Newton Method for Sparse Logistic Regression: Quadratic Convergence and Extensive Simulations

Article · February 2021

CITATIONS
4

READS
262

3 authors, including:

Shenglong Zhou
University of Southampton
35 PUBLICATIONS **101** CITATIONS
SEE PROFILE

Naihua Xiu
Beijing Jiaotong University
178 PUBLICATIONS **2,044** CITATIONS
SEE PROFILE

Newton Method for Sparse Logistic Regression: Quadratic Convergence and Extensive Simulations*

Rui Wang[a], Naihua Xiu[a], Shenglong Zhou[b,*]

[a]*Department of Applied Mathematics, Beijing Jiaotong University, Beijing, China*
[b]*School of Mathematics, University of Southampton, Southampton, UK*

Abstract

Sparse logistic regression, as an effective tool of classification, has been developed tremendously in recent two decades, from its origination the ℓ_1-regularized version to the sparsity constrained models. This paper is carried out on the sparsity constrained logistic regression by the Newton method. We begin with establishing its first-order optimality condition associated with a τ-stationary point. This point can be equivalently interpreted as an equation system which is then efficiently solved by the Newton method. The method has a considerably low computational complexity and enjoys global and quadratic convergence properties. Numerical experiments on random and real data demonstrate its superior performance when against seven state-of-the-art solvers.

Keywords: Sparse logistic regression, Newton method, global and quadratic convergence, numerical experiments

1. Introduction

As one of effective tools of classification, logistic regression has its high reputation with extensive applications ranging from machine learning, data mining, pattern recognition, medical science to statistics. It describes the relationship

*This work is supported by the National Natural Science Foundation of China (11971052) and Beijing Natural Science Foundation (Z190002).

*Corresponding author

Email addresses: wangruibjtu@bjtu.edu.com (Rui Wang), nhxiu@bjtu.edu.cn (Naihua Xiu), shenglong.zhou@soton.ac.uk (Shenglong Zhou)

between a sample data $\mathbf{x}$ and its associated binary response/label $y \in \{0, 1\}$ through the conditional probability

$$\Pr(y|\mathbf{x}, \mathbf{z}) = \left[1 + e^{-\langle \mathbf{x}, \mathbf{z} \rangle}\right]^{-1}, \tag{1}$$

where $\Pr(y|\mathbf{x}, \mathbf{z})$ is the conditional probability of the label y, given the sample $\mathbf{x}$ and a parameter vector $\mathbf{z}$, and $\langle \mathbf{x}, \mathbf{z} \rangle$ is the vector inner product. To find the maximum likelihood estimate of the parameter $\mathbf{z}$, a set of n independently and identically distributed samples $\{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, n\}$ are first drawn, where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$, yielding a joint likelihood of the interested parameter/classifier $\mathbf{z}$. Then the maximum likelihood estimate is obtained by minimizing the classical logistic regression loss function,

$$\ell(\mathbf{z}) := \frac{1}{n} \sum_{i=1}^{n} \left[\ln(1 + e^{\langle \mathbf{x}_i, \mathbf{z} \rangle}) - y_i \langle \mathbf{x}_i, \mathbf{z} \rangle\right]. \tag{2}$$

The logistic loss function is strictly convex and thus admits a unique minimizer provided that the sample matrix is full row rank. Therefore, the minimization performs relatively well when the number of samples is larger than the number of features, i.e., $n \geq p$. But, the case $n < p$ may lead to an over-fitting: the solved classifier through minimizing (2) well fits the model (making the loss sufficiently small) on training data but behaves poorly on unseen data.

On the one hand, the case $n < p$ occurs often in many real applications. For instance, one piece of gene expression data sample is made of thousands of genes whilst common medical equipments are only able to obtain very limited samples. In image processing, an image consists of large amounts of pixels, which is far more than the number of observed images. One the other hand, despite numerous features in those data, there is only a small portion that is of importance. For example, apart from the classification task, the micro-array data experiments also attempt to identify a small set of informative genes (to distinguish the tumour and the normal tissues) in each gene expression data so as to remove the irrelevant genes to simplify the inference. This naturally gives rise to the topic of the sparse logistic regression.

1.1. Sparse logistic regression

Sparse logistic regression (SLR) was originated from the ℓ_1-regularized logistic regression proposed by [1],

$$\min_{\mathbf{z}\in\mathbb{R}^p} \quad \ell(\mathbf{z}) + \nu\|\mathbf{z}\|_1, \tag{3}$$

where $\|\mathbf{z}\|_1$ is the ℓ_1-norm and $\nu > 0$. Under the help of ℓ_1-regularization, this model is capable of rendering a sparse solution allowing for capturing key features among others. A vector is called sparse if only a few entries are non-zero and the rest are zeros. With the advance in sparse optimization in recent decade, (3) has been extensively extended to the following general model,

$$\min_{\mathbf{z}\in\mathbb{R}^p} \quad \ell_\phi(\mathbf{z}) := \ell(\mathbf{z}) + \phi_\nu(\mathbf{z}), \tag{4}$$

where the regularized function $\phi_\nu(\mathbf{z}) : \mathbb{R}^p \to \mathbb{R}$ is designed to pursue a sparse solution and associated with some given non-negative parameters ν.

An alternative is to consider logistic regression with a sparsity constraint, which was first studied in [2, 3] separately and then well investigated in [4]. They perform the following sparsity constrained logistic regression

$$\min_{\mathbf{z}\in\mathbb{R}^p} \quad \ell(\mathbf{z}), \quad \text{s.t.} \quad \|\mathbf{z}\|_0 \leq s, \tag{5}$$

where $\|\mathbf{z}\|_0$ is the ℓ_0 pseudo norm of $\mathbf{z}$, counting the number of non-zero elements of $\mathbf{z}$. The discreteness of the sparsity constraint makes tackling this model NP-hard. Nevertheless, compared with the regularized model, the sparsity constrained version enjoys various appealing features, such as being penalty parameter-free, ease of sparsity controlling, and low computational complexity in terms of numerical computation and to name a few.

Therefore, the generalization of the problem (5), where $\ell(\mathbf{z})$ is replaced by a more general function, has been thoroughly investigated in [5, 6] since it was first introduced by [2] and [7]. Particularly, in statistics, the model with the logistic loss function being replaced by the least squares of linear regression is the so-called best subspace/feature selection [8, 9, 10, 11, 12]. Those research

bring fruitful results and provide a series of effective numerical tools to conquer the NP-hardness.

However, as stated in [2] that '*one can achieve arbitrarily small loss values by tending the parameters to infinity along certain directions*' for (5), authors [2] suggests to address the following regularized model

$$\min \quad f(\mathbf{z}) := \ell(\mathbf{z}) + (\lambda/2)\|\mathbf{z}\|_2^2, \quad \text{s.t.} \quad \|\mathbf{z}\|_0 \leq s, \tag{6}$$

where $\lambda > 0$ is a given penalty parameter. Now the objective function f is strongly convex and thus (6) admits finitely many (local or global) bounded minimizers. So the work in this paper is carried out along with this model.

1.2. Methods of solving SLR

Since there is a vast body of methods that have been proposed to deal with the sparse optimization problems containing the SLR as a special case, we present a brief overview of methods processing (4)-(6) directly.

Regularization methods. Most versions of the model (4) are unconstrained and continuous. Then generic optimization methods, known as the relaxation (regularization) methods from the perspective of optimization, are tractable. Dependent on the convexity of the penalty functions ϕ_ν, those methods can be summarized into two categories.

Convex regularizations are mainly associated with the usage of ℓ_1-norm:

- $\phi_\nu(\mathbf{z}) = \nu\|\mathbf{z}\|_1$. Some earliest work can be traced back to [13, 14], where expectation maximization methods were developed. Later relevant work can be found in [15, 16, 17, 18, 19].

- $\phi_\nu(\mathbf{z}) = \nu_1\|\mathbf{z}\|_2^2 + \nu_2\|\mathbf{z}\|_1$, where $\nu = [\nu_1, \nu_2] > 0$. For this penalty, two powerful packages `SLEP` [20] and `GLMNET` [21, 22] have been created.

- $\phi_\nu(\mathbf{z}) = \nu\|\mathbf{z}\|^2 + \delta_{\|\mathbf{z}\|_1 \leq t}(\mathbf{z})$, where $t > 0$ is a given parameter, and $\delta_{\|\mathbf{z}\|_1 \leq t}(\mathbf{z}) = 0$ if $\|\mathbf{z}\|_1 \leq t$ and $+\infty$ otherwise. Such a problem can be addressed by `Lassplore` [23] or `SLEP` [20]. When $\nu = 0$, the above model

is the ℓ_1 constrained logistic regression, which was addressed by `IRLS-LARS` in [24]. Here, `LARS` was adopt from [25].

Nonconvex regularizations differ slightly. In the early stage, scholars from statistics have proposed a number of excellent methods including the smoothly clipped absolute deviation (`SCAD` [26]), one step local linear approximation [27] and the group bridge method for multiple regression problems [28]. Then, a general iterative shrinkage and thresholding algorithm (`GIST`) has been proposed in [29]. Recently, the accelerated proximal gradient method (`APG`) in [30], the efficient hybrid optimization algorithm for non-convex regularized problems (`HONOR`) in [31] and the proximal Newton method based on the scheme of the difference of two convex functions in [32] are worth exploring.

Greedy Methods. An impressive body of approaches have been developed to solve the sparsity constrained models (5) or (6). The first work in [2] generalized the compressive sampling matching pursuit [33] to derive the gradient support pursuit (`GraSP`). Then authors in [34] adopted the orthogonal matching pursuit (`OMP` [35]) to develop a group `OMP` method. Other relating methods can be seen those in [36, 37, 38]. Very lately, three effective Newton type methods have been designed. They are the Newton greedy pursuit method `NTGP` in [39], greedy projected gradient-Newton method (`GPGN` [4]) and the fast Newton hard thresholding pursuit [40]. In particular, we would like to mention the methods, the zero-CW search method and the full-CW search method, proposed in [5]. Both methods first carefully search an index set T and then solve a subproblem where the variable has support within T to update the next point.

1.3. Our contributions

Those aforementioned methods have been testified to have the excellent numerical performance to deal with (5) or (6). However, only a very few of them established strong theoretical guarantees (such as global convergence property or quadratic convergence rate) from the perspective of deterministic optimization. Therefore, in this paper, we aim to develop a second-order method that

possesses such strong theoretical guarantees. The main contributions are summarized as follows.

C1) We start with establishing the optimality condition of the model (6) by introducing a τ-stationary point (see Definition 2.2 for more details) which turns out to be at least a locally optimal solution by Theorem 2.3. More importantly, a τ-stationary point draws forth a system of equations (17) that makes the classic Newton method applicable.

C2) Differing with any of the above mentioned algorithms, we perform the Newton method on solving a system of equations (17), one of the optimality conditions of the problem (6). The proposed Newton method dubbed as `NSLR`, an abbreviation for the Newton method for the SLR, has a simple framework (see Algorithm 1) that makes its implementation easy and has a low computational complexity per each iteration. Such a low computational complexity is due to a small-scale linear equation system with s variables and s equations being solved to update the Newton direction.

C3) It is worth mentioning that the standard Newton-type methods derive the directions for a fixed system of equations. However, in each iteration, the system of equations (17) varies when the index set α changes. Consequently, `NSLR` updates Newton directions on unfixed systems of equations. Because of this, some common approaches to establish the convergence results of Newton-type methods for solving a fixed system of equations fail to be employed for `NSLR`. Nevertheless, we still show that the whole sequence generated by `NSLR` converges to a τ-stationary point, at least a locally optimal solution. Moreover, the convergence enjoys a quadratic rate, well testifying the proposed method would perform extraordinarily theoretically.

C4) Finally, the efficiency of `NSLR` is demonstrated against seven state-of-the-art methods by solving a number of randomly generated and real datasets. The fitting accuracy and computational speed are very competitive. Especially, in high dimensional data setting, `NSLR` outperforms the others in terms of the computational time.

We note that there are some methods that also have a close link to τ-stationary point, such as two methods in [5] and `GPGN` [4]. We would like to highlight the difference between them and `NSLR`. For the methods in [5], since an optimal solution to a subproblem needs to be found to update the next point in each step, the methods can terminate within finitely many steps. However, `NSLR` updates the next point by Newton direction with a line search scheme and has been shown to enjoy the global and quadratic convergence properties. For `GPGN`, the procedure `IHT` from [7] to update the next point dominates most steps, and Newton steps are imposed only when two consecutive points have the same support sets. It is shown to converge quadratically only when the solution has s nonzeros but sublinearly otherwise. By contrast, `NSLR` always performs Newton step to update the next point and converges quadratically without additional assumptions. Moreover, differing from papers [5] and [4] where comprehensive optimality conditions have been investigated, the primary aim of this paper is to develop a Newton-type method and establish its convergence properties.

1.4. Organization and notation

This paper is organized as follows. To explore the optimality conditions of (6), the next section introduces the τ-stationary point by Definition 2.2 and establishes its relationships with a local/global minimizer in Theorem 2.3. This τ-stationary point is then equivalently transferred to an equation system (17). Section 3 develops the method `NSLR`, an abbreviation for Newton method for SLR, which turns out to have a simple algorithmic framework and low computational complexity. The global and quadratic convergence properties of the method are then established. In Section 4, the superior performance of `NSLR` is demonstrated against some of the state-of-the-art solvers on randomly generated and real datasets in high dimensional scenarios. Concluding remarks are made in the last section.

We end this section by defining some notation employed throughout this paper. Let $X := [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times p}$ be the sample matrix and $\mathbf{y} = (y_1, y_2, \cdots, y_n)^\top \in \mathbb{R}^n$ be the response vector. For an index set $\alpha \subseteq [p] :=$

$\{1, 2, \cdots, p\}$, let $|\alpha|$ be the cardinality of α and $\overline{\alpha} := [p] \backslash \alpha$ be the complementary set of α. The support set of a vector $\mathbf{z}$ is denoted by $\mathrm{supp}(\mathbf{z}) := \{i \in [p] : z_i \neq 0\}$. We denote $[\mathbf{z}]_i^{\downarrow}$ the ith largest (in absolute) elements of $\mathbf{z}$. Write $\mathbf{z}_\alpha \in \mathbb{R}^{|\alpha|}$ as the sub-vector of $\mathbf{z}$ containing elements indexed on α. Similarly, for a matrix $A \in \mathbb{R}^{p \times p}$, $A_{\alpha\beta}$ is the sub-matrix containing rows indexed on α and columns indexed on β, particularly, $A_{\alpha:} = A_{\alpha[p]}$ if $\beta = [p]$. Let $\|\cdot\|$ denote the Spectral norm for a matrix and Euclidean norm for a vector respectively. Furthermore, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are the minimal and maximal eigenvalues of A.

2. Optimality

This section is devoted to investigate the optimality conditions of (6), before which we summarize some properties of the objective function $\ell(\mathbf{z})$ from [4].

Proposition 2.1 (Lemma 2.2-2.4, Lemma A.3 [4]). *The function $\ell(\mathbf{z})$ is twice continuously differentiable and has the following basic properties:*

i) It is non-negative, convex and strongly smooth on $\mathbb{R}^p$ with a parameter

$$\lambda_{\mathrm{x}} := \lambda_{\max}(X^\top X)/(4n).$$

ii) The gradient is Lipschitz continuous with the Lipschitz constant λ_{x}.

iii) The Hessian matrix is Lipschitz continuous with the Lipschitz constant $M := 12\lambda_{\mathrm{x}} \max_{i \in [n]} \|\mathbf{x}_i\|_1$.

These properties of $\ell(\mathbf{z})$ are also enjoyed by the function $f(\mathbf{z}) = \ell(\mathbf{z}) + (\lambda/2)\|\mathbf{z}\|^2$. Since the proofs are easy, we only summarize them here. The function f is strongly convex with a constant λ, and strongly smooth with a parameter $L := \lambda + \lambda_{\mathrm{x}}$, namely, for any $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^p$,

$$f(\mathbf{z}) \geq f(\mathbf{z}') + \langle \nabla f(\mathbf{z}'), \mathbf{z} - \mathbf{z}' \rangle + (\lambda/2)\|\mathbf{z} - \mathbf{z}'\|^2, \tag{7}$$

$$f(\mathbf{z}) \leq f(\mathbf{z}') + \langle \nabla f(\mathbf{z}'), \mathbf{z} - \mathbf{z}' \rangle + (L/2)\|\mathbf{z} - \mathbf{z}'\|^2. \tag{8}$$

The gradient is Lipschitz continuous with the Lipschitz constant L, namely,

$$\|\nabla f(\mathbf{z}) - \nabla f(\mathbf{z}')\| \leq L\|\mathbf{z} - \mathbf{z}'\|^2,$$

for any $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^p$. The Hessian matrix is Lipschitz continuous with the Lipschitz constant M, namely, for any $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^p$,

$$\|\nabla^2 f(\mathbf{z}) - \nabla^2 f(\mathbf{z}')\| \leq M\|\mathbf{z} - \mathbf{z}'\|^2. \tag{9}$$

When it comes to characterize the solutions of the problem (6), we need the projection of a vector $\mathbf{z} \in \mathbb{R}^p$ onto the feasible region defined by

$$\Pi_s(\mathbf{z}) := \text{argmin}_{\mathbf{x}\in\mathbb{R}^p}\{\|\mathbf{z} - \mathbf{x}\| : \|\mathbf{x}\|_0 \leq s\},$$

which sets all but s largest absolute value components of $\mathbf{z}$ to zero. Since the right hand side may have multiple solutions, $\Pi_s(\mathbf{z})$ is a set. Based on the projection, we introduce the concept of the τ-stationary point which is also known as the L stationary point in [7, Definition 2.3].

Definition 2.2. *[7, Definition 2.3] A vector $\mathbf{z}$ is called a τ-stationary point of (6) if there is a $\tau > 0$ such that*

$$\mathbf{z} \in \Pi_s\left(\mathbf{z} - \tau\nabla f(\mathbf{z})\right). \tag{10}$$

By [7, Lemma 2.2], $\mathbf{z}$ is a τ-stationary point if and only if

$$\|\mathbf{z}\|_0 \leq s, \quad \tau\left|(\nabla f(\mathbf{z}))_i\right| \begin{cases} = & 0, \quad i \in \text{supp}(\mathbf{z}), \\ \leq & [\mathbf{z}]_s^{\downarrow}, \quad i \notin \text{supp}(\mathbf{z}). \end{cases} \tag{11}$$

Based on the definition of the τ-stationary point, our first main result is establishing its relationships with a locally/globally optimal solution to (6).

Theorem 2.3. *The following results hold for the problem (6).*

i) *A global minimizer is a τ-stationary point $\mathbf{z}^*$ for any $0 < \tau < 1/L$.*

ii) *A τ-stationary point $\mathbf{z}^*$ for some $\tau > 0$ is a unique local minimizer if $\|\mathbf{z}^*\|_0 = s$ and a unique global minimizer if $\|\mathbf{z}^*\|_0 < s$.*

iii) *A τ-stationary point for some $\tau > 1/\lambda$ is a unique global minimizer.*

Proof i) The proof is the same as that in [7, Theorem 2.2].

ii) Let $\mathbf{z}^*$ be a τ-stationary point for some $\tau > 0$. Then we have (11), i.e.,

$$\tau\,|(\nabla f(\mathbf{z}^*))_i| \begin{cases} = & 0, \qquad i \in \mathrm{supp}(\mathbf{z}^*) =: \alpha_*, \\ \leq & [\mathbf{z}^*]_s^{\downarrow}, \quad i \notin \mathrm{supp}(\mathbf{z}^*). \end{cases} \tag{12}$$

For the case $\|\mathbf{z}^*\|_0 = s$, consider a local region $N(\mathbf{z}^*) := \{\mathbf{z} : \|\mathbf{z} - \mathbf{z}^*\| < [\mathbf{z}^*]_s^{\downarrow}\}$. Then for any feasible point $\mathbf{z} \in N(\mathbf{z}^*)$ and any $i \in \alpha_*$, we have $|z_i| \geq |z_i^*| - |z_i^* - z_i| > |z_i^*| - [\mathbf{z}^*]_s^{\downarrow} \geq 0$, which means $\alpha_* \subseteq \mathrm{supp}(\mathbf{z})$. Since $\|\mathbf{z}\|_0 \leq s = |\alpha_*|$, it holds $\alpha_* = \mathrm{supp}(\mathbf{z})$ for any feasible point $\mathbf{z} \in N(\mathbf{z}^*)$, namely $\mathbf{z}_{\overline{\alpha}_*} = \mathbf{z}^*_{\overline{\alpha}_*} = 0$. Then the strong convexity of f in (7) leads to

$$\begin{aligned} 2f(\mathbf{z}) - 2f(\mathbf{z}^*) &\geq 2\langle \nabla f(\mathbf{z}^*), \mathbf{z} - \mathbf{z}^* \rangle + \lambda \|\mathbf{z} - \mathbf{z}^*\|^2 && (13) \\ &= 2\langle (\nabla f(\mathbf{z}^*))_{\overline{\alpha}_*}, (\mathbf{z} - \mathbf{z}^*)_{\overline{\alpha}_*} \rangle + \lambda \|\mathbf{z} - \mathbf{z}^*\|^2 \\ &\overset{(12)}{=} \lambda \|\mathbf{z} - \mathbf{z}^*\|^2. && (14) \end{aligned}$$

Thus $\mathbf{z}^*$ is a unique local minimizer of (6).

For the case $\|\mathbf{z}^*\|_0 < s$, the condition (12) implies $\nabla f(\mathbf{z}^*) = 0$ due to $[\mathbf{z}^*]_s^{\downarrow} = 0$. Then (14) is true for any $\|\mathbf{z}\|_0 \leq s$. So $\mathbf{z}^*$ is a unique global minimizer of the problem (6).

iii) Let $\mathbf{z}^*$ be a τ-stationary point for some $\tau > 0$. Then $\mathbf{z}^* \in \Pi_s\left(\mathbf{z}^* - \tau \nabla f(\mathbf{z}^*)\right)$, which together with the definition of the projection Π_s implies that

$$\|\mathbf{z}^* - (\mathbf{z}^* - \tau \nabla f(\mathbf{z}^*))\|^2 \leq \|\mathbf{z} - (\mathbf{z}^* - \tau \nabla f(\mathbf{z}^*))\|^2.$$

This leads to

$$2\langle \nabla f(\mathbf{z}^*),\ \mathbf{z} - \mathbf{z}^* \rangle \geq -(1/\tau)\|\mathbf{z} - \mathbf{z}^*\|^2.$$

The above condition together with (13) derives

$$2f(\mathbf{z}) \geq 2f(\mathbf{z}^*) + (\lambda - 1/\tau)\|\mathbf{z} - \mathbf{z}^*\|^2,$$

which shows the unique global optimality of $\mathbf{z}^*$ if $\tau > 1/\lambda$. □

We note that the necessary optimality condition in Theorem 2.3 i) is directly adopted from [7, Theorem 2.2] or [5, Theorem 5.3] where $\mathbb{B} = \mathbb{R}^n$. However, we also establish the sufficient optimality conditions, see Theorem 2.3 ii) and

iii). The above relationships show that a τ-stationary point is at least a unique locally optimal solution to the problem (6). This allows us to focus on a τ-stationary point itself to pursue a 'good' solution. Therefore, we define a set

$$\Sigma_s(\mathbf{z}) := \{\ \alpha \in [p] :\ |\alpha| = s,\ |z_i| \geq |z_j|,\ \forall\ i \in \alpha,\ j \in \overline{\alpha}\ \}. \tag{15}$$

Each element α in $\Sigma_s(\mathbf{z})$ coincides the indices of the first s largest (in absolute) components of $\mathbf{z}$. Note that $\Sigma_s(\mathbf{z})$ may have multiple elements. For instance, $\mathbf{z} = (3, -2, 2, 1, 0)^\top$, $\Sigma_3(\mathbf{z}) = \{\{1, 2, 3\}\}$ and $\Sigma_2(\mathbf{z}) = \{\{1, 2\}, \{1, 3\}\}$. The notation allows us to rewrite $\Pi_s(\mathbf{z})$ as follows

$$\Pi_s(\mathbf{z}) = \{(\mathbf{z}_\alpha^\top\ 0)^\top :\ \alpha \in \Sigma_s(\mathbf{z})\}. \tag{16}$$

Then a point satisfying (10) can be interpreted as that there is an $\alpha \in \Sigma_s(\mathbf{z} - \tau\nabla f(\mathbf{z}))$ satisfying $\mathbf{z}_\alpha = (\mathbf{z} - \tau\nabla f(\mathbf{z}))_\alpha$ and $\mathbf{z}_{\overline{\alpha}} = 0$, which is equivalent to

$$(\nabla f(\mathbf{z}))_\alpha = 0, \qquad \mathbf{z}_{\overline{\alpha}} = 0. \tag{17}$$

Therefore, to find a τ-stationary point of (6), one can seek for a solution to the equation system (17). This is summarized into the following theorem.

Theorem 2.4. *A point $\mathbf{z}$ is a τ-stationary point of (6) if and only if there is an $\alpha \in \Sigma_s(\mathbf{z} - \tau\nabla f(\mathbf{z}))$ satisfying (17).*

3. Newton Method

In this section, we turn our attention to solve the equations (17) to pursue a τ-stationary point of the problem (6), at least a unique local minimizer.

Given a point $\mathbf{z}^k$, for notational convenience, let

$$H^k := \nabla^2 f(\mathbf{z}^k), \quad \boldsymbol{g}^k := \nabla f(\mathbf{z}^k). \tag{18}$$

3.1. The framework

Suppose we have a point $\mathbf{z}^k$ computed already. Then we can pick an index set α from $\Sigma_s(\mathbf{z}^k - \tau\boldsymbol{g}^k)$. For such a fixed index set α, we apply Newton step

on the equations (17) just once to derive the Newton direction by

$$\begin{bmatrix} H^k_{\alpha\alpha} & H^k_{\alpha\overline{\alpha}} \\ 0 & I_{p-s} \end{bmatrix} \mathbf{d}^k = - \begin{bmatrix} \boldsymbol{g}^k_\alpha \\ \mathbf{z}^k_{\overline{\alpha}} \end{bmatrix} =: -\theta^k_\alpha, \tag{19}$$

where $\mathbf{d}^k$ can be calculated explicitly by

$$\begin{cases} H^k_{\alpha\alpha}\mathbf{d}^k_\alpha &= H^k_{\alpha\overline{\alpha}}\mathbf{z}^k_{\overline{\alpha}} - \boldsymbol{g}^k_\alpha, \\ \mathbf{d}^k_{\overline{\alpha}} &= -\mathbf{z}^k_{\overline{\alpha}}. \end{cases} \tag{20}$$

Since f is strongly convex, H^k is non-singular and so are its any principal submatrices, i.e., $H^k_{\alpha\alpha}$ is invertible for any k and any α. Now we have the direction. If the full Newton step size is adopted, i.e., $\mathbf{z}^{k+1} = \mathbf{z}^k + \mathbf{d}^k$, then (20) implies

$$\|\mathbf{z}^{k+1}\|_0 = \|\mathbf{z}^{k+1}_\alpha\|_0 \leq |\alpha| = s.$$

Therefore, the updated point $\mathbf{z}^{k+1}$ is feasible to the problem (6). However, the full Newton step size generally does not guarantee the descent property of the objective function, that is, $f(\mathbf{z}^{k+1}) \leq f(\mathbf{z}^k)$ can not be ensured. To overcome such a drawback, we exploit the following operator

$$\mathbf{z}^k(\sigma) := \begin{bmatrix} \mathbf{z}^k_\alpha + \sigma\mathbf{d}^k_\alpha \\ \mathbf{z}^k_{\overline{\alpha}} + \mathbf{d}^k_{\overline{\alpha}} \end{bmatrix} = \begin{bmatrix} \mathbf{z}^k_\alpha + \sigma\mathbf{d}^k_\alpha \\ 0 \end{bmatrix}. \tag{21}$$

For some carefully chosen σ_k, we set $\mathbf{z}^{k+1} := \mathbf{z}^k(\sigma_k)$. Then we will show that in this way, $\mathbf{z}^{k+1}$ is not only always feasible to the problem (6) but also satisfies the descent property (see Lemma 3.3). Now we summarize the whole framework of Newton method in Algorithm 1.

Remark 3.1. *Regarding* `NSLR`*, we have some comments.*

i) To pick an $\alpha \in \Sigma_s(\mathbf{z}^k - \tau\boldsymbol{g}^k)$, only s largest elements (in absolute) are selected, which enables us to use a MATLAB built-in function `mink`*. The computational complexity is about $\mathcal{O}(p + s\log s)$.*

ii) Updating $\mathbf{d}^k$ by (20) involves two main calculations $H^k_{\alpha\alpha}$ with $|\alpha| = s$ and its inverse. Their computational complexities are $\mathcal{O}(sn + s^2n)$ and $\mathcal{O}(s^3)$.

Algorithm 1 `NSLR`: Newton method for SLR

1: **Initialize** $\mathbf{z}^0, \tau > 0, c \in (0,1)$. Set $k := 0$.

2: **while** the halting condition is violated **do**

3: Pick an $\alpha \in \Sigma_s(\mathbf{z}^k - \tau \boldsymbol{g}^k)$.

4: Update $\mathbf{d}^k$ by (20).

5: Find the smallest non-negative integer r such that

$$2f(\mathbf{z}^k(c^r)) \leq 2f(\mathbf{z}^k) + c^r \langle \boldsymbol{g}^k, \mathbf{d}^k \rangle. \tag{22}$$

6: Set $\sigma_k := c^r, \mathbf{z}^{k+1} := \mathbf{z}^k(\sigma_k)$ and $k := k + 1$.

7: **end while**

8: **return** $\mathbf{z}^k$

So, the whole complexity is $\mathcal{O}(s^3 + s^2 n)$*, which means the computation is quite fast if* $\max\{s, n\} \ll p$.

iii) For a halting condition, we will calculate the quantity $\|\theta_\alpha^k\|$*. If* $\|\theta_\alpha^k\| = 0$*, then* $\mathbf{z}^k$ *satisfies* (17)*. This means* $\mathbf{z}^k$ *is a* τ*-stationary point by Theorem 2.4. Therefore, it makes sense to terminate* `NSLR` *if the quantity* $\|\theta_\alpha^k\|$ *is sufficiently small.*

3.2. Global and quadratic convergence

To derive the convergence properties, we denote some notation hereafter. Let β be the index set related the previous iteration $\mathbf{z}^{k-1}$ selected by

$$\beta \in \Sigma_s(\mathbf{z}^{k-1} - \tau \boldsymbol{g}^{k-1}).$$

Based on $\mathbf{z}^k = \mathbf{z}^{k-1}(\sigma_{k-1}), \mathbf{z}^{k+1} = \mathbf{z}^k(\sigma_k)$ in Algorithm 1 and the definition of $\mathbf{z}^k(\sigma)$ in (21), we must have

$$\mathrm{supp}(\mathbf{z}^k) \subseteq \beta, \quad \mathrm{supp}(\mathbf{z}^{k+1}) \subseteq \alpha. \tag{23}$$

Let $\gamma := \beta \setminus \alpha$. Then one can observe that

$$-\mathbf{d}_{\overline{\alpha}}^k = \mathbf{z}_{\overline{\alpha}}^k = \begin{bmatrix} \mathbf{z}_{\beta \cap \overline{\alpha}}^k \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{\beta \setminus \alpha}^k \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{\gamma}^k \\ 0 \end{bmatrix}. \tag{24}$$

This gives rise to the following properties

$$\|\mathbf{d}_{\overline{\alpha}}^k\| = \|\mathbf{d}_{\gamma}^k\| = \|\mathbf{z}_{\gamma}^k\| = \|\mathbf{z}_{\overline{\alpha}}^k\|, \tag{25}$$

$$\langle \boldsymbol{g}_{\overline{\alpha}}^k, \mathbf{d}_{\overline{\alpha}}^k \rangle = \langle \boldsymbol{g}_{\gamma}^k, \mathbf{d}_{\gamma}^k \rangle. \tag{26}$$

Based on these, we have the following results.

Lemma 3.2. *Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 1. We have the following properties:*

$$\begin{aligned} 2\langle \mathbf{d}_{\alpha}^k, \boldsymbol{g}_{\alpha}^k \rangle &\leq -2\lambda \|\mathbf{d}_{\alpha}^k\|^2 + L\|\mathbf{d}_{\overline{\alpha}}^k\|^2, & (27)\\ 2\langle \mathbf{d}_{\overline{\alpha}}^k, \boldsymbol{g}_{\overline{\alpha}}^k \rangle &\leq \tau L^2 \|\mathbf{d}_{\alpha}^k\|^2 + (\tau L^2 - 1/\tau)\|\mathbf{d}_{\overline{\alpha}}^k\|^2. & (28) \end{aligned}$$

Proof It follows from (20) that

$$\boldsymbol{g}_{\alpha}^k = -H_{\alpha\alpha}^k \mathbf{d}_{\alpha}^k + H_{\alpha\overline{\alpha}}^k \mathbf{z}_{\overline{\alpha}}^k \overset{(20)}{=} -H_{\alpha\cdot}^k \mathbf{d}^k. \tag{29}$$

Direct calculation yields the following chain of equations,

$$\begin{aligned} & \langle \mathbf{d}^k, H^k \mathbf{d}^k \rangle - \langle \mathbf{d}_{\overline{\alpha}}^k, H_{\overline{\alpha}\overline{\alpha}}^k \mathbf{d}_{\overline{\alpha}}^k \rangle \\ = \; & \langle \mathbf{d}_{\alpha}^k, H_{\alpha\alpha}^k \mathbf{d}_{\alpha}^k \rangle + 2\langle H_{\alpha\overline{\alpha}}^k \mathbf{d}_{\overline{\alpha}}^k, \mathbf{d}_{\alpha}^k \rangle \\ = \; & 2\langle H_{\alpha\alpha}^k \mathbf{d}_{\alpha}^k + H_{\alpha\overline{\alpha}}^k \mathbf{d}_{\overline{\alpha}}^k, \mathbf{d}_{\alpha}^k \rangle - \langle \mathbf{d}_{\alpha}^k, H_{\alpha\alpha}^k \mathbf{d}_{\alpha}^k \rangle \\ \overset{(24)}{=} \; & 2\langle H_{\alpha\alpha}^k \mathbf{d}_{\alpha}^k - H_{\alpha\overline{\alpha}}^k \mathbf{z}_{\overline{\alpha}}^k, \mathbf{d}_{\alpha}^k \rangle - \langle \mathbf{d}_{\alpha}^k, H_{\alpha\alpha}^k \mathbf{d}_{\alpha}^k \rangle \\ \overset{(29)}{=} \; & -2\langle \mathbf{d}_{\alpha}^k, \boldsymbol{g}_{\alpha}^k \rangle - \langle H_{\alpha\alpha}^k \mathbf{d}_{\alpha}^k, \mathbf{d}_{\alpha}^k \rangle, \end{aligned}$$

which leads to the truth

$$\begin{aligned} 2\langle \mathbf{d}_{\alpha}^k, \boldsymbol{g}_{\alpha}^k \rangle &= \langle \mathbf{d}_{\overline{\alpha}}^k, H_{\overline{\alpha}\overline{\alpha}}^k \mathbf{d}_{\overline{\alpha}}^k \rangle - \langle H_{\alpha\alpha}^k \mathbf{d}_{\alpha}^k, \mathbf{d}_{\alpha}^k \rangle - \langle \mathbf{d}^k, H^k \mathbf{d}^k \rangle \\ &\leq L\|\mathbf{d}_{\overline{\alpha}}^k\|^2 - \lambda\|\mathbf{d}_{\alpha}^k\|^2 - \lambda\|\mathbf{d}_{\alpha\cup\overline{\alpha}}^k\|^2 \\ &\leq L\|\mathbf{d}_{\overline{\alpha}}^k\|^2 - 2\lambda\|\mathbf{d}_{\alpha}^k\|^2, \end{aligned}$$

where the inequality is from the fact that f being strongly convex with the constant λ and strongly smooth with the constant L so as to satisfy

$$\lambda \leq \lambda_{\min}(H^k) \leq \lambda_{\max}(H^k) \leq L. \tag{30}$$

For the part $\overline{\alpha}$, since $\alpha \in \Sigma_s(\mathbf{z}^k - \tau \boldsymbol{g}^k)$, the definition of $\alpha \in \Sigma_s$ in (15) implies

$$\forall\ i \in \alpha,\ \ |z_i^k - \tau g_i^k| \geq |z_j^k - \tau g_j^k|,\ \ \forall\ j \in \overline{\alpha}.$$

Now for $i \in \alpha \setminus \beta =: \eta$, we have $z_i^k = 0$ due to (23). Then the above condition and $\gamma \subseteq \overline{\alpha}$ result in

$$\forall\ i \in \eta,\ \ |\tau g_i^k| \geq |z_j^k - \tau g_j^k|,\ \ \forall\ j \in \gamma, \tag{31}$$

which together with $\mathbf{z}_{\overline{\alpha}}^k = -\mathbf{d}_{\overline{\alpha}}^k$ from (20) and $|\eta| = |\alpha| - |\alpha \cap \beta| = s - |\alpha \cap \beta| = |\beta| - |\alpha \cap \beta| = |\gamma|$ leads to

$$\|\tau \boldsymbol{g}_\eta^k\|^2 \geq \|\mathbf{z}_\gamma^k - \tau \boldsymbol{g}_\gamma^k\|^2 \overset{(24)}{=} \|\mathbf{d}_\gamma^k + \tau \boldsymbol{g}_\gamma^k\|^2.$$

The above condition allows us to derive that

$$\begin{aligned} 2\langle \mathbf{d}_{\overline{\alpha}}^k, \boldsymbol{g}_{\overline{\alpha}}^k \rangle &\overset{(26)}{=} 2\langle \mathbf{d}_\gamma^k, \boldsymbol{g}_\gamma^k \rangle \\ &\leq \tau \|\boldsymbol{g}_\eta^k\|^2 - \tau \|\boldsymbol{g}_\gamma^k\|^2 - (1/\tau)\|\mathbf{d}_\gamma^k\|^2 \\ &\leq \tau \|\boldsymbol{g}_\alpha^k\|^2 - (1/\tau)\|\mathbf{d}_\gamma^k\|^2 \\ &\leq \tau L^2 \|\mathbf{d}^k\|^2 - (1/\tau)\|\mathbf{d}_{\overline{\alpha}}^k\|^2 \\ &= \tau L^2 \|\mathbf{d}_\alpha^k\|^2 + (\tau L^2 - 1/\tau)\|\mathbf{d}_{\overline{\alpha}}^k\|^2, \end{aligned}$$

where the last inequality is owing to (25) and

$$\begin{aligned} \|\boldsymbol{g}_\alpha^k\|^2 \overset{(29)}{=} \|H_{\alpha:}^k \mathbf{d}^k\|^2 &\leq \|H_{\alpha:}^k \mathbf{d}^k\|^2 + \|H_{\overline{\alpha}:}^k \mathbf{d}^k\|^2 \\ = \|H^k \mathbf{d}^k\|^2 &\overset{(30)}{\leq} L^2 \|\mathbf{d}^k\|^2, \end{aligned} \tag{32}$$

showing the desired results. □

The first result below shows that the Newton direction $\mathbf{d}^k$ is a descent direction and the Amijio-type step size σ_k always exists and is away from zero.

Lemma 3.3 (Descent property). *Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 1 and*

$$0 < \tau < \overline{\tau} := \min\left\{ \frac{2\lambda}{L^2},\ \frac{c\lambda^2}{4L^3} \right\}. \tag{33}$$

Denote $C := \min\left\{1/\tau - L - \tau L^2,\ 2\lambda - \tau L^2\right\} > 0$. *Then we have*

$$2\langle \boldsymbol{g}^k, \mathbf{d}^k\rangle \leq -C\|\mathbf{d}^k\|^2. \tag{34}$$

Moreover, for any $c\overline{\sigma} \leq \sigma \leq \overline{\sigma} := \lambda/(2L)$, *it holds*

$$2f(\mathbf{z}^k(\sigma)) \leq 2f(\mathbf{z}^k) + \sigma\langle \boldsymbol{g}^k, \mathbf{d}^k\rangle. \tag{35}$$

This indicates $\inf_{k\geq 0}\sigma_k \geq c\overline{\sigma} > 0$.

Proof Denote two parameters

$$C_1 := \frac{1}{\tau} - L - \tau L^2, \quad C_2 := 2\lambda - \tau L^2, \quad C = \min\{C_1, C_2\}.$$

It is easy to see that $C_2 > 0$ by (33) and $C_1 > 0$ due to

$$0 < \tau < \overline{\tau} \leq c\lambda^2/(4L^3) = (c\lambda^2/L^2)/(4L) \leq 1/(4L)$$

and $0 < c, \lambda/L < 1$. Overall, $C > 0$. Then by (27) and (28), we have

$$\begin{aligned} 2\langle \mathbf{d}^k, \boldsymbol{g}^k\rangle &= 2\langle \mathbf{d}_\alpha^k, \boldsymbol{g}_\alpha^k\rangle + 2\langle \mathbf{d}_{\overline{\alpha}}^k, \boldsymbol{g}_{\overline{\alpha}}^k\rangle \\ &\leq -C_1\|\mathbf{d}_\alpha^k\|^2 - C_2\|\mathbf{d}_{\overline{\alpha}}^k\|^2 \\ &\leq -C\|\mathbf{d}_\alpha^k\|^2 - C\|\mathbf{d}_{\overline{\alpha}}^k\|^2 \\ &= -C\|\mathbf{d}^k\|^2. \end{aligned}$$

Note from (21) that

$$\mathbf{z}^k(\sigma) - \mathbf{z}^k = \begin{bmatrix} \sigma\mathbf{d}_\alpha^k \\ \mathbf{d}_{\overline{\alpha}}^k \end{bmatrix}. \tag{36}$$

The strong smoothness of f with the constant L yields

$$\begin{aligned} & 2f(\mathbf{z}^k(\sigma)) - 2f(\mathbf{z}^k) - \sigma\langle \boldsymbol{g}^k, \mathbf{d}^k\rangle \\ \leq\ & 2\langle \boldsymbol{g}^k, \mathbf{z}^k(\sigma) - \mathbf{z}^k\rangle + L\|\mathbf{z}^k(\sigma) - \mathbf{z}^k\|^2 - \sigma\langle \boldsymbol{g}^k, \mathbf{d}^k\rangle \\ =\ & \sigma\langle \boldsymbol{g}_\alpha^k, \mathbf{d}_\alpha^k\rangle + (2-\sigma)\langle \boldsymbol{g}_{\overline{\alpha}}^k, \mathbf{d}_{\overline{\alpha}}^k\rangle + L\sigma^2\|\mathbf{d}_\alpha^k\|^2 + L\|\mathbf{d}_{\overline{\alpha}}^k\|^2 \\ =:\ & F. \end{aligned}$$

We next show $F \leq 0$. It follows from (27) and (28) that

$$\begin{aligned} 2F &\leq \left(-2\sigma\lambda + (2-\sigma)\tau L^2\right) \|\mathbf{d}_\alpha^k\|^2 \\ &+ \left[\sigma L + (2-\sigma)(\tau L^2 - 1/\tau)\right] \|\mathbf{d}_{\overline{\alpha}}^k\|^2 + 2\sigma^2 L \|\mathbf{d}_\alpha^k\|^2 + 2L\|\mathbf{d}_{\overline{\alpha}}^k\|^2 \\ &= \left(-2\sigma\lambda + (2-\sigma)\tau L^2 + 2\sigma^2 L\right) \|\mathbf{d}_\alpha^k\|^2 \\ &+ \left[\sigma L + (2-\sigma)(\tau L^2 - 1/\tau) + 2L\right] \|\mathbf{d}_{\overline{\alpha}}^k\|^2. \end{aligned}$$

One can check that, by $0 < \tau < \overline{\tau} \leq c\lambda^2/(4L^3)$, it follows

$$\begin{aligned} & -2\sigma\lambda + (2-\sigma)\tau L^2 + 2\sigma^2 L \\ \leq\ & -2\sigma\lambda + (2-\sigma)c\lambda^2/(4L) + 2\sigma^2 L \\ =\ & 2L\sigma^2 - \left(2\lambda + c\lambda^2/(4L)\right)\sigma + c\lambda^2/(2L) \leq 0, \end{aligned}$$

where the last inequality is true if $c\lambda/(2L) \leq \sigma \leq \lambda/(2L)$. For the $\overline{\alpha}$ part,

$$\begin{aligned} & \sigma L + (2-\sigma)\left(\tau L^2 - 1/\tau\right) + 2L \\ \leq\ & \sigma L + (2-\sigma)L + (\sigma - 2)/\tau + 2L \\ \leq\ & 4L - 1/\tau \leq 0, \end{aligned}$$

where the above three inequalities used the facts that

(a) $\tau L^2 \leq c\lambda^2/(4L) \leq \lambda(c/4)(\lambda/L) \leq L$;

(b) $\sigma \leq 1$ due to $\sigma_k = c^r$ and $c \in (0,1)$ in Algorithm 1;

(c) $\tau \leq c\lambda^2/(4L^3) \leq 1/(4L)$.

Therefore, $F \leq 0$, displaying (35). Then the Armijo step-size rule indicates that $\inf_{k\geq 0} \sigma_k \geq c\lambda/(2L) > 0$. □

Now we are ready to display the main results of the method including the global convergence to a τ-stationary point, the support set identification, and the quadratic convergence rate.

Theorem 3.4 (Global and quadratic convergence). *Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 1 and $0 < \tau < \overline{\tau}$. The following results hold.*

i) The whole sequence converges to a τ-stationary point denoted by $\mathbf{z}^*$, which is at least a local minimizer of the problem (6).

ii) For sufficiently large k, the support sets of the sequence are identified by

$$\mathrm{supp}(\mathbf{z}^*) = \begin{cases} \mathrm{supp}(\mathbf{z}^k) = \alpha, & \|\mathbf{z}^*\|_0 = s, \\ \mathrm{supp}(\mathbf{z}^k) \cap \alpha, & \|\mathbf{z}^*\|_0 < s. \end{cases} \tag{37}$$

iii) The sequence converges to $\mathbf{z}^*$ quadratically, namely,

$$\|\mathbf{z}^{k+1} - \mathbf{z}^*\| \leq M/(2\lambda)\|\mathbf{z}^k - \mathbf{z}^*\|^2. \tag{38}$$

Proof i) Lemma 3.3 shows that $\sigma_k \geq c\overline{\sigma}$ and results in

$$\begin{aligned} 2f(\mathbf{z}^{k+1}) &= 2f(\mathbf{z}^k(\sigma_k)) \\ &\overset{(35)}{\leq} 2f(\mathbf{z}^k) + \sigma_k\langle \boldsymbol{g}^k, \mathbf{d}^k\rangle \\ &\overset{(34)}{\leq} 2f(\mathbf{z}^k) - \sigma_k C\|\mathbf{d}^k\|^2 \\ &\leq 2f(\mathbf{z}^k) - \overline{\sigma}cC\|\mathbf{d}^k\|^2. \end{aligned}$$

Then it follows from the above inequality that

$$\begin{aligned} \sum_{k=0}^{\infty} \overline{\sigma}cC\|\mathbf{d}^k\|^2 &\leq \sum_{k=0}^{\infty}\left[2f(\mathbf{z}^k) - 2f(\mathbf{z}^{k+1})\right] \\ &= 2f(\mathbf{z}^0) - \lim_{k\to+\infty} 2f(\mathbf{z}^k) \\ &\leq 2f(\mathbf{z}^0), \end{aligned}$$

where the last inequality is due to f is positive. Hence $\|\mathbf{d}^k\| \to 0$, which suffices to $\|\mathbf{z}^{k+1} - \mathbf{z}^k\| \to 0$ since

$$\|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2 \overset{(36)}{=} \sigma^2\|\mathbf{d}_\alpha^k\|^2 + \|\mathbf{d}_{\overline{\alpha}}^k\|^2 \to 0.$$

This also indicates $\|\mathbf{z}_{\overline{\alpha}}^k\|^2 = \|\mathbf{d}_{\overline{\alpha}}^k\|^2 \to 0$ and by (32) suffices to

$$\|\boldsymbol{g}_\alpha^k\| \leq L\|\mathbf{d}^k\| \to 0. \tag{39}$$

Let $\{\mathbf{z}^{k_\ell}\}$ be the convergent subsequence of $\{\mathbf{z}^k\}$ that converges to $\mathbf{z}^*$ and

$$\alpha_\ell \in \Sigma_s(\mathbf{z}^{k_\ell} - \tau\boldsymbol{g}^{k_\ell}), \quad \ell \geq 1.$$

Since there are only finitely many choices for $\alpha_\ell \subseteq [p]$, (re-subsequencing if necessary) we may without loss of any generality assume that the sequence of the index sets $\{\alpha_\ell\}$ shares a same index set, denoted as α_∞. That is

$$\alpha_\ell = \alpha_{\ell+1} = \cdots = \alpha_\infty. \tag{40}$$

Now by letting $\boldsymbol{g}^* := \nabla f(\mathbf{z}^*)$, one can show that

$$\|\boldsymbol{g}^*_{\alpha_\infty}\| = \lim_{k_\ell \to \infty} \|\boldsymbol{g}^{k_\ell}_{\alpha_\infty}\| = \lim_{k_\ell \to \infty} \|\boldsymbol{g}^{k_\ell}_{\alpha_\ell}\| \overset{(39)}{=} 0. \tag{41}$$

In addition, the definition of Σ_s in (15) implies

$$\forall\ i \in \alpha_\ell = \alpha_\infty,\ |z_i^{k_\ell} - \tau g_i^{k_\ell}| \geq |z_j^{k_\ell} - \tau g_j^{k_\ell}|,\ \forall\ j \in \overline{\alpha}_\ell = \overline{\alpha}_\infty.$$

Taking the limit of both sides of the above inequality yields

$$\forall\ i \in \alpha_\infty,\quad |z_i^*| \geq |\tau g_j^*|,\quad \forall\ j \in \overline{\alpha}_\infty. \tag{42}$$

Here, we used the facts that (41) and $\|\mathbf{z}^{k_\ell}_{\overline{\alpha}_\infty}\| = \|\mathbf{z}^{k_\ell}_{\overline{\alpha}_\ell}\| \to 0$. Since $\mathbf{z}^{k_\ell} \to \mathbf{z}^*$, we have $\mathrm{supp}(\mathbf{z}^*) \subseteq \alpha_\infty$.

- If $\|\mathbf{z}^*\|_0 = s$, then $\mathrm{supp}(\mathbf{z}^*) = \alpha_\infty$, which by (42) derives $|\tau g_j^*| \leq [\mathbf{z}^*]_s^\downarrow, j \notin \mathrm{supp}(\mathbf{z}^*)$. This together with (41) results in condition (11).

- If $\|\mathbf{z}^*\|_0 < s$, then $\mathrm{supp}(\mathbf{z}^*) \subset \alpha_\infty$, which by (42) delivers $|\tau g_j^*| \leq [\mathbf{z}^*]_s^\downarrow = 0, j \notin \overline{\alpha}_\infty$. This together with (41) yields $\boldsymbol{g}^* = 0$, which also satisfies (11).

Overall, both cases show $\mathbf{z}^*$ is a τ stationary point of (6). From Theorem 2.3, a τ stationary point of (6) is a unique local minimizer, namely, $\mathbf{z}^*$ is isolated. By [41, Lemma 4.10], the whole sequence converges to $\mathbf{z}^*$ because $\mathbf{z}^*$ is isolated and $\|\mathbf{z}^{k+1} - \mathbf{z}^k\| \to 0$.

ii) We proved that the whole sequence converges to $\mathbf{z}^*$. Denote $\alpha_* := \mathrm{supp}(\mathbf{z}^*)$. If $\mathbf{z}^* = 0$, then the conclusion holds clearly due to $\alpha_* = \emptyset$. Consider $\mathbf{z}^* \neq 0$. For sufficiently large k, we must have

$$\|\mathbf{z}^k - \mathbf{z}^*\| < \min_{i \in \alpha_*} |z_i^*| =: \delta.$$

If $\alpha_* \nsubseteq \mathrm{supp}(\mathbf{z}^k)$, then there is an $i_0 \in \alpha_* \setminus \mathrm{supp}(\mathbf{z}^k)$ satisfying

$$\delta > \|\mathbf{z}^k - \mathbf{z}^*\| \geq |z_{i_0}^k - z_{i_0}^*| = |z_{i_0}^*| \geq \delta,$$

which is a contradiction. Therefore, $\alpha_* \subseteq \mathrm{supp}(\mathbf{z}^k)$. By (23), we have $\mathrm{supp}(\mathbf{z}^k) \subseteq \beta$, where $|\beta| = s$ by (15). Therefore, if $\|\mathbf{z}^*\|_0 = s$ then $\beta \equiv \mathrm{supp}(\mathbf{z}^k) \equiv \alpha_*$ for any sufficiently large k. Particularly, $\alpha_* \equiv \mathrm{supp}(\mathbf{z}^{k+1}) \equiv \alpha$. If $\|\mathbf{z}^*\|_0 < s$ then $\alpha_* \subseteq \mathrm{supp}(\mathbf{z}^k)$ and $\alpha_* \subseteq \mathrm{supp}(\mathbf{z}^{k+1}) \subseteq \alpha$ from (23). So (37) is true.

iii) For sufficiently large k, it follows from $\alpha_* \subseteq \alpha$ by ii) that $\mathbf{z}_{\overline{\alpha}}^* = 0$. Since $\mathbf{z}^*$ is a τ-stationary point, (11) indicates $\boldsymbol{g}^* = 0$ if $\|\mathbf{z}^*\|_0 < s$ and $\boldsymbol{g}_{\alpha_*}^* = 0$ if $\|\mathbf{z}^*\|_0 = s$. While for the latter case, there is $\alpha_* = \alpha$ by ii). Overall, we have

$$\mathbf{z}_{\overline{\alpha}}^* = 0, \quad \boldsymbol{g}_{\alpha}^* = 0. \tag{43}$$

For any $0 \leq t \leq 1$, denote $\mathbf{z}(t) := \mathbf{z}^* + t(\mathbf{z}^k - \mathbf{z}^*)$ and $H^k(t) := \nabla^2 f(\mathbf{z}^k(t))$. Then (9) derives

$$\|H^k - H^k(t)\| \leq M\|\mathbf{z}^k - \mathbf{z}(t)\| = (1-t)M\|\mathbf{z}^k - \mathbf{z}^*\|. \tag{44}$$

Moreover, by Taylor expansion, one has

$$\boldsymbol{g}^k - \boldsymbol{g}^* = \int_0^1 H^k(t)(\mathbf{z}^k - \mathbf{z}^*)dt. \tag{45}$$

Thanks to (23) and (43), we have $\mathbf{z}_{\overline{\alpha}}^{k+1} = \mathbf{z}_{\overline{\alpha}}^* = 0$ and the following chain of inequalities

$$\begin{aligned}
\|\mathbf{z}^{k+1} - \mathbf{z}^*\|^2 &= \|\mathbf{z}_{\alpha}^{k+1} - \mathbf{z}_{\alpha}^*\|^2 \overset{(36)}{=} \|\mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^* + \sigma_k \mathbf{d}_{\alpha}^k\|^2 \\
&= \|(1-\sigma_k)(\mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^*) + \sigma_k(\mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^* + \mathbf{d}_{\alpha}^k)\|^2 \\
&\leq (1-\sigma_k)\|\mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^*\|^2 + \sigma_k\|\mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^* + \mathbf{d}_{\alpha}^k\|^2 \qquad (46) \\
&\leq (1 - c\overline{\sigma})\|\mathbf{z}^k - \mathbf{z}^*\|^2 + \overline{\sigma}\|\mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^* + \mathbf{d}_{\alpha}^k\|^2, \qquad (47)
\end{aligned}$$

where (46) is due to $\|\cdot\|^2$ is a convex function and the last inequality is from Lemma 3.3 that $c\overline{\sigma} \leq \sigma_k \leq \overline{\sigma}$. For the second term of (47), we have

$$\begin{aligned}
\lambda\|\mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^* + \mathbf{d}_{\alpha}^k\| &\overset{(20)}{=} \lambda\|(H_{\alpha\alpha}^k)^{-1}(H_{\alpha\overline{\alpha}}^k \mathbf{z}_{\overline{\alpha}}^k - \boldsymbol{g}_{\alpha}^k) + \mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^*\| \\
&\leq \lambda\|(H_{\alpha\alpha}^k)^{-1}\|\|H_{\alpha\overline{\alpha}}^k \mathbf{z}_{\overline{\alpha}}^k - \boldsymbol{g}_{\alpha}^k + H_{\alpha\alpha}^k(\mathbf{z}_{\alpha}^k - \mathbf{z}_{\alpha}^*)\|
\end{aligned}$$

$$
\begin{aligned}
&\overset{(30)}{\leq} && \|H^k_{\alpha:}\mathbf{z}^k - \boldsymbol{g}^k_\alpha - H^k_{\alpha\alpha}\mathbf{z}^*_\alpha\| \\
&\overset{(43)}{=} && \|H^k_{\alpha:}\mathbf{z}^k - \boldsymbol{g}^k_\alpha - H^k_{\alpha:}\mathbf{z}^* + \boldsymbol{g}^*_\alpha\| \\
&\overset{(45)}{=} && \|H^k_{\alpha:}(\mathbf{z}^k - \mathbf{z}^*) - \int_0^1 H^k_{\alpha:}(t)(\mathbf{z}^k - \mathbf{z})dt\| \\
&= && \|\int_0^1 (H^k_{\alpha:} - H^k_{\alpha:}(t))(\mathbf{z}^k - \mathbf{z}^*)dt\| \\
&\leq && \int_0^1 \|(H^k_{\alpha:} - H^k_{\alpha:}(t))(\mathbf{z}^k - \mathbf{z}^*)\|dt \\
&\leq && \int_0^1 \|H^k - H^k(t)\|\|\mathbf{z}^k - \mathbf{z}^*\|dt \\
&\overset{(44)}{\leq} && M\|\mathbf{z}^k - \mathbf{z}^*\|^2 \int_0^1 (1-t)dt \\
&= && (M/2)\|\mathbf{z}^k - \mathbf{z}^*\|^2, \qquad (48)
\end{aligned}
$$

where the forth inequality used a fact that

$$\|A_{\alpha:}\mathbf{z}\|^2 \leq \|A_{\alpha:}\mathbf{z}\|^2 + \|A_{\overline{\alpha}:}\mathbf{z}\|^2 = \|A\mathbf{z}\|^2 \leq \|A\|^2\|\mathbf{z}\|^2.$$

It follows from $\mathbf{d}^k_{\overline{\alpha}} = -\mathbf{z}^k_{\overline{\alpha}}$ and (43) that

$$\|\mathbf{z}^k + \mathbf{d}^k - \mathbf{z}^*\| = \|\mathbf{z}^k_\alpha + \mathbf{d}^k_\alpha - \mathbf{z}^*_\alpha\|,$$

leading to the following fact

$$\frac{\|\mathbf{z}^k + \mathbf{d}^k - \mathbf{z}^*\|}{\|\mathbf{z}^k - \mathbf{z}^*\|} = \frac{\|\mathbf{z}^k_\alpha + \mathbf{d}^k_\alpha - \mathbf{z}^*_\alpha\|}{\|\mathbf{z}^k - \mathbf{z}^*\|} \overset{(48)}{\leq} \frac{M\|\mathbf{z}^k - \mathbf{z}^*\|^2}{2\lambda\|\mathbf{z}^k - \mathbf{z}^*\|} \to 0. \qquad (49)$$

Now we have three facts: (49), $\mathbf{z}^k \to \mathbf{z}^*$ from i), and $\langle \boldsymbol{g}^k, \mathbf{d}^k\rangle \leq -C\|\mathbf{d}^k\|^2$ from Lemma 3.3. They and [42, Theorem 3.3] allow us to claim that eventually the step size σ_k determined by the Armijo rule is 1, namely, $\sigma_k = 1$. Then it follows from (46) that

$$
\begin{aligned}
\|\mathbf{z}^{k+1} - \mathbf{z}^*\|^2 &\overset{(46)}{\leq} (1-\sigma_k)\|\mathbf{z}^k_\alpha - \mathbf{z}^*_\alpha\|^2 + \sigma_k\|\mathbf{z}^k_\alpha - \mathbf{z}^*_\alpha + \mathbf{d}^k_\alpha\|^2 \\
&= \|\mathbf{z}^k_\alpha - \mathbf{z}^*_\alpha + \mathbf{d}^k_\alpha\|^2 \overset{(48)}{\leq} M^2/(2\lambda)^2\|\mathbf{z}^k - \mathbf{z}^*\|^4,
\end{aligned}
$$

delivering the quadratic convergence property of the sequence. □

4. Numerical Experiments

In this part, we will conduct extensive numerical experiments of `NSLR`[1] by using MATLAB (R2017b) on a desktop of 8GB of memory and Inter Core i5 2.7Ghz CPU, against seven leading solvers on both synthetic and real datasets.

4.1. Test examples

We first adopt two types of randomly generated data: the one with the identically independently generated features $[\mathbf{x}_1 \ \cdots \ \mathbf{x}_n] \in \mathbb{R}^{p\times n}$ and the one with independent features with each of $\mathbf{x}_i$ being generated by an autoregressive process [43]. Then eight real datasets are taken into consideration to test the selected methods.

Example 4.1 (Independent Data [36, 38]). *To generate data labels* $\mathbf{y} \in \{0,1\}^n$*, we first randomly divide* $[n]$ *into two parts and set* $y_i = 0$ *for one part and* $y_i = 1$ *for the other. Then the feature data is produced by*

$$\mathbf{x}_i = y_i v_i \boldsymbol{1} + \mathbf{w}_i, \qquad i \in [n]$$

with $\mathbb{R} \ni v_i \sim \mathcal{N}(0,1)$*,* $\mathbb{R}^p \ni \mathbf{w}_i \sim \mathcal{N}(0, I_p)$*, where* $\mathcal{N}(0,I)$ *is the normal distribution with mean zero and variance identity. Here,* $\boldsymbol{1}$ *is the vector with all entries being ones. Since the sparse parameter* $\mathbf{z}^* \in \mathbb{R}^p$ *is unknown, different* $s(< n)$ *will be tested to pursue a sparse solution.*

Example 4.2 (Correlated Data [44, 2]). *The sparse parameter* $\mathbf{z}^* \in \mathbb{R}^p$ *has* s *nonzero entries drawn independently from the standard Gaussian distribution. Each data sample* $\mathbf{x}_i = [x_{i1} \ \cdots \ x_{ip}]^\top, i \in [n]$ *is an independent instance of the random vector generated by an autoregressive process [43] determined by*

$$x_{i(j+1)} = \rho x_{ij} + \sqrt{1-\rho^2} v_{ij}, \quad j \in [p-1]$$

with $x_{i1} \sim \mathcal{N}(0,1)$*,* $v_{ij} \sim \mathcal{N}(0,1)$ *and* $\rho \in [0,1]$ *being the correlation parameter. The data labels* $y_i \in \{0,1\}$ *are then drawn randomly according to the Bernoulli distribution with the conditional probability* (1)*.*

[1] Available at https://github.com/ShenglongZhou/NSLR

Example 4.3 (Real data). *Eight real data sets are taken into consideration. They are summarized in Table 1, where* `arcene` *and* `newsgroup` *are taken from UCI repository*[2] *and* `glmnet` *package*[3]*, and the rest of them are LIBSVM data*[4]*. Moreover, all datasets are feature-wisely scaled to* $[-1, 1]$*. All* -1*s in the label classes* $\mathbf{y}$ *are replaced by 0. The sizes of training data and testing data are denoted by* m_1 *and* m_2 *respectively.*

Table 1: Details of eight real datasets.

			Training	Testing
Data name	n	p	m_1	m_2
`arcene`	100	10,000	100	0
`colon-cancer`	62	2,000	62	0
`news20.binary`	19,996	1,355,191	19,996	0
`newsgroup`	11,314	777,811	11,314	0
`duke breast-cancer`	42	7,129	38	4
`leukemia`	72	7,129	38	34
`gisette`	7,000	5,000	6,000	1,000
`rcv1.binary`	40,242	47,236	20,242	20,000

4.2. Implementation

In the model (6), we set $\lambda = 10^{-5}/n$. As mentioned in Remark 3.1, we terminate the method if $\|\theta^k\| < 10^{-10}\sqrt{p}$ or $k > 2000$. For the starting point and parameters of `NSLR`, we set $\mathbf{z}^0 = 0$ and $c = 0.5$. For the parameter τ, Theorem 3.4 indicates $0 < \tau < \overline{\tau}$. While this is a sufficient condition. So it is unnecessary to choose a τ to satisfy the condition strictly, not to mention, $\overline{\tau}$ might be too small if we set a tiny λ.

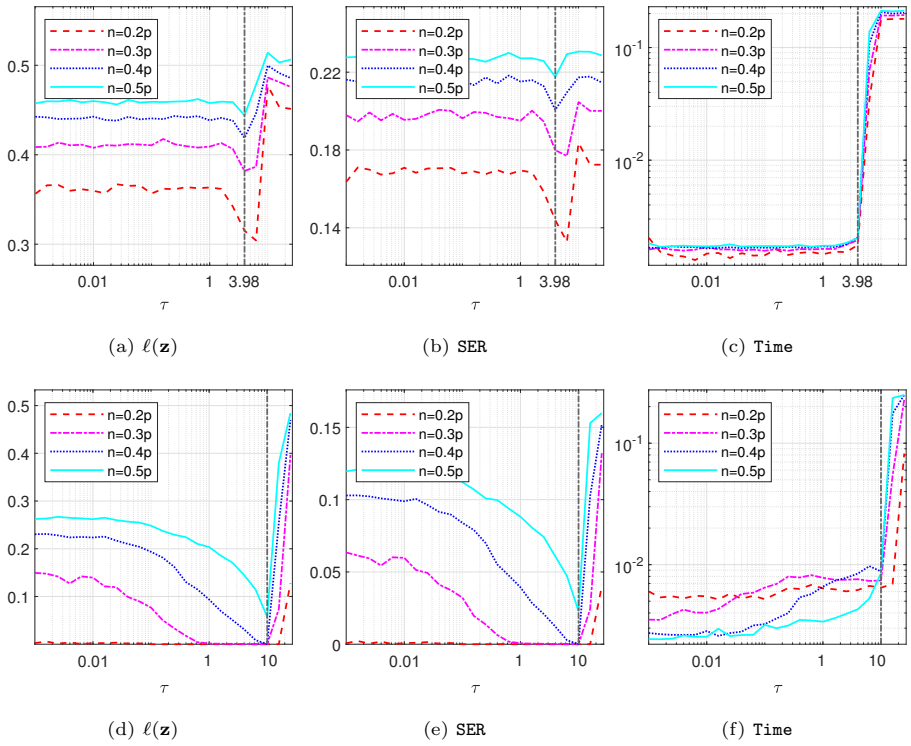
Alternative is to pick a proper τ by tuning it from a wide range of values. For instance, we tested `NSLR` on a range of selections of $\tau = 10^{\varrho}$ with $\varrho \in \{-3, -2.8, \cdots, 1, 1.2, 1.4\}$ for solving Example 4.1. As reported in Figure 1, for $s = 5$, `NSLR` generated the best results when τ was around 3.98, while for

[2] http://archive.ics.uci.edu/ml/index.php

[3] https://web.stanford.edu/~hastie/glmnet_matlab/

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

$s = 25$, it delivered the best results when τ was around 10. Therefore, for different scenarios, the best option τ may be varied, which indicates the manual selection of τ is necessary to achieve a better performance. This apparently would incur expensive computational costs.



(a) $\ell(\mathbf{z})$ (b) SER (c) Time

(d) $\ell(\mathbf{z})$ (e) SER (f) Time

Figure 1: Effect to τ of NSLR for Example 4.1 with $p = 500$, $s = 5$ (the above three sub-figures) and $s = 25$ (the below three sub-figures).

However, the empirical numerical experience have demonstrated that adaptively updating parameters during the process is a practicable strategy. Hence, we start τ with a fixed scalar $\tau_0 = 15$ and update $\tau_{k+1} = 0.75\tau_k$ if k is a multiple of 10 and $\|\theta^k\| > 1/k$, and $\tau_{k+1} = \tau_k$ otherwise. In the sequel, we adopt this strategy for NSLR and the numerical comparisons with other leading solvers will show the superior performance of our method under such a strategy.

4.3. Benchmark methods

Since there is an impressive body of methods that have been developed to address the sparse logistic regression, we only focus on those programmed by Matlab. Solvers with codes being online unavailable or being written by other languages, such as R and C, are not selected for comparisons. We thus choose 7 solvers mentioned in Subsection 1.2, which should be enough to make comprehensive comparisons. We summarize them into the following table.

Table 2: Benchmark Methods

Models	(4) with convex ϕ_ν	(4) with non-convex ϕ_ν	(6)
first-order	`SLEP`	`APG`, `GIST`	`GraSP`
second-order	`IRLS-LARS`	`--`	`NTGP`, `GPGN`

For `SLEP`, we use it to solve (4) with $\phi_\nu(\mathbf{z}) = \nu_1\|\mathbf{z}\|_2^2 + \nu_2\|\mathbf{z}\|_1$, whilst `IRLS-LARS` aims to solve the case of $\nu = 0$. `APG` and `GIST` are taken to solve the capped ℓ_1 logistic regression with $\phi_\nu(\mathbf{z}) = \nu_3 \min(|\mathbf{z}_i|, \nu_4)$. We only use non-monotonous version of `APG` since its numerical performance was better than that of the monotonous version [30]. Note that methods that aim at solving the model (4) involve a penalty parameter ν, whilst those tackling (6) need the sparsity level s. To make results comparable, we adjust their default parameters ν for each method to guarantee the generated solution $\mathbf{z}$ satisfying $\|\mathbf{z}\|_0 \leq p/2$. We will report three indicators: ($\ell(\mathbf{z})$, `SER`, `Time`) to illustrate the performance of methods, where `Time` (in seconds) is the CPU time, $\mathbf{z}$ is the solution obtained by each method and `SER` is the sign error rate defined by

$$\texttt{SER} := \frac{1}{n}\sum_{i=1}^{m} |y_i - \text{sign}(\langle \mathbf{x}_i, \mathbf{z}\rangle_+)| \,.$$

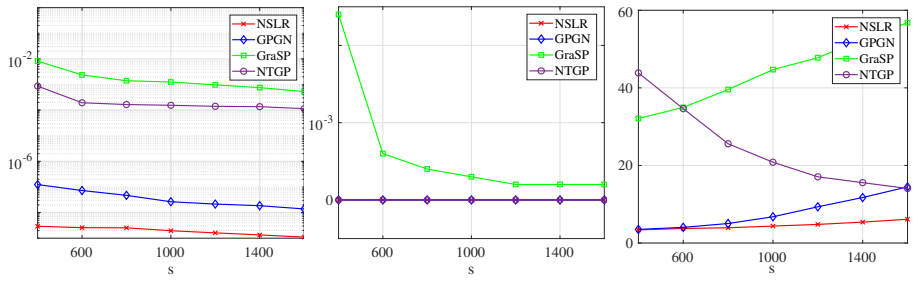
Here $\text{sign}(a_+)$ is the sign of the projection of a onto a non-negative space, namely, it returns 1 if $a > 0$ and 0 otherwise.

4.4. Numerical comparison

We now report the performance of eight methods on the above three examples. To avoid randomness, we report average results over 10-time independent

trails for Examples 4.1 and 4.2 since they involve in randomly generated data.

(a) Comparison on Example 4.1. To observe the influence of the sparsity level s on four greedy methods: `NSLR`, `GPGN`, `GraSP` and `NTGP`, we fix $p = 10000, n = p/5$ and alter $s \in \{400, 600, \cdots, 1600\}$. As demonstrated in Figure 2, `NSLR` outperforms others in terms of the lowest $\ell(\mathbf{z})$ and `SER` and the shortest time, followed by `GPGN`. By contrast, `GraSP` always performs the worst results, which means this first-order method is not competitive when against the other three methods, three second-order methods.



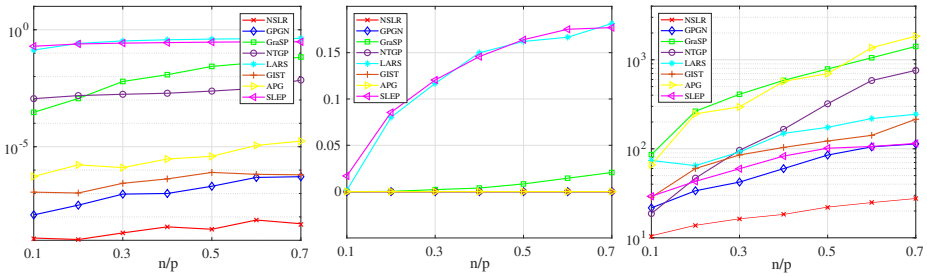
(a) $\ell(\mathbf{z})$ (b) `SER` (c) `Time`

Figure 2: Comparison of four methods for Example 4.1 with $p = 10000, n = 0.2p$

To observe the influence of the ratio of the sample size n and the number of features p on all eight methods, we fix $p = 20000$, $s = 0.1p$ and vary $n/p \in \{0.1, 0.2, \cdots, 0.7\}$. Apart from recording the four indicators, we also report the number of non-zeros of the solution $\mathbf{z}$ generated by each method. Here, for the `LARS`, we stop it when s variables are selected and their default stopping conditions are met since `LARS` only adds one variable at each iteration (see [45]). We set $\nu_1 = 10^{-1}, \nu_2 = 10^{-2}$ for `SLEP`, $\nu_3 = 10^{-2}$, $\nu_4 = 10^{-4}$ for `APG` and $\nu_3 = 10^{-3}$`abs(randn)`, $\nu_4 = 10^{-5}$`abs(randn)` for `GIST`.

As presented in Figure 3, in terms of $\ell(\mathbf{z})$ and `SER`, again `NSLR` performs the best results, followed by `GPGN` and `GIST`. It is obvious that `LARS` and `SLEP` produce undesirable results compared with other methods. For the computational time, `NSLR` runs the fastest, while `GraSP` and `APG` run relatively slow with over 1000 seconds when $n/p \geq 0.6$. Table 3 shows the sparsity levels $\|\mathbf{z}\|_0$ only in

LARS is lower than our NSLR. This is because LARS fails to recover the support and vanishes when $s = 500$ in this numerical experiment (this phenomenon had also been observed in [45, 46].)

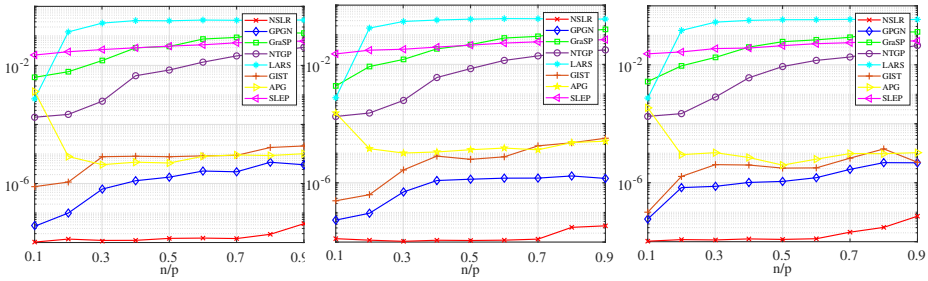


(a) $\ell(\mathbf{z})$ (b) SER (c) Time

Figure 3: Comparison of eight methods for Example 4.1 with $p = 20000, s = 0.1p$.

Table 3: Sparsity levels $\|\mathbf{z}\|_0$ of eight methods for Example 4.1 with $p = 20000, s = 0.1p$.

n/p	0.1	0.2	0.3	0.4	0.5	0.6	0.7
NSLR, GPGN GraSP, NTGP	2000	2000	2000	2000	2000	2000	2000
LARS	500	500	500	500	500	500	500
GIST	4403	4309	5274	7832	7913	8614	8904
APG	6138	5857	5720	6170	5574	5048	5043
SLEP	2076	2534	2980	3235	3498	3596	3873



(a) $\rho = 0$ (b) $\rho = 1/3$ (c) $\rho = 1/2$

Figure 4: $\ell(\mathbf{z})$ obtained by eight methods for Example 4.2 with $p = 1000, s = 0.1p$.

(b) Comparison on Example 4.2. To observe the influence of the correlation parameter ρ on eight methods, we set $p = 1000, s = 0.1p$ but choose $\rho \in \{0, 1/3, 1/2\}$. Figure 4 shows the average $\ell(\mathbf{z})$ gotten by eight methods for a wide range of the ratio $n/p \in \{0.1, 0.2, \cdots, 0.9\}$. Apparently, at three different values of ρ, `NSLR` always performs stably best results. Moreover, the trends in these eight methods perform generally consistent, which indicates the correlation parameter has little influence on these methods. Therefore, we further fix $\rho = 1/2$ and observe the performance of eight methods under higher dimensions.

Table 4: Average results for Example 4.2.

	$s = 0.05p, n = 0.2p$				$s = 0.1p, n = 0.2p$			
	$\ell(\mathbf{z})$	SER	Time	$\|\mathbf{z}\|_0$	$\ell(\mathbf{z})$	SER	Time	$\|\mathbf{z}\|_0$
				$p = 10000$				
NSLR	3.2e-10	0.0e+0	0.436	500	1.1e-10	0.0e+0	0.918	1000
GPGN	1.09e-7	0.0e+0	4.192	500	6.84e-8	0.0e+0	6.962	1000
GraSP	1.20e-2	5.00e-3	17.10	500	7.41e-3	1.70e-3	17.01	1000
NTGP	4.37e-3	0.0e+0	33.54	500	2.18e-3	0.0e+0	13.53	1000
LARS	1.43e-1	1.25e-2	27.69	500	2.85e-1	4.90e-3	71.80	1000
GIST	4.81e-6	0.0e+0	12.54	2381	8.08e-6	0.0e+0	11.40	2974
APG	1.29e-4	0.0e+0	30.00	1407	1.25e-4	0.0e+0	27.82	5211
SLEP	1.75e-1	2.00e-3	6.56	811	1.32e-1	0.0e+0	10.84	1019
				$p = 20000$				
NSLR	1.6e-10	0.0e+0	2.522	1000	5.4e-11	0.0e+0	4.939	2000
GPGN	8.48e-8	0.0e+0	16.47	1000	5.91e-8	0.0e+0	15.83	2000
GraSP	1.29e-2	5.25e-3	62.22	1000	5.00e-3	1.50e-3	92.69	2000
NTGP	5.43e-3	0.0e+0	134.65	1000	2.30e-3	0.0e+0	54.50	2000
LARS	3.96e-1	5.43e-2	107.45	1000	4.21e-1	6.18e-2	117.1	1000
GIST	7.20e-7	0.0e+0	33.52	1542	9.98e-7	0.0e+0	54.55	4137
APG	5.06e-5	0.0e+0	24.94	1849	6.04e-5	0.0e+0	69.46	4323
SLEP	1.88e-1	3.25e-3	22.67	1511	1.45e-1	0.0e+0	34.59	2005
				$p = 30000$				
NSLR	1.1e-10	0.0e+0	7.364	1500	3.8e-11	0.0e+0	12.79	3000
GPGN	8.58e-8	0.0e+0	35.06	1500	4.09e-8	0.0e+0	51.49	3000
GraSP	2.28e-2	7.50e-3	181.9	1500	8.96e-3	2.83e-3	208.1	3000
NTGP	5.36e-3	0.0e+0	307.1	1500	2.32e-3	0.0e+0	125.1	3000
LARS	4.59e-1	1.03e-1	205.7	1000	4.99e-1	1.21e-1	206.7	1000
GIST	2.76e-6	0.0e+0	121.1	6124	7.97e-7	0.0e+0	150.9	6278
APG	9.40e-4	1.67e-4	206.7	7460	3.11e-4	0.0e+0	247.1	6450
SLEP	1.96e-1	2.33e-3	54.20	2392	1.49e-1	1.67e-4	80.19	3009

Now we alter $p \in \{10000, 20000, 30000\}$ with $n = 0.2p$, $s = 0.05p$ or $s = 0.1p$. Here, we set $\nu_1 = 10^{-2}, \nu_2 = 10^{-1}$ for `SLEP` $\nu_3 = 10^{-2}$, $\nu_4 = 5 \times 10^{-4}$ for `APG` and $\nu_3 = 5 \times 10^{-3}$`abs(randn)`, $\nu_4 = 5 \times 10^{-5}$`abs(randn)` for `GIST`. As reported in Table 4, for cases of $p = 20000, 30000$, `LARS` basically fails to render desirable solutions due to the highest $\ell(\mathbf{z})$. It can be clearly seen that `NSLR` always provides the best accuracies with consuming shortest time.

(c) Comparison on Example 4.3. To observe the performance for above all eight methods on real data sets, we select eight real data sets with different dimensions. The highest dimension is up to millions (see `news20.binary`). Table 5 reports results for eight methods on four datasets without testing data ($m_2 = 0$): `arcene`, `colon-cancer`, `news20.binary` and `newsgroup`. For the last two datasets, `LARS` makes our desktop run out of memory, thus its results are omitted here. Clearly, `NSLR` is more efficient than others for all test instances. For example, `NSLR` only uses 1.365 seconds for data `newsgroup` with $p = 777811$ features and achieves the smallest logistic loss with the sparsest solution.

Table 5: Results for Example 4.3 with $m_2 = 0$.

	$\ell(\mathbf{z})$	SER	Time	$\|\|\mathbf{z}\|\|_0$	$\ell(\mathbf{z})$	SER	Time	$\|\|\mathbf{z}\|\|_0$
Data	Arcene				colon-cancer			
NSLR	4.57e-7	0.0e+0	0.14	60	1.90e-8	0.0e+0	0.08	20
GPGN	1.84e-5	0.0e+0	0.58	60	3.35e-5	0.0e+0	0.17	20
GraSP	1.88e-3	0.0e+0	1.23	60	3.03e-1	1.61e-2	0.26	20
NTGP	1.09e-1	1.00e-2	2.81	60	4.51e-3	0.0e+0	0.19	20
LARS	7.98e-2	0.0e+0	0.97	60	2.84e-2	0.0e+0	0.23	30
GIST	3.99e-7	0.0e+0	5.64	134	6.22e-7	0.0e+0	0.13	41
APG	1.84e-5	0.0e+0	3.34	430	1.79e-7	0.0e+0	0.23	20
SLEP	1.05e-1	0.0e+0	3.66	66	1.68e-1	4.84e-2	0.14	23
Data	news20.binary				newsgroup			
NSLR	1.11e-2	3.50e-3	3.236	2500	1.46e-2	8.00e-3	1.365	3000
GPGN	2.94e-2	8.00e-3	25.44	2500	5.17e-2	1.20e-2	30.71	3000
GraSP	2.46e-2	9.25e-3	205.7	2500	5.08e-1	4.75e-2	33.06	3000
NTGP	7.94e-2	6.55e-3	93.20	2500	3.11e-1	5.80e-2	13.07	3000
LARS	—	—	—	—	—	—	—	—
GIST	3.18e-2	6.60e-3	27.57	4091	6.84e-2	1.52e-2	10.55	3017
APG	4.18e-2	1.24e-2	37.15	3869	5.12e-2	1.77e-2	10.62	3257
SLEP	1.49e-1	3.34e-2	43.70	5299	2.60e-1	4.60e-2	23.89	5138

When all methods solve the datasets with testing data ($m_2 > 0$): `duke breast-cancer`, `leukemia`, `gisette` and `rcv1.binary`, the table is a little different. Since the testing data is taken into consideration, we add two indicators to illustrate the performance of each method: $\ell(\mathbf{z})$-test and `SER`-test. Results are reported in Table 6, where $\ell(\mathbf{z})$ and $\ell(\mathbf{z})$-test denote the objective function value on training data and testing data respectively, and similar to `SER`-train and `SER`-test. For cases of `duke breast-cancer` and `leukemia`, `LARS` stops when the maximum number of iterations reaches the $\min\{m_1, p\}$. Hence its produced $||\mathbf{z}||_0$s are less than other methods. We can see that `NSLR` could guarantee a good performance on the testing data as well as the training data.

5. Conclusion

Despite the NP-hardness of the sparsity constrained logistic regression (6), we benefited from its nice properties of the objective function and introduced the τ-stationary point as an optimality condition. This can be converted to an equation system that makes the Newton method effective. Since an order s (which is far smaller than p) principal sub-matrix of the whole Hessian of the objective function is taken into account in each step, the proposed method `NSLR` has a relatively low computational complexity. The success in acquiring the global convergence stems from the realization of the Armijo-type line search in the method. What is more, the generated sequence also converges to a τ-stationary point quadratically, which well justifies the outstanding performance of `NSLR` theoretically. It is worth mentioning that we reasonably extended the classical Newton method for solving unconstrained and continuous problems to the sparsity constrained logistic regression. The numerical performance against several state-of-the-art methods demonstrated that `NSLR` is remarkably efficient and competitive, especially in large scale settings.

We feel that the proposed method might be capable of solving the general strong convex optimization problems with the sparsity constraint. This deserves exploring in future.

Table 6: Results for Example 4.3 with $m_2 > 0$.

	$\ell(\mathbf{z})$	$\ell(\mathbf{z})$-test	SER-train	SER-test	Time	$\|\|\mathbf{z}\|\|_0$
Data			duke breast-cancer			
NSLR	4.10e-9	2.45e-6	0.0e+0	0.0e+0	0.11	100
GPGN	1.31e-5	2.55e-3	0.0e+0	0.0e+0	0.44	100
GraSP	3.57e-3	1.82e-3	0.0e+0	0.0e+0	0.50	100
NTGP	1.21e-5	1.20e-4	0.0e+0	0.0e+0	0.64	100
LARS	1.01e-4	1.21e-4	0.0e+0	0.0e+0	0.61	37
GIST	6.64e-9	1.60e-1	0.0e+0	0.0e+0	0.54	614
APG	6.35e-7	2.68e-7	0.0e+0	0.0e+0	0.74	136
SLEP	1.93e-3	1.04e-2	0.0e+0	0.0e+0	0.43	203
Data			leukemia			
NSLR	3.09e-6	7.22e-2	0.0e+0	0.0e+0	0.11	150
GPGN	1.29e-5	2.71e+0	0.0e+0	1.47e-1	0.35	150
GraSP	3.40e-3	5.08e-1	0.0e+0	1.47e-1	0.56	150
NTGP	4.22e-4	2.11e-1	0.0e+0	5.88e-2	0.75	150
LARS	6.07e-4	1.11e-1	0.0e+0	8.82e-2	1.05	37
GIST	5.68e-3	1.82e-1	0.0e+0	1.18e-1	0.42	295
APG	1.05e-4	3.63e-1	0.0e+0	8.82e-2	0.54	1066
SLEP	1.71e-1	2.85e-1	0.0e+0	2.94e-2	0.58	269
Data			gisette			
NSLR	2.88e-6	6.51e-1	0.0e+0	4.30e-2	1.22	500
GPGN	2.25e-4	2.63e-1	0.0e+0	4.60e-2	1.41	500
GraSP	1.51e-4	2.30e-1	0.0e+0	4.82e-2	2.43	500
NTGP	1.17e-3	9.58e-1	0.0e+0	4.18e-2	2.55	500
LARS	1.63e-1	1.39e+0	1.00e-3	4.18e-2	8.26	500
GIST	2.40e-4	1.02e+0	0.0e+0	4.30e-2	1.78	1303
APG	3.72e-4	1.31e+0	0.0e+0	4.90e-2	1.64	907
SLEP	1.92e-2	8.30e-1	0.0e+0	4.97e-2	2.04	1569
Data			rcv1.binary			
NSLR	5.82e-2	2.01e-1	1.95e-2	5.45e-2	3.71	1000
GPGN	2.90e-2	2.35e-1	8.05e-3	5.58e-2	6.81	1000
GraSP	3.09e-1	1.98e+0	4.15e-2	9.51e-2	21.7	1000
NTGP	7.48e-2	1.37e-1	1.06e-2	4.64e-2	4.47	1000
LARS	2.27e-1	2.61e-1	5.13e-1	5.46e-1	33.5	1000
GIST	3.44e-2	1.41e-1	8.20e-3	4.85e-2	4.57	1545
APG	2.51e-2	2.78e-1	7.31e-3	6.00e-2	7.57	1537
SLEP	1.24e-1	1.65e-1	3.02e-2	5.23e-2	8.45	3527

References

[1] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) (1996) 267–288.

[2] S. Bahmani, B. Raj, P. T. Boufounos, Greedy sparsity-constrained optimization, Journal of Machine Learning Research 14 (Mar) (2013) 807–841.

[3] Y. Plan, R. Vershynin, Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach, IEEE Transactions on Information Theory 59 (1) (2013) 482–494.

[4] R. Wang, N. Xiu, C. Zhang, Greedy projected gradient-newton method for sparse logistic regression, IEEE transactions on neural networks and learning systems 31 (2) (2019) 527–538.

[5] A. Beck, N. Hallak, On the minimization over sparse symmetric sets: projections, optimality conditions, and algorithms, Mathematics of Operations Research 41 (1) (2015) 196–223.

[6] L.-L. Pan, N.-H. Xiu, S.-L. Zhou, On solutions of sparsity constrained optimization, Journal of the Operations Research Society of China 3 (4) (2015) 421–439.

[7] A. Beck, Y. C. Eldar, Sparsity constrained nonlinear optimization: Optimality conditions and algorithms, SIAM Journal on Optimization 23 (3) (2013) 1480–1509.

[8] T. Hastie, R. Tibshirani, R. J. Tibshirani, Extended comparisons of best subset selection, forward stepwise selection, and the lasso, arXiv preprint arXiv:1707.08692 (2017).

[9] R. Mazumder, P. Radchenko, A. Dedieu, Subset selection with shrinkage: Sparse linear modeling when the snr is low, arXiv preprint arXiv:1708.03288 (2017).

[10] H. Hazimeh, R. Mazumder, Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms, Operations Research 68 (5) (2020) 1517–1537.

[11] W. Xie, X. Deng, The CCP selector: Scalable algorithms for sparse ridge regression from chance-constrained programming, arXiv preprint arXiv:1806.03756 (2018).

[12] T. Pang, F. Nie, J. Han, X. Li, Efficient feature selection via $\ell_{2,0}$-norm constrained sparse regression, IEEE Transactions on Knowledge and Data Engineering 31 (5) (2019) 880–893.

[13] M. A. Figueiredo, Adaptive sparseness for supervised learning, IEEE transactions on pattern analysis and machine intelligence 25 (9) (2003) 1150–1159.

[14] B. Krishnapuram, L. Carin, M. A. Figueiredo, A. J. Hartemink, Sparse multinomial logistic regression: Fast algorithms and generalization bounds, IEEE transactions on pattern analysis and machine intelligence 27 (6) (2005) 957–968.

[15] G. Andrew, J. Gao, Scalable training of ℓ_1-regularized log-linear models, in: Proceedings of the 24th international conference on Machine learning, ACM, 2007, pp. 33–40.

[16] K. Koh, S.-J. Kim, S. Boyd, An interior-point method for large-scale ℓ_1-regularized logistic regression, Journal of Machine learning research 8 (Jul) (2007) 1519–1555.

[17] J. Yu, S. Vishwanathan, S. Günter, N. N. Schraudolph, A quasi-newton approach to nonsmooth convex optimization problems in machine learning, Journal of Machine Learning Research 11 (Mar) (2010) 1145–1200.

[18] J. Shi, W. Yin, S. Osher, P. Sajda, A fast hybrid algorithm for large-scale ℓ_1-regularized logistic regression, Journal of Machine Learning Research 11 (Feb) (2010) 713–741.

[19] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, C.-J. Lin, A comparison of optimization methods and software for large-scale ℓ_1-regularized linear classification, Journal of Machine Learning Research 11 (Nov) (2010) 3183–3234.

[20] J. Liu, S. Ji, J. Ye, et al., Slep: Sparse learning with efficient projections, Arizona State University 6 (491) (2009) 7.

[21] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, Journal of statistical software 33 (1) (2010) 1.

[22] G.-X. Yuan, C.-H. Ho, C.-J. Lin, An improved glmnet for ℓ_1-regularized logistic regression, Journal of Machine Learning Research 13 (Jun) (2012) 1999–2030.

[23] J. Liu, J. Chen, J. Ye, Large-scale sparse logistic regression, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 547–556.

[24] S.-I. Lee, H. Lee, P. Abbeel, A. Y. Ng, Efficient l_1 regularized logistic regression, in: AAAI, Vol. 6, 2006, pp. 401–408.

[25] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al., Least angle regression, The Annals of statistics 32 (2) (2004) 407–499.

[26] J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, Journal of the American statistical Association 96 (456) (2001) 1348–1360.

[27] H. Zou, R. Li, One-step sparse estimates in nonconcave penalized likelihood models, Annals of statistics 36 (4) (2008) 1509.

[28] J. Huang, S. Ma, H. Xie, C.-H. Zhang, A group bridge approach for variable selection, Biometrika 96 (2) (2009) 339–355.

[29] P. Gong, C. Zhang, Z. Lu, J. Huang, J. Ye, A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems, in: International Conference on Machine Learning, 2013, pp. 37–45.

[30] H. Li, Z. Lin, Accelerated proximal gradient methods for nonconvex programming, in: Advances in neural information processing systems, 2015, pp. 379–387.

[31] P. Gong, J. Ye, Honor: Hybrid optimization for non-convex regularized problems, in: Advances in Neural Information Processing Systems, 2015, pp. 415–423.

[32] A. Rakotomamonjy, R. Flamary, G. Gasso, Dc proximal newton for nonconvex optimization problems, IEEE transactions on neural networks and learning systems 27 (3) (2016) 636–647.

[33] D. Needell, J. A. Tropp, Cosamp: Iterative signal recovery from incomplete and inaccurate samples, Applied and computational harmonic analysis 26 (3) (2009) 301–321.

[34] A. Lozano, G. Swirszcz, N. Abe, Group orthogonal matching pursuit for logistic regression, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 452–460.

[35] S. Mallat, Z. Zhang, Matching pursuit with time-frequency dictionaries, Tech. rep., Courant Institute of Mathematical Sciences New York United States (1993).

[36] Z. Lu, Y. Zhang, Sparse approximation via penalty decomposition methods, SIAM Journal on Optimization 23 (4) (2013) 2448–2478.

[37] X.-T. Yuan, P. Li, T. Zhang, Gradient hard thresholding pursuit for sparsity-constrained optimization, in: International Conference on Machine Learning, 2014, pp. 127–135.

[38] L. Pan, S. Zhou, N. Xiu, H.-D. Qi, A convergent iterative hard thresholding for nonnegative sparsity optimization, Pacific Journal of Optimization 13 (2) (2017) 325–353.

[39] X.-T. Yuan, Q. Liu, Newton-type greedy selection methods for ℓ_0-constrained minimization, IEEE transactions on pattern analysis and machine intelligence 39 (12) (2017) 2437–2450.

[40] J. Chen, Q. Gu, Fast newton hard thresholding pursuit for sparsity constrained nonconvex optimization, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 757–766.

[41] J. J. Moré, D. C. Sorensen, Computing a trust region step, SIAM Journal on Scientific and Statistical Computing 4 (3) (1983) 553–572.

[42] F. Facchinei, Minimization of sc1 functions and the maratos effect, Operations Research Letters 17 (3) (1995) 131–138.

[43] J. D. Hamilton, Time series analysis, Vol. 2, Princeton university press Princeton, NJ, 1994.

[44] A. Agarwal, S. Negahban, M. J. Wainwright, Fast global convergence rates of gradient methods for high-dimensional statistical recovery, in: Advances in Neural Information Processing Systems, 2010, pp. 37–45.

[45] J. Huang, Y. Jiao, Y. Liu, X. Lu, A constructive approach to l0 penalized regression, Journal of Machine Learning Research 19 (10) (2018).

[46] R. Garg, R. Khandekar, Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 337–344.