponds to a particular instance of the so-called IRLS class of algorithms that is still the subject of active and fruitful researchs in the field of Compressed Sensing, Sparse Signal Recovery and Nonconvex Optimization. Then, we discussed about the practical implementation of the AR Algorithm, with a particular focus on the handling of numerical

errors related to matrix conditioning issues. We performed an in-depth experimental study of the AR Algorithm that may hopefully benefit to its potential users, by providing some useful insights about its behavior and parameter tuning in practical situations. Last, using again the variational formulation of the ℓ^q penalty, we extended the AR Algorithm to address the problem of minimization of a functional C over nonconvex ℓ^q balls. Depending on the setting of the ν parameter in Proposition 1, the derived scheme boils down to iterating operations of minimizations of C over elipsoid ($\nu = 2$) or diamond ($\nu = 1$) convex constraint sets. As for the AR scheme, we found out that the underlying variational formulation of the ℓ^q penalty yielded a very natural and simple derivation of this scheme and its mathematical properties. An implementation of this scheme, based on modern proximal algorithms, was proposed in the case where C is a quadratic function. Thanks to the considerable advances made in the field of convex and nonsmooth optimization within the two last decades, this algorithm can be easily extended to handle more complex functionals C (e.g., ℓ^1 or Poisson log-likelihood functionals). More importantly, we believe that nonconvex ℓ^q-ball constrained minimization opens very interesting alternative to the more commonly considered ℓ^q penalized minimization framework, especially because the hyperparameter setting seems greatly simplified in the constrained case. As pointed out in [50, Section 1.1], the literature dedicated to this topic is very limited and we believe that the underlying ℓ^q-ball constrained minimization model opens up perspectives of interesting advances both in terms of mathematical analysis and numerical algorithm development.

Appendix A: Proof of Proposition 3

Assume that $C : \beta \mapsto \sum_{j=1}^{p} c_j(\beta_j)$ where $c_j : \mathbb{R} \to \mathbb{R}$ for all $j \in \{1, 2, \dots, p\}$. Then, $E_{\lambda,q}$ is additively separable,

$$\forall \beta \in \mathbb{R}^p\,, \quad E_{\lambda,q}(\beta) = \sum_{j=1}^{p} \left(e_j(\beta_j) := c_j(\beta_j) + \lambda\,|\beta_j|^q\right)\,.$$

Thanks to Lemma (1), when $\nu > q$, for all $j \in \{1, 2, \dots, p\}$ and for all $\eta_j \in \mathbb{R}_+^*$, we have the upper-bound

$$\forall \beta_j \in \mathbb{R}\,, \quad e_j(\beta_j) \le s_j(\beta_j, \eta_j) := c_j(\beta_j) + \lambda\,\ell_q^\nu(\beta_j, \eta_j)\,.$$

Besides, when $\beta_j \neq 0$, we have the equality $e_j(\beta_j) = s_j(\beta_j, |\beta_j|^{\nu-q})$. Now, given $k \ge 0$, since $\eta_j^{(k)} = |\beta_j^{(k)}|^{\nu-q}$ and thanks to the additive separability of C, (19a) yields, for all $j \in \{1, 2, \dots, p\}$,

$$\beta_j^{(k+1)} \in \operatorname*{argmin}_{\beta_j \in \mathbb{R}^p} f_j(\beta_j) := c_j(\beta_j) + \frac{\lambda q}{\nu}\, r(\beta_j, |\beta_j^{(k)}|^{\nu-q}). \tag{84}$$

When $\beta_j^{(k)} = 0$, $f_j(\beta_j)$ is finite for $\beta_j = 0$ and takes the value $+\infty$ otherwise, leading to $\beta_j^{(k+1)} = 0$. When $\beta_j^{(k)} \neq 0$, the minimizers of f_j are the same as

those of $\beta_j \mapsto s_j(\beta_j, |\beta_j^{(k)}|^{\nu-q})$ since, in that case, $f_j(\beta_j) = s_j(\beta_j, |\beta_j^{(k)}|^{\nu-q})$. Finally, (84) yields

$$\begin{cases} \beta_j^{(k+1)} = 0 & \text{if } \beta_j^{(k)} = 0 \\ \beta_j^{(k+1)} \in \underset{\beta_j \in \mathbb{R}^p}{\operatorname{argmin}} \; s_j(\beta_j, |\beta_j^{(k)}|^{\nu-q}) & \text{otherwise.} \end{cases}$$

Therefore, when $\beta_j^{(k)} = 0$, we have $e_j(\beta_j^{(k+1)}) = e_j(\beta_j^{(k)})$. Otherwise, when $\beta_j^{(k)} \neq 0$, we have

$$e_j(\beta_j^{(k+1)}) \leq s_j(\beta_j^{(k+1)}, |\beta_j^{(k)}|^{\nu-q}) \leq s_j(\beta_j^{(k)}, |\beta_j^{(k)}|^{\nu-q}) = e_j(\beta_j^{(k)}) \,.$$

Finally, we get $e_j(\beta_j^{(k+1)}) \leq e_j(\beta_j^{(k)})$ whatever the value of $\beta_j^{(k)} \in \mathbb{R}$ and, summing for $j \in \{1, 2, \ldots, p\}$, we obtain $E_{\lambda,q}(\beta^{(k+1)}) \leq E_{\lambda,q}(\beta^{(k)})$ as announced.

Appendix B: Proof of Lemma 2

Thanks to Proposition 1, we have, for all $r \in (0,1)$ and for all $z \in (\mathbb{R}^*)^p$,

$$\|z\|_r^r = \inf_{\eta \in (\mathbb{R}_+^*)^p} \mathcal{L}_r^1(z, \eta) = \inf_{\eta \in (\mathbb{R}_+^*)^p} \; r \sum_{j=1}^p \frac{|z_j|}{\eta_j} + (1-r) \sum_{j=1}^p \eta_j^{\frac{r}{1-r}} \tag{85}$$

and this infimum is in fact a minimum (since $z \in (\mathbb{R}^*)^p$) which is attained at $\eta = |z|^{1-r}$. Let $\beta \in \mathbb{R}^p$, $\nu > q$ and $\delta > 0$, taking $r = \frac{q}{\nu}$ and $z = |\beta|^\nu + \delta^\nu$ in (85) yields

$$\left\| |\beta|^\nu + \delta^\nu \right\|_{q/\nu}^{q/\nu} = \min_{\eta \in (\mathbb{R}_+^*)^p} \; \frac{q}{\nu} \sum_{j=1}^p \frac{|\beta_j|^\nu + \delta^\nu}{\eta_j} + \frac{\nu - q}{\nu} \sum_{j=1}^p \eta_j^{\frac{q}{\nu-q}} \,, \tag{86}$$

with the minimum attained at $\eta = (|\beta|^\nu + \delta^\nu)^{\frac{\nu-q}{\nu}}$. Remarking that we have $E_{\lambda,q}^{\nu,\delta} : \beta \mapsto C(\beta) + \lambda \left\| |\beta|^\nu + \delta^\nu \right\|_{q/\nu}^{q/\nu}$ and using (86) yields (23) as announced.

Appendix C: Proof of Proposition 5

First, let us consider the unidimensional case. For all $\nu > q$, let us show that

$$\forall \beta \in \mathbb{R} \,, \quad |\beta|^q = \min_{\eta \geq 0} \; \frac{q}{\nu} \cdot r(|\beta|, \eta) + \frac{\nu - q}{\nu} \cdot \eta^{\frac{q}{\nu-q}} \,, \tag{87}$$

and that that minimum is attained at $\eta = |\beta|^{\nu-q}$.

For $\beta = 0$, we have $r(|\beta|, \eta) = 0$ for any $\eta \geq 0$. Therefore, the minimum in (87) is attained at $\eta = 0 = |\beta|^{\nu-q}$ and both sides of (87) vanish.

For $\beta \neq 0$, $r(|\beta|, \eta)$ takes the value $+\infty$ for $\eta = 0$ and takes the (finite) value $\frac{|\beta|}{\eta}$ for $\eta > 0$. Therefore the minimum over $\mathbb{R}_+$ in (87) can be restricted to $\mathbb{R}_+^*$, and the right-hand side of (87) becomes

$$\min_{\eta>0} \ \frac{q}{\nu} \cdot \frac{|\beta|}{\eta} + \frac{\nu - q}{\nu} \cdot \eta^{\frac{q}{\nu-q}} .$$

which is equal to $|\beta|^q$ (with the minimum attained at $\eta = |\beta|^{\nu-q}$) as stated in Lemma 1.

In higher dimension, for all $\beta \in \mathbb{R}^p$ and for all $\nu > q$, using (87) and the additive separability of the ℓ^q penalty, we get

$$\|\beta\|_q^q = \sum_{j=1}^p |\beta_j|^q = \min_{\eta \in \mathbb{R}_+^p} \ \frac{q}{\nu} \sum_{j=1}^p r(|\beta_j|, \eta_j) + \frac{\nu - q}{\nu} \sum_{j=1}^p \eta_j^{\frac{q}{\nu-q}} , \tag{88}$$

with the minimum attained at $\eta = |\beta|^{\nu-q}$, from which (28) follows.

Appendix D: Details about Scheme (72) and proof of Proposition 8

Let $\delta > 0$, $q \in (0, 2)$ and $\nu > q$. Using Proposition 1, we have

$$\forall \beta \in \mathbb{R}^p , \quad \| |\beta|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} = \min_{\eta \in (\mathbb{R}_+^*)^p} \ \mathcal{L}^1_{q/\nu}(|\beta|^\nu + \delta^\nu, \eta) \tag{89}$$

and the minimum in (89) is attained at $\eta = (|\beta|^\nu + \delta^\nu)^{\frac{\nu-q}{\nu}}$. Thus, an alternating minimization approach to address (71) can be implemented by setting $\beta^{(0)}$ such as $\| |\beta^{(0)}|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} \leq t$ and iterating, for $k \geq 0$,

$$\begin{cases} \eta^{(k+1)} = \left(|\beta^{(k)}|^\nu + \delta^\nu \right)^{\frac{\nu-q}{\nu}} & \text{(90a)} \\ \beta^{(k+1)} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ C(\beta) \quad \text{subject to} \quad \mathcal{L}^1_{q/\nu}(|\beta|^\nu + \delta^\nu, \eta^{(k+1)}) \leq t . & \text{(90b)} \end{cases}$$

Since (73) satisfies

$$\forall \eta \in (\mathbb{R}_+^*)^p , \quad \mathscr{E}_q^{\nu,\delta}(\eta, t) = \left\{ \beta \in \mathbb{R}^p , \ \mathcal{L}^1_{q/\nu}(|\beta|^\nu + \delta^\nu, \eta) \leq t \right\} ,$$

the scheme iteration (72) is none other than (90).

Now, let us prove Proposition 8 by following exactly the same steps as in the proof of Proposition 7. Let $k \geq 0$, and $\beta \in \mathscr{E}_q^{\nu,\delta}(\eta^{(k+1)}, t)$. From (89), we have

$$\| |\beta|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} = \min_{\eta \in (\mathbb{R}_+^*)^p} \mathcal{L}^1_{q/\nu}(|\beta|^\nu, \eta) \leq \mathcal{L}^1_{q/\nu}(|\beta|^\nu, \eta^{(k+1)}) \leq t .$$

Therefore, we have,

$$\forall k \geq 0 , \quad \mathscr{E}_q^{\nu,\delta}(\eta^{(k+1)}, t) \subset \mathscr{B}_q^{\nu,\delta}(t^{\frac{1}{q}}) := \left\{ \beta \in \mathbb{R}^p , \ \| |\beta|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} \leq t \right\} .$$

Since we imposed $\beta^{(0)} \in \mathscr{B}_q^{\nu,\delta}(t^{\frac{1}{q}})$, and since (90b) ensures that, for all $k \geq 0$, $\beta^{(k+1)} \in \mathscr{E}_q^{\nu,\delta}(\eta^{(k+1)}, t) \subset \mathscr{B}_q^{\nu,\delta}(t^{\frac{1}{q}})$, it follows that the sequence $(\beta^{(k)})_{k\geq 0}$ has all its elements in $\mathscr{B}_q^{\nu,\delta}(t^{\frac{1}{q}})$, as announced in Proposition 8. Let $k \geq 0$, from $\beta^{(k)} \in \mathscr{B}_q^{\nu,\delta}(t^{\frac{1}{q}})$ and, using again (89), we get

$$t \geq \| |\beta^{(k)}|^\nu + \delta^\nu \|_{q/\nu}^{q/\nu} = \min_{\eta \in (\mathbb{R}_+^*)^p} \mathcal{L}^1_{q/\nu}(\beta^{(k)}, \eta) = \mathcal{L}^1_{q/\nu}(\beta^{(k)}, \eta^{(k+1)})$$

since $\eta^{(k+1)} = \left(|\beta^{(k)}|^\nu + \delta^\nu\right)^{\frac{\nu-q}{\nu}}$. It follows that $\beta^{(k)} \in \mathscr{E}_q^{\nu,\delta}(\eta^{(k+1)}, t)$. Consequently, (90b) necessarily generates $\beta^{(k+1)}$ such as $C(\beta^{(k+1)}) \leq C(\beta^{(k)})$ which ends the proof of Proposition 8.

Appendix E: Proof of Lemma 4

Let $\widetilde{\beta} \in \mathscr{B}_q(t)$, let $\widetilde{\eta} = |\widetilde{\beta}|^{\nu-q}$ and let $z_q^\nu(\widetilde{\eta}, t)$ the quantity defined in (76). Let us denote by $J = \{j \in \{1, 2, \ldots, p\},\ \widetilde{\beta}_j \neq 0\}$ the support of $\widetilde{\beta}$ and by $J^c = \{1, 2, \ldots, p\} \setminus J$ its complementary in $\{1, 2, \ldots, p\}$. The statement

$$\beta \in \mathscr{D}(\widetilde{\eta}^{\frac{1}{\nu}}) \mathscr{B}_\nu(z_q^\nu(\widetilde{\eta}, t)) \tag{91}$$

is equivalent to

$$\exists \beta' \in \mathscr{B}_\nu(z_q^\nu(\widetilde{\eta}, t)) \quad \text{such that} \quad \forall j \in \{1, 2, \ldots, p\},\ \ \beta_j = \widetilde{\eta}_j^{\frac{1}{\nu}} \beta'_j. \tag{92}$$

Since J is also the support of $\widetilde{\eta}$ (i.e., $\widetilde{\eta}_j \neq 0$ for all $j \in J$ and $\widetilde{\eta}_j = 0$ for all $j \in J^c$), (92) is equivalent to

$$\exists \beta' \in \mathscr{B}_\nu(z_q^\nu(\widetilde{\eta}, t)) \quad \text{such that} \quad \begin{cases} \forall j \in J^c, & \beta_j = 0 \\ \forall j \in J, & \beta_j / \widetilde{\eta}_j^{\frac{1}{\nu}} = \beta'_j \end{cases} \tag{93}$$

and to

$$\begin{cases} \forall j \in J^c, & \beta_j = 0 \\ \displaystyle\sum_{j \in J} \frac{|\beta_j|^\nu}{\widetilde{\eta}_j} \leq \left(z_q^\nu(\widetilde{\eta}, t)\right)^\nu. \end{cases} \tag{94}$$

For all $j \in J^c$ we have $\widetilde{\eta}_j = 0$ and, by definition of the generalized ratio function r (see (18)), we have $r(0, 0) = 0$. Therefore (94) implies that

$$\sum_{j=1}^p r(|\beta_j|^\nu, \widetilde{\eta}_j) \leq \left(z_q^\nu(\widetilde{\eta}, t)\right)^\nu. \tag{95}$$

Reciprocally, (95) imposes that $\beta_j = 0$ for all $j \in J^c$ (otherwise, the left-hand side term in (95) is $+\infty$), and thus ensures that (94) is fulfilled. Therefore, (94) and (95) are equivalent. Now, using (76), we can see that (95) is equivalent to

$$\sum_{j=1}^p r(|\beta_j|^\nu, \widetilde{\eta}_j) \leq \frac{\nu}{q} \left(t - \frac{\nu - q}{\nu} \cdot \sum_{j=1}^p \widetilde{\eta}_j^{\frac{q}{\nu-q}} \right), \tag{96}$$

and to

$$\frac{q}{\nu}\cdot\sum_{j=1}^{p} r(|\beta_j|^\nu,\widetilde{\eta}_j)+\frac{\nu-q}{q}\cdot\sum_{j=1}^{p}\widetilde{\eta}_j^{\frac{q}{\nu-q}}\leq t\,. \tag{97}$$

Last, we can see that (97) exactly means that $\mathscr{L}_q^\nu(\beta,\widetilde{\eta})\leq t$ (see (65)), i.e., that $\beta\in\mathscr{E}_q^\nu(\widetilde{\eta},t)$ (see (68)). Finally, we have shown the equivalence

$$\beta\in\mathscr{D}(\widetilde{\eta}^{\frac{1}{\nu}})\mathscr{B}_\nu(z_q^\nu(\widetilde{\eta},t))\;\Leftrightarrow\;\beta\in\mathscr{E}_q^\nu(\widetilde{\eta},t)$$

which was announced in Lemma E.

References

[1] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. and Sorensen, D. (1999). *LAPACK Users' Guide*, Third ed. Society for Industrial and Applied Mathematics, Philadelphia, PA. ISBN 978-0-898714-47-0.

[2] Aubert, G. and Vese, L. (1997). A Variational Method in Image Recovery. *SIAM Journal on Numerical Analysis* **34** 1948-1979. https://doi.org/10.1137/S003614299529230X.

[3] Black, M. J. and Rangarajan, A. (1996). On the Unification of Line Processes, Outlier Rejection, and Robust Statistics with Applications in Early Vision. *International Journal of Computer Vision* **19** 57–91. https://doi.org/10.1007/BF00131148.

[4] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press ISBN 9780521833783.

[5] Candes, E. J., Wakin, M. B. and Boyd, S. P. (2008). Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier analysis and applications* **14** 877–905. https://doi.org/10.1007/s00041-008-9045-x.

[6] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* **40** 120–145. https://doi.org/10.1007/s10851-010-0251-1.

[7] Chambolle, A. and Pock, T. (2016). On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming* **159** 253–287. https://doi.org/10.1007/s10107-015-0957-3.

[8] Chan, R. H. and Liang, H.-X. (2014). Half-Quadratic Algorithm for ℓ_p-ℓ_q Problems with Applications to TV-ℓ_1 Image Restoration and Compressive Sensing. In *Efficient Algorithms for Global Optimization Methods in Computer Vision* (A. Bruhn, T. Pock and X.-C. Tai, eds.) 78–103. Springer Berlin Heidelberg, Berlin, Heidelberg. https://dx.doi.org/10.1007/978-3-642-54774-4_4.

[9] Charbonnier, P., Blanc-Feraud, L., Aubert, G. and Barlaud, M. (1994). Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing* **2** 168-172. https://doi.org/10.1109/ICIP.1994.413553.

[10] Chartrand, R. and Y., W. (2008). Iteratively reweighted algorithms for compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing* 3869-3872. https://doi.org/10.1109/ICASSP.2008.4518498.

[11] Condat, L. (2013). A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of optimization theory and applications* **158** 460–479. https://doi.org/10.1007/s10957-012-0245-9.

[12] Daubechies, I., DeVore, R., Fornasier, M. and Güntürk, C. S. (2010). Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics* **63** 1-38. https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.20303.

[13] Donoho, D. L., Elad, M. and Temlyakov, V. N. (2006). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory* **52** 6-18. https://doi.org/10.1109/TIT.2005.860430.

[14] Drori, Y., Sabach, S. and Teboulle, M. (2015). A simple algorithm for a class of nonsmooth convex–concave saddle-point problems. *Operations Research Letters* **43** 209-214. https://doi.org/10.1016/j.orl.2015.02.001.

[15] Duchi, J., Shalev-Shwartz, S., Singer, Y. and Chandra, T. (2008). Efficient Projections onto the L1-Ball for Learning in High Dimensions. In *Proceedings of the 25th International Conference on Machine Learning. ICML '08* 272–279. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1390156.1390191.

[16] Ekeland, I. and Témam, R. (1999). *Convex Analysis and Variational Problems*. Society for Industrial and Applied Mathematics https://epubs.siam.org/doi/abs/10.1137/1.9781611971088.

[17] Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association* **96** 1348-1360. https://doi.org/10.1198/016214501753382273.

[18] Fornasier, M., Peter, S., Rauhut, H. and Worm, S. (2016). Conjugate gradient acceleration of iteratively re-weighted least squares methods. *Computational optimization and applications* **65** 205–259. https://doi.org/10.1007/s10589-016-9839-8.

[19] Foucart, S. and Lai, M.-J. (2009). Sparsest solutions of underdetermined linear systems via ℓ^q-minimization for $0 < q \leq 1$. *Applied and Computational Harmonic Analysis* **26** 395-407. https://doi.org/10.1016/j.acha.2008.09.001.

[20] Frommlet, F. and Nuel, G. (2016). An Adaptive Ridge Procedure for L0 Regularization. *PLOS ONE* **11** 1-23. https://doi.org/10.1371/journal.pone.0148620.

[21] Geman, D. and Reynolds, G. (1992). Constrained Restoration and the Recovery of Discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14** 367–383. https://doi.org/10.1109/34.120331.

[22] Geman, D. and Yang, C. (1995). Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing* **4** 932-946. https://doi.org/10.1109/83.392335.

[23] Goepp, V. (2019). An Iterative Regularized Method for Segmentation with Applications to Statistics, Theses, Université de Paris / Université Paris Descartes (Paris 5) https://hal.archives-ouvertes.fr/tel-02473848.

[24] Goepp, V., Thalabard, J.-C., Nuel, G. and Bouaziz, O. (2021). Regularized Bidimensional Estimation of the Hazard Rate. *The International Journal of Biostatistics*. https://doi.org/10.1515/ijb-2019-0003.

[25] Gribonval, R. and Nielsen, M. (2007). Highly sparse representations from dictionaries are unique and independent of the sparseness measure. *Applied and Computational Harmonic Analysis* **22** 335-355. https://doi.org/10.1016/j.acha.2006.09.003.

[26] Hastie, T., Tibshirani, R. and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* **2**. Springer https://doi.org/10.1007/978-0-387-21606-5.

[27] Hoerl, A. E. and Kennard, R. W. (2000). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **42** 80–86. https://doi.org/10.1080/00401706.2000.10485983.

[28] Hunter, D. R. and Li, R. (2005). Variable selection using MM algorithms. *Annals of statistics* **33** 1617-1642. https://doi.org/10.1214/009053605000000200.

[29] Idier, J. (2001). Convex half-quadratic criteria and interacting auxiliary variables for image restoration. *IEEE Transactions on Image Processing* **10** 1001-1009. https://doi.org/10.1109/83.931094.

[30] Jenatton, R., Obozinski, G. and Bach, F. (2010). Structured Sparse Principal Component Analysis. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterington, eds.). *Proceedings of Machine Learning Research* **9** 366–373. PMLR, Chia Laguna Resort, Sardinia, Italy. https://proceedings.mlr.press/v9/jenatton10a.html.

[31] Lai, M. J., Xu, Y. and Yin, W. (2013). Improved Iteratively Reweighted Least Squares for Unconstrained Smoothed ℓ_q Minimization. *SIAM Journal on Numerical Analysis* **51** 927-957. https://doi.org/10.1137/110840364.

[32] Liang, J., Fadili, J. and Peyré, G. (2017). Activity Identification and Local Linear Convergence of Forward–Backward-type Methods. *SIAM Journal on Optimization* **27** 408-437. https://doi.org/10.1137/16M106340X.

[33] Mairal, J., Bach, F. and Ponce, J. (2014). Sparse Modeling for Image and Vision Processing. *Foundations and Trends® in Computer Graphics and Vision* **8** 85-283. https://dx.doi.org/10.1561/0600000058.

[34] Needell, D. (2009). Noisy signal recovery via iterative reweighted L1-minimization. In *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers* 113-117. https://doi.org/

10.1109/ACSSC.2009.5470154.

[35] Nikolova, M. (2016). Relationship between the optimal solutions of least squares regularized with ℓ_0-norm and constrained by k-sparsity. *Applied and Computational Harmonic Analysis* **41** 237-265. https://doi.org/10.1016/j.acha.2015.10.010.

[36] Nikolova, M. and Chan, R. H. (2007). The Equivalence of Half-Quadratic Minimization and the Gradient Linearization Iteration. *IEEE Transactions on Image Processing* **16** 1623-1627. https://dx.doi.org/10.1109/TIP.2007.896622.

[37] Nikolova, M. and Ng, M. K. (2005). Analysis of Half-Quadratic Minimization Methods for Signal and Image Recovery. *SIAM Journal on Scientific Computing* **27** 937-966. https://doi.org/10.1137/030600862.

[38] Notay, Y. (1993). On the convergence rate of the conjugate gradients in presence of rounding errors. *Numerische Mathematik* **65** 301–317. https://doi.org/10.1007/BF01385754.

[39] Ochs, P., Dosovitskiy, A., Brox, T. and Pock, T. (2015). On Iteratively Reweighted Algorithms for Nonsmooth Nonconvex Optimization in Computer Vision. *SIAM Journal on Imaging Sciences* **8** 331-372. https://doi.org/10.1137/140971518.

[40] Peleg, D. and Meir, R. (2008). A bilinear formulation for vector sparsity optimization. *Signal Processing* **88** 375-389. https://doi.org/10.1016/j.sigpro.2007.08.015.

[41] Peter, S. (2016). Algorithms for Robust and Fast Sparse Recovery, Dissertation, Technische Universität München, München.

[42] Rippe, R. C. A., Meulman, J. J. and Eilers, P. H. C. (2012). Visualization of Genomic Changes by Segmented Smoothing Using an L0 Penalty. *PLOS ONE* **7** 1-14. https://doi.org/10.1371/journal.pone.0038230.

[43] Rockafellar, R. T. (1970). Convex Analysis (Princeton Mathematical Series). *Princeton University Press* **46** 49. https://doi.org/10.1515/9781400873173.

[44] Soubies, E., Blanc-Féraud, L. and Aubert, G. (2015). A Continuous Exact ℓ_0 Penalty (CEL0) for Least Squares Regularized Problem. *SIAM Journal on Imaging Sciences* **8** 1607-1639. https://doi.org/10.1137/151003714.

[45] Strakoš, Z. (1991). On the real convergence rate of the conjugate gradient method. *Linear Algebra and its Applications* **154-156** 535-549. https://doi.org/10.1016/0024-3795(91)90393-B.

[46] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **58** 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[47] Vũ, B. C. (2013). A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics* **38** 667–681. https://doi.org/10.1007/s10444-011-9254-8.

[48] Van der Sluis, A. and van der Vorst, H. A. (1986). The rate of convergence of conjugate gradients. *Numerische Mathematik* **48** 543–560. https://doi.org/10.1007/BF01389450.

[49] Voronin, S. and Daubechies, I. (2015). An Iteratively Reweighted Least Squares Algorithm for Sparse Regularization. https://arxiv.org/abs/1511.08970.

[50] Wang, H., Yang, X. and Deng, X. (2021). A Hybrid First-Order Method for Nonconvex ℓ_p-ball Constrained Optimization. https://doi.org/10.48550/ARXIV.2104.04400.

[51] Wang, H., Yang, X. and Jiang, W. (2021). An Iteratively Reweighted Method for Sparse Optimization on Nonconvex ℓ_p Ball. https://doi.org/10.48550/ARXIV.2104.02912.

[52] Yang, X., Wang, J. and Wang, H. (2022). Towards An Efficient Approach for the Nonconvex lp Ball Projection: Algorithm and Analysis. *Journal of Machine Learning Research* **23** 1–31. http://jmlr.org/papers/v23/21-0133.html.

[53] Zhang, N. and Li, Q. (2017). On optimal solutions of the constrained ℓ_0 regularization and its penalty problem. *Inverse Problems* **33** 025010. https://doi.org/10.1088/1361-6420/33/2/025010.

[54] Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association* **101** 1418-1429. https://doi.org/10.1198/016214506000000735.

[55] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)* **67** 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[56] Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics* **36** 1509-1533. https://doi.org/10.1214/009053607000000802.