Regularization path via ℓ_0 Bregman relaxations

Mhamed ESSAFRI[1] Luca CALATRONI[2] Emmanuel SOUBIES[1]

[1]IRIT, Université de Toulouse, CNRS, Toulouse, France

[2]MaLGa Centre, DIBRIS, Università di Genova & MMS, Istituto Italiano di Tecnologia, Genoa, Italy

Résumé – L'optimisation des problèmes impliquant la pseudo-norme ℓ_0 joue un rôle important en traitement du signal et en apprentissage automatique. En raison de la nature intrinsèquement NP-difficile de ces problèmes, des relaxations continues (potentiellement non convexes) ont attiré une attention significative ces dernières années. En particulier, la notion de "ℓ_0 Bregman relaxation" (B-rex)—une classe de relaxations exactes pour des critères régularisés en norme ℓ_0 avec des termes d'attache aux données généraux définie à partir de distances de Bregman—a été proposée. Ces relaxations sont exactes dans le sens où elles préservent les minimiseurs globaux tout en éliminant certains minimiseurs locaux. En s'appuyant sur ces propriétés, nous proposons un nouvel algorithme pour estimer le chemin des solutions pour différents niveaux de parcimonie. Nous présentons les aspects méthodologiques de cette approche, ainsi que des exemples illustratifs et numériques démontrant son efficacité.

Abstract – The optimization of problems involving the ℓ_0-pseudo norm plays an important role in signal processing and machine learning. Due to the intrinsic NP-hardness of such problems, continuous (potentially non-convex) relaxations have gained significant attention in recent years. In particular, the notion of the ℓ_0 Bregman relaxations (B-rex)—a class of exact relaxations for ℓ_0-regularized criteria with general data terms defined in terms of Bregman distances—have been proposed. These relaxations are exact in the sense that they preserve global minimizers while eliminating certain local minimizers. Building on these properties, we propose a new algorithm to estimate the path of solutions across a range of sparsity levels. We discuss the methodological aspects of this approach, along with illustrative and numerical examples that demonstrate its effectiveness.

1 Introduction

Over the recent years, the increasing predominance of high-dimensional data has sparked significant interest in sparse models in many scientific fields, such as machine learning, statistics, and signal/image processing. Sparsity is particularly valuable in such settings as it enables the construction of compact models with reduced complexity. To that end, the natural approach is to solve problems of the form

$$\hat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathcal{C}^N}{\operatorname{argmin}} \left\{ J_0(\mathbf{x}) := F_{\mathbf{y}}(\mathbf{A}\mathbf{x}) + \lambda_0\|\mathbf{x}\|_0 + \frac{\lambda_2}{2}\|\mathbf{x}\|_2^2 \right\} \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{M\times N}$ is a given design matrix, and $\mathbf{y} \in \mathbb{R}^M$ represents the observed data. The function $F_{\mathbf{y}} : \mathbb{R}^M \mapsto \mathbb{R}$ serves as a data-fidelity term, measuring the discrepancy between the linear model $\mathbf{Ax}$ and the observation $\mathbf{y}$. The set $\mathcal{C}$ is a constraint set, which is either $\mathbb{R}$ or $\mathbb{R}_{\geq 0}$ in this paper. The regularization terms include the sparisty-promoting ℓ_0-pseudo norm $\|\cdot\|_0$, which counts the number of nonzero elements in its argument, and the standard ℓ_2-norm $\|\cdot\|_2^2$, which introduces ridge regularization. The hyperparameters $\lambda_0 > 0$ and $\lambda_2 \geq 0$ control the strength of the sparsity-promoting ℓ_0 term and the smoothness-inducing ℓ_2 term, respectively.

Examples of data-fidelity terms $F_{\mathbf{y}}$ include least squares (LS) [4], Kullback-Leibler (KL) divergence [6], and logistic regression (LR) [3], which commonly appear in various fields. Due to the presence of the ℓ_0-pseudo norm, Problem (1) is NP-hard [13]. Additionally, selecting an appropriate value for the parameter λ_0 balancing the amount of regularization with the data term is generally a challenging task. Over the past decades, much research efforts have focused on two directions: (i) developing relaxed formulations of the problem to make it more tractable, and (ii) constructing solutions to (1) for a range of λ_0 values, which is often called ℓ_0 *path*.

Exact relaxations. One strategy to simplify (1) seeks to replace the ℓ_0-pseudo norm with a continuous approximation, ensuring that the resulting relaxation (typically non-convex) preserves the global minimizers of the original problem. Moreover, these relaxed formulations shall also eliminate certain local minimizers, thereby simplifying the optimization landscape and making it more amenable to efficient optimization algorithms. Over the years, various exact relaxation approaches have been proposed. For least squares data terms, the authors in [16] introduced an exact relaxation known as CEL0 (Continuous Exact ℓ_0). This particular penalty can be seen as the "quadratic envelope" of the ℓ_0 pseudo-norm, as introduced in [7]. Moving beyond least squares, the authors in [12] proposed a class of exact relaxations based on Mathematical Programs with Equilibrium Constraints (MPEC). In [2] the authors demonstrated that the capped-ℓ_1 penalty leads to exact relaxations when the data term is Lipschitz continuous. Finally, extending the analysis of [16, 7] to general (non-quadratic) data terms, we introduced in [9] the ℓ_0 Bregman relaxation (B-rex), which provides a class of exact relaxations by leveraging Bregman divergences.

Regularization path. For least-squares, the construction of solutions to (1) for different λ_0 with $\lambda_2 = 0$ has been studied in [15]. Replacing the ℓ_0 term by MCP (Minimax Concave Penalty) penalty –of which CEL0 is a specific instance–, the authors proposed a pathwise algorithm based on coordinate descent together with warm-start and graduated non-convexity (GNC). There, the path is built for a given grid of λ_0 which is given as input. In [5], the authors propose two algorithms for constructing the ℓ_0 path. Both algorithms exploit the piecewise linear nature and concavity of the so-called ℓ_0-curve (see Section 3 for details). As opposed to [15], the explored λ_0 are directly computed by the algorithms and not required as input. Finally, in [11], the authors develop fast algorithms based on coordinate descent and local combinatorial optimization. They use continuation on a grid of λ_0 values with a warm-start strategy. Paper [8] generalizes this work for binary classification problems, including logistic regression.

Contributions and outline. We propose a new algorithm, called `L0PathBrex`, which exploits the nice properties of B-rex in order to estimate an ℓ_0 path. In Section 2, we first recall the concept of the ℓ_0-curve associated to the ℓ_0 path introduced in [5], as well as the B-rex proposed in [9]. Then, Section 3 provides a description of the proposed algorithm, `L0PathBrex`. Finally, in Section 4 we benchmark our proposed method against two algorithms of the `L0Learn` package [11].

2 Preliminaries

In this section, we first recall some properties of the minimizers of J_0 in (1) and define the associated ℓ_0-curve. Then, we review the key components of the ℓ_0 Bregman-relaxation [9].

2.1 Local minimizers of J_0 and ℓ_0-curve

Local minimizers of J_0. In [9, Proposition 2] a characterization of the local minimizers of Problem (1) is provided, showing that finding local minimizers of J_0 reduces to solving convex optimization problems for fixed supports, i.e., subsets of $[N] := \{1, \dots, N\}$ containing the nonzero entries of the minimizer. In particular, these subproblems are *independent of* λ_0, meaning that local minimizers of J_0 are the same whatever the value λ_0. Furthermore, as stated in [9, Theorem 3], the strict minimizers of J_0 are countable, although their number grows exponentially with the dimension.

The ℓ_0-curve. By the invariance of the set of minimizers discussed above, one can see that, in the $(\lambda_0, J_0(\hat{\mathbf{x}}))$-plane, each local minimizer $\hat{\mathbf{x}}$ of J_0 defines an affine line with slope $\|\hat{\mathbf{x}}\|_0$ and constant $\hat{b} = F_{\mathbf{y}}(\mathbf{A}\hat{\mathbf{x}}) + \frac{\lambda_2}{2}\|\hat{\mathbf{x}}\|_2^2$. Indeed, we have

$$J_0(\hat{\mathbf{x}}) = \hat{b} + \lambda_0\|\hat{\mathbf{x}}\|_0 := g(\lambda_0).$$

An illustration, showing only strict minimizers with different supports which include the global ones, is provided in Figure 1 (left). The lower concave envelope of these affine functions, known as the ℓ_0-curve [5], is exactly defined by the minimizers $\hat{\mathbf{x}}$ belonging to the ℓ_0 path which we aim to estimate. This curve is piecewise affine, with singular points (indicated as $\hat{\lambda}_0^i,\ i = \{1, 2\}$ in Figure 1) corresponding to critical values of λ_0 where the support of the global solution changes.

2.2 The ℓ_0 Bregman relaxation (B-rex)

We recall the definition of B-rex and the sufficient conditions required to build an exact continuous relaxation of J_0.

Definition 1 (B-rex [9]) *Let* $\mathbf{x} \in \mathcal{C}^N$ *and* $\Psi(\mathbf{x}) = \sum_{n=1}^N \psi_n(x_n)$*, where* $\psi_n : \mathcal{C} \to \mathbb{R}$ *are strictly convex and twice differentiable functions. Then, the* ℓ_0 *Bregman relaxation (B-rex) is given by* $B_\Psi(\mathbf{x}) = \sum_{n=1}^N \beta_{\psi_n}(x_n)$*, with*

$$\beta_{\psi_n}(x) = \begin{cases} \psi_n(0) - \psi_n(x) + \psi_n'(\alpha_n^-)x, & \text{if } x \in [\alpha_n^-, 0], \\ \psi_n(0) - \psi_n(x) + \psi_n'(\alpha_n^+)x, & \text{if } x \in [0, \alpha_n^+], \\ \lambda_0, & \text{otherwise.} \end{cases}$$

where, the interval $[\alpha_n^-, \alpha_n^+] \ni 0$ *defines the* λ_0*-sublevel set of* $x \mapsto \psi_n(0) - \psi_n(x) + x\psi_n'(x)$.

Given a family Ψ, defining a B-rex requires solving the inequality $\psi_n(0) - \psi_n(x) + x\psi_n'(x) \le \lambda_0$. We showed in [9] that this can be done for common generating functions ψ_n found in the literature, such as the squared function, Shannon entropy and Kullback-Leibler divergence [9, 10]. A continuous relaxation of J_0 can be thus defined in terms of B-rex as:

$$J_\Psi(\mathbf{x}) = F_{\mathbf{y}}(\mathbf{A}\mathbf{x}) + B_\Psi(\mathbf{x}) + \frac{\lambda_2}{2}\|\mathbf{x}\|_2^2. \tag{2}$$

In Theorem 2, we recall a sufficient condition on Ψ under which J_Ψ is an exact relaxation of J_0, meaning that J_Ψ preserves the global minimizers of the original problem, while potentially removing some of the local minimizers.

Theorem 2 (**Exact relaxation property** [9, Theorem 9]) *Let* J_Ψ *be defined as in* (2)*. If for all* $n \in [N]$*,* $\mathbf{x} \in \mathbb{R}^N$ *and* $t \in (\alpha_n^-, 0) \cup (0, \alpha_n^+)$ *the following condition holds*

$$\frac{\partial^2}{\partial t^2} F_{\mathbf{y}}(\mathbf{A}(\mathbf{x}^{(n)} + t\mathbf{e}_n)) + \lambda_2 < \psi_n''(t), \tag{CC}$$

Then,

$$\underset{\mathbf{x}\in\mathcal{C}^N}{\operatorname{argmin}}\ J_\Psi(\mathbf{x}) = \underset{\mathbf{x}\in\mathcal{C}^N}{\operatorname{argmin}}\ J_0(\mathbf{x}), \tag{3}$$

$\hat{\mathbf{x}}$ *local minimizer of* $J_\Psi \implies \hat{\mathbf{x}}$ *local minimizer of* J_0.

Moreover, from [9, Proposition 10], a local minimizer $\hat{\mathbf{x}}$ of J_0, remains a local minimizer of J_Ψ if and only if

$$\forall n \in \sigma(\hat{\mathbf{x}}),\ \hat{x}_n \in \mathcal{C}\backslash[\alpha_n^-, \alpha_n^+], \tag{4}$$

$$\forall n \in \sigma^c(\hat{\mathbf{x}}),\ -\langle \mathbf{a}_n, \nabla F_{\mathbf{y}}(\mathbf{A}\hat{\mathbf{x}})\rangle \in \left[\ell_n^-, \ell_n^+\right], \tag{5}$$

where $\sigma(\hat{\mathbf{x}}) = \{n \in [N] : \hat{x}_n \neq 0\}$ is the support of $\hat{\mathbf{x}}$, and $\ell_n^\pm = -\psi_n'(0) + \psi_n'(\alpha_n^\pm)$. Note that the quantities $\alpha_n^\pm, \ell_n^\pm$ depend on λ_0. As opposed to J_0, the set of minimizers of J_Ψ depends on λ_0, a property that we want to exploit next.

3 The ℓ_0 path with B-rex

To quantify the extent to which the local minimizers of J_Ψ vary with λ_0 we have the following Proposition.

Proposition 3 *Let* $\hat{\mathbf{x}}$ *be a local minimizer of* J_0*. Then, there exist critical thresholds*

$$\underline{\lambda}_0(\hat{\mathbf{x}}) = \min_{n\in\sigma(\hat{\mathbf{x}})} \left\{\phi_n^{-1}(|\hat{x}_n|)\right\}, \tag{6}$$

$$\bar{\lambda}_0(\hat{\mathbf{x}}) = \max_{n\in\sigma(\hat{\mathbf{x}})^c} \xi_n^{-1}\left(|\langle \mathbf{a}_n, \nabla F_{\mathbf{y}}(\mathbf{A}\hat{\mathbf{x}})\rangle|\right) \tag{7}$$

such that for all $\lambda_0 \in [\underline{\lambda}_0(\hat{\mathbf{x}}), \bar{\lambda}_0(\hat{\mathbf{x}})]$*,* $\hat{\mathbf{x}}$ *is a critical point of* J_Ψ*. In the above two equations,* $\phi_n(\lambda_0) = \max\left(|\alpha_n^-(\lambda_0)|, \alpha_n^+(\lambda_0)\right)$, $\xi_n(\lambda_0) = \min\{|\ell_n^-(\lambda_0)|, \ell_n^+(\lambda_0)\}$*, and* $\ell_n^\pm = -\psi_n'(0) + \psi_n'(\alpha_n^\pm)$.

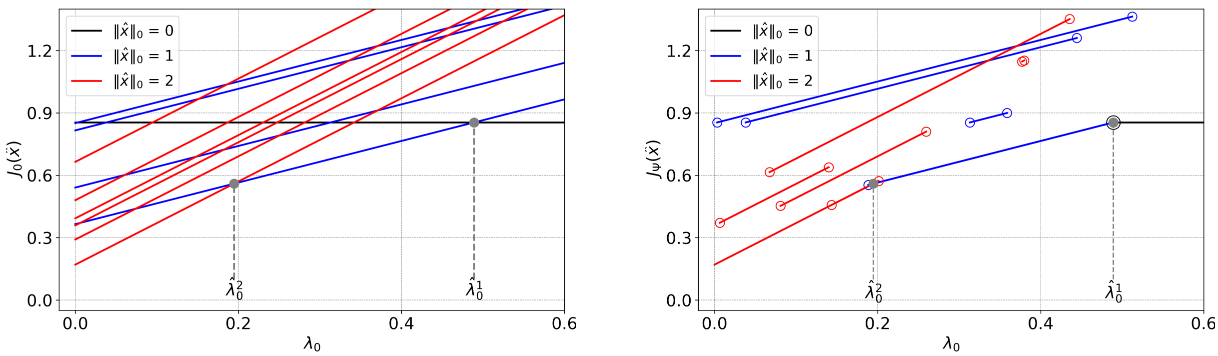

Figure 1 – Representation of strict local minimizers of J_0 and J_Ψ versus λ_0. Each line (left) (resp., segment (right)) is associated to a strict local minimizer $\hat{\mathbf{x}}$ of J_0 (resp., J_Ψ). The ℓ_0-curve corresponds to the lower envelope of this set of lines (resp., segments).

Algorithm 1: Pseudo-code of `L0PathBrex`

Input: $k_{\max} \in \mathbb{N}, \varepsilon > 0, N_{\text{pass}} \in \mathbb{N}$, $\text{Algo}(\mathbf{x}, \lambda_0, \boldsymbol{\rho})$
Output: $\mathcal{S}, \Lambda$ ▷ ℓ_0-path made of $(\mathbf{x}, \lambda_0) \in \mathcal{S} \times \Lambda$
$\mathbf{x} = \mathbf{0}$;
for $p = 1$ *to* N_{pass} **do**
  Forward Pass
  while $\|\mathbf{x}\|_0 \leq k_{\max}$ **do**
    Choose $\lambda_0 \in (\underline{\lambda}_0(\mathbf{x}) - \varepsilon, \underline{\lambda}_0(\mathbf{x}))$ ▷ using (6)
    $\mathbf{x}_{\text{next}} = \text{Algo}(\mathbf{x}, \lambda_0, \boldsymbol{\rho})$
    $\mathcal{S} \leftarrow \mathbf{x}_{\text{next}}$ and $\mathbf{x} = \mathbf{x}_{\text{next}}$
  end
  Backward Pass
  while $\|\mathbf{x}\|_0 > 1$ **do**
    Choose $\lambda_0 \in (\bar{\lambda}_0(\mathbf{x}), \bar{\lambda}_0(\mathbf{x}) + \varepsilon)$ ▷ using (7)
    $\mathbf{x}_{\text{next}} = \text{Algo}(\mathbf{x}, \lambda_0, \boldsymbol{\rho})$
    $\mathcal{S} \leftarrow \mathbf{x}_{\text{next}}$ and $\mathbf{x} = \mathbf{x}_{\text{next}}$
  end
end
$(\mathcal{S}, \Lambda) = \text{Prune}(\mathcal{S})$

The proof relies on conditions (4) and (5). In addition, it is worth mentioning that $\bar{\lambda}_0(\hat{\mathbf{x}}) = +\infty$ only for the local minimizer $\hat{\mathbf{x}} = \mathbf{0}$ (which corresponds to the global minimizer for sufficiently large λ_0) and $\underline{\lambda}_0(\hat{\mathbf{x}}) = 0$ only for local minimizers $\hat{\mathbf{x}}$ that belong to the solution set of the unpenalized problem (i.e.; Problem (1) with $\lambda_0 = 0$). Unlike the case of J_0, where local minimizers persist for all λ_0, each local minimizer $\hat{\mathbf{x}}$ of the exact relaxation J_Ψ defines a segment in the $(\lambda_0, J_\Psi(\hat{\mathbf{x}}))$-plane with support $[\underline{\lambda}_0(\hat{\mathbf{x}}), \bar{\lambda}_0(\hat{\mathbf{x}})]$ and slope $\|\hat{\mathbf{x}}\|_0$. As such, the graph in Figure 1 (left) built with J_0 is transformed to the (equivalent) one in Figure 1 (right) when exploiting J_Ψ.

Within this context, we propose to exploit the hyperparameter ranges $[\underline{\lambda}_0(\hat{\mathbf{x}}), \bar{\lambda}_0(\hat{\mathbf{x}})]$ to deploy warm-start strategies for estimating the ℓ_0 path. The rationale behind this idea is that a local minimizer $\hat{\mathbf{x}}$ with λ_0-range $[\underline{\lambda}_0(\hat{\mathbf{x}}), \bar{\lambda}_0(\hat{\mathbf{x}})]$ is likely to be a good initial point if $\lambda_0 \in (\underline{\lambda}_0(\hat{\mathbf{x}}) - \varepsilon, \underline{\lambda}_0(\hat{\mathbf{x}})) \cup (\bar{\lambda}_0(\hat{\mathbf{x}}), \bar{\lambda}_0(\hat{\mathbf{x}}) + \varepsilon)$.

Algorithm Description. The pseudo-code in Algorithm 1 illustrates the main idea of the proposed `L0PathBrex`, which exploits the interval where a minimizer of J_Ψ exists together with the warm-start strategy described above. This is performed sequentially through forward and backward passes in order to better explore the optimization landscape so as to refine the estimation of the ℓ_0 path. $\text{Algo}(\mathbf{x}_0, \lambda_0, \boldsymbol{\rho})$ solves the relaxed problem for a given initial point $\mathbf{x}_0$, a value λ_0, and a set of algorithmic parameters $\boldsymbol{\rho}$ ensuring the convergence of the algorithm. As a final step, from all the local minimizer gathered in $\mathcal{S}$, the function $\text{Prune}(\mathcal{S})$ extracts an estimate of the ℓ_0 path. This is achieved through the computation of the lower (concave) envelope of all affine functions associated to local minimizers obtained in $\mathcal{S}$. This discards points in $\mathcal{S}$ not involved in the estimated ℓ_0 path. Moreover, it allows us to identify the critical values of λ_0 at which the solution changes. These values are determined by the intersection of two affine functions with different slopes (see Figure 1), associated to two different local minimizers.

4 Numerical results

We now present the experimental validation of Algorithm `L0PathBrex` on ℓ_0-regularized LS ($F_{\mathbf{y}}(\mathbf{Ax}) = \frac{1}{2}\|\mathbf{Ax} - \mathbf{y}\|_2^2$) and LR ($F_{\mathbf{y}}(\mathbf{Ax}) = \sum_{m=1}^{M} \log\left(1 + \exp([\mathbf{Ax}]_m)\right) - y_m[\mathbf{Ax}]_m$) criteria.

Tested Algorithms. We compare `L0PathBrex` with two methods of the `L0Learn` package [11, 8]: 'CD', a cyclic coordinate descent, and 'CDPSI', which performs local combinatorial search on top of CD. Both algorithms provide a ℓ_0 path. In `L0PathBrex`, we consider two variants with different $\text{Algo}(\mathbf{x}, \lambda_0, \boldsymbol{\rho})$: the iteratively reweighted ℓ_1 (IRL1) [14] and the forward-backward splitting (FBS) algorithm [1].

Data generation. The instances $(\mathbf{A}, \mathbf{y})$ are generated following [11, 8] for LS and LR, respectively. For LS, we construct the rows of $\mathbf{A} \in \mathbb{R}^{M \times N}$, with $(M, N) = (500, 1000)$, as independent samples drawn from a multivariate normal distribution with zero mean and a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times N}$, where each entry is defined as $\Sigma_{ij} = \varrho^{|i-j|}$ for some $\varrho \in (0, 1)$. The observation vector is given by $\mathbf{y} = \mathbf{Ax}^* + \boldsymbol{\varepsilon}$, where $\mathbf{x}^* \in \mathbb{R}^N$ has k^* evenly spaced nonzero entries, each set to 1, and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. The signal-to-noise ratio (SNR) is defined as $\text{SNR} = \frac{\text{Var}(\mathbf{Ax}^*)}{\text{Var}(\boldsymbol{\varepsilon})} = \frac{(\mathbf{x}^*)^T \boldsymbol{\Sigma} \mathbf{x}^*}{\sigma^2}$.

For LR, $\mathbf{A}$ and $\mathbf{x}^*$ are generated similarly to the LS case. The label vector $\mathbf{y}$ is binary, with $y_m \in \{-1, 1\}$, where $y_m = 1$ is determined with probability $P(y_m = 1 \mid \mathbf{a}_m) = 1/1 + e^{-s\langle \mathbf{a}_m, \mathbf{x}^* \rangle}$. There, $\mathbf{a}_m$ denotes the m-th row of $\mathbf{A}$, and $s > 0$ controls the SNR.

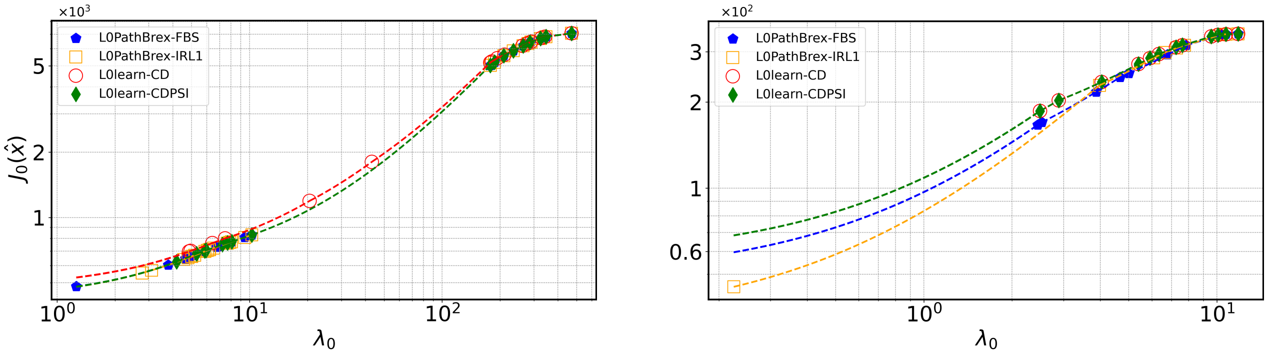

Figure 2 – ℓ_0-curves (log scale) associated to obtained ℓ_0 paths for LS (left) and LR (right). The parameters are $(k^*, \varrho, \mathrm{SNR}) = (25, 0.9, 10)$ for LS and $(k^*, \varrho, s) = (25, 0.9, 1)$ for LR. We ran `L0PathBrex` with $k_{\max} = 2k^*$, $N_{\text{pass}} = 20$, and $\lambda_2 = 0$ for LS and $\lambda_2 = 0.01$ for LR.

$F_{\mathbf{y}}$	IRL1	FBS	CD	CDPSI
LS	1.0010	1.0011	1.0088	**1.0004**
LR	**1.0272**	1.0463	1.0583	1.0574

Table 1 – Normalized ℓ_0-curve area (20 of instances of $(\mathbf{A}, \mathbf{y})$).

Results. In Figure 2, we present the ℓ_0-curve associated to the obtained ℓ_0 paths. Note that the pruning step $\mathrm{Prune}(\mathcal{S})$ in Algorithm 1 is also applied to the set of solutions given by CD and CDPSI. We use a logarithmic scale for better visualization, which results in the loss of concavity in the ℓ_0-curve. In the LS case (left panel of Figure 2), algorithm `L0PathBrex`, with both IRL1 and FBS, as well as CDPSI, produces similar ℓ_0-curves, which are clearly better than the one obtained by CD. For the LR case (right panel of Figure 2), we observe that `L0PathBrex`, with both IRL1 and FBS, achieves a more optimal ℓ_0-curve compared to both CD and CDPSI, leading to solutions with the lowest objective value for J_0. Furthermore, in this case, IRL1 outperforms FBS.

Next, we consider 20 instances of $(\mathbf{A}, \mathbf{y})$ generated under the same parametric setting. We use the area under the ℓ_0-curve as an assessment metric (the smaller, the better regularization path). In Table 1, we present the average of the normalized area: for each data generation we normalized the area obtained by each method with the minimal one. We observe that for LS, the CDPSI gives the better path, followed by the proposed `L0PathBrex` with IRL1 and FBS and then CD, which always attaints the worst path. In constrast, for LR, `L0PathBrex` with IRL1 leads to the best performance among the four tested methods.

Regarding computational time, while `L0Learn` computes an entire solution path within milliseconds, the proposed `L0PathBrex` require few minutes. Yet, this must be interpreted with care, as `L0Learn` is implemented in C++, whereas `L0PathBrex` is written in `Python`. Moreover, the computational time is also influenced by the choice of the inner optimization routine.

5 Conclusion

In this paper, we proposed a new algorithm (`L0PathBrex`) to estimate the ℓ_0 path in ℓ_0-regularized optimization problems. Our methodology leverages the properties of the Bregman exact continuous relaxations of the original problem to deploy warm-start strategies in a tree-search fashion. Numerical experiments show that the proposed method achieves similar results to the CDPSI method of the `L0Learn` package on LS problems while outperforming it for LR problems.

Acknowledgments. The authors acknowledge the ANR EROSION (ANR-22-CE48-0004) and ERC MALIN (grant 101117133).

References

[1] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Math. Prog.*, 2013.

[2] W. Bian and X. Chen. A Smoothing Proximal Gradient Algorithm for Nonsmooth Convex Regression with Cardinality Penalty. *SINUM*, 2020.

[3] P. Bühlmann and S. Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011.

[4] D. Brie C. Soussen, J. Idier and J. Duan. From bernoulli–gaussian deconvolution to sparse signal restoration. *IEEE TSP*, 2011.

[5] J. Duan C. Soussen, J. Idier and D. Brie. ℓ_2-ℓ_0 regularization path tracking algorithms. *IEEE TSP*, 2014.

[6] A. C. Cameron and P. K. Trivedi. *Regression analysis of count data*. Cambridge University Press, 2013.

[7] M. Carlsson. On Convex Envelopes and Regularization of Non-convex Functionals Without Moving Global Minima. *JOTA*, 2019.

[8] A. Dedieu, H. Hazimeh, and R. Mazumder. Learning sparse classifiers: Continuous and mixed integer optimization perspectives. *JMLR*, 2021.

[9] M. Essafri, L. Calatroni, and E. Soubies. Exact Continuous Relaxations of ℓ_0-Regularized Criteria with Non-quadratic Data Terms. *arXiv:2402.06483*, 2024.

[10] M. Essafri, L. Calatroni, and E. Soubies. On ℓ_0 Bregman-Relaxations for Kullback-Leibler Sparse Regression. In *Proc. IEEE MLSP*, 2024.

[11] H. Hazimeh and R. Mazumder. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Oper. Res.*, 2020.

[12] Y. Liu, S. Bi, and S. Pan. Equivalent lipschitz surrogates for zero-norm and rank optimization problems. *J. Glob. Optim.*, 2018.

[13] T. T. Nguyen, C. Soussen, J. Idier, and E-H. Djermoune. NP-hardness of ℓ_0 minimization problems: revision and extension to the non-negative setting. In *Proc. SAMPTA*, Bordeaux, 2019.

[14] P Ochs, A Dosovitskiy, T Brox, and T Pock. On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIIMS*, 2015.

[15] T. Hastie R. Mazumder, J. Friedman. Sparsenet: Coordinate descent with nonconvex penalties. *J. Am. Stat. Assoc.*, 2010.

[16] E. Soubies, L. Blanc-Féraud, and G. Aubert. A Continuous Exact ℓ_0 Penalty (CEL0) for Least Squares Regularized Problem. *SIIMS*, 2015.