

A Pathwise Greedy Selection Framework for Scalable Sparse Regression and Guaranteed Statistical Recovery

Chris Junchi Li[◊]

Department of Electrical Engineering and Computer Sciences[◊]
University of California, Berkeley

October 6, 2024

Abstract

In high-dimensional regression problems, identifying sparse structures is essential for efficient modeling and interpretation. This paper introduces PICASSO (Path-wIse CalibrAted Sparse Shooting algOrithm), a novel pathwise optimization framework designed to enhance both computational efficiency and statistical guarantees in sparse learning. PICASSO employs a greedy selection rule for active set updates and a convex relaxation-based warm start initialization, offering robust convergence properties for general loss functions beyond least squares. We provide theoretical guarantees, proving that PICASSO achieves linear convergence to a sparse local optimum with optimal support recovery. The efficacy of PICASSO is demonstrated through extensive numerical experiments on sparse regression tasks.

Keywords: Sparse regression, high-dimensional data, pathwise optimization, greedy selection, support recovery, PICASSO.

1 Introduction

The growing availability of high-dimensional datasets across fields like genomics and neuroimaging has heightened the need for methods that can handle such data effectively. Sparse learning models, which assume only a small subset of variables are relevant, offer a promising approach. For instance, sparse linear regression models aim to recover a sparse coefficient vector by solving a regularized optimization problem that enforces sparsity.

Recent developments in coordinate optimization methods have provided scalable algorithms for tackling sparse learning problems. However, most existing methods either suffer from computational inefficiencies or lack strong theoretical guarantees. Additionally, methods that rely on non-convex regularizers for bias reduction in variable selection often face challenges in guaranteeing convergence to desirable solutions.

To address these issues, we introduce PICASSO (Path-wIse CalibrAted Sparse Shooting algOrithm), a robust pathwise coordinate optimization algorithm. PICASSO incorporates three key innovations: (1) a new greedy selection rule for updating active sets, (2) a convex relaxation-based warm start initialization, and (3) a pathwise optimization framework that can handle general loss functions. These improvements enable PICASSO to efficiently recover sparse structures with strong statistical guarantees, even in high-dimensional settings.

Backgrounds Modern data acquisition routinely produces massive amount of high dimensional data, where the number of variables d greatly exceeds the sample size n , such as high-throughput genomic data [Neale et al. (2012)] and image data from functional magnetic resonance imaging [Eloyan et al. (2012), Liu, Wang and Zhao (2015)]. To handle high dimensionality, we often

assume that only a small subset of variables are relevant in modeling [Tibshirani (1996)]. Such a parsimonious assumption motivates various sparse learning approaches. Taking sparse linear regression as an example, we consider a linear model $y = X\theta^* + \varepsilon$, where $y \in \mathbb{R}^n$ is the response vector, $X \in \mathbb{R}^{n \times d}$ is the design matrix, $\theta^* = (\theta_1, \dots, \theta_d)^\top \in \mathbb{R}^d$ is the unknown sparse regression coefficient vector, and $\varepsilon \sim N(0, \sigma^2 I_n)$ is the random noise. Here, $I_n \in \mathbb{R}^{n \times n}$ is an identity matrix. Let $\|\cdot\|_2$ denote the ℓ_2 norm, and $\mathcal{R}_\lambda(\theta)$ denote a sparsity-inducing regularizer with a regularization parameter $\lambda > 0$. We can obtain a sparse estimator of θ^* by solving the following regularized least square optimization problem:

$$\min_{\theta \in \mathbb{R}^d} \mathcal{F}_\lambda(\theta) \quad \text{where } \mathcal{F}_\lambda(\theta) = \frac{1}{2n} \|y - X\theta\|_2^2 + \mathcal{R}_\lambda(\theta) \quad (1.1)$$

Popular choices of $\mathcal{R}_\lambda(\theta)$ are usually coordinate decomposable, $\mathcal{R}_\lambda(\theta) = \sum_{j=1}^d r_\lambda(\theta_j)$, including the ℓ_1 [Lasso, Tibshirani (1996)], SCAD [Smooth Clipped Absolute Deviation, Fan and Li (2001)] and MCP [Minimax Concavity Penalty, Zhang (2010)] regularizers. For example, the ℓ_1 regularizer takes $\mathcal{R}_\lambda(\theta) = \lambda \|\theta\|_1 = \lambda \sum_j |\theta_j|$ with $r_\lambda(|\theta_j|) = \lambda |\theta_j|$ for $j = 1, \dots, d$.

The ℓ_1 regularizer is convex and computationally tractable, but often induces large estimation bias, and requires a restrictive irrepresentable condition to attain variable selection consistency [Zhao and Yu (2006), Meinshausen and Bühlmann (2006), Zou (2006)]. To address this issue, nonconvex regularizers such as SCAD and MCP have been proposed to obtain nearly unbiased estimators. Throughout the rest of the paper, we only consider MCP as an example due to space limit, but the extension to SCAD is straightforward. Particularly, let \mathcal{E} be an event, we define $\mathbb{1}_{\{\mathcal{E}\}}$ as an indicator function with $\mathbb{1}_{\{\mathcal{E}\}} = 1$ if \mathcal{E} holds and $\mathbb{1}_{\{\mathcal{E}\}} = 0$ otherwise. Given $\gamma > 1$, MCP has

$$r_\lambda(|\theta_j|) = \lambda \left(|\theta_j| - \frac{\theta_j^2}{2\lambda\gamma} \right) \cdot \mathbb{1}_{\{|\theta_j| < \lambda\gamma\}} + \frac{\lambda^2\gamma}{2} \cdot \mathbb{1}_{\{|\theta_j| \geq \lambda\gamma\}} \quad (1.2)$$

We call γ the concavity parameter of MCP, since it essentially characterizes the concavity of the MCP regularizer: A larger γ implies that the regularizer is less concave. We observe that the MCP regularizer can be written as

$$\mathcal{R}_\lambda(\theta) = \lambda \|\theta\|_1 + \mathcal{H}_\lambda(\theta) \quad (1.3)$$

where $\mathcal{H}_\lambda(\theta) = \sum_{j=1}^d h_\lambda(|\theta_j|)$ is a smooth, concave and also coordinate decomposable function with

$$h_\lambda(|\theta_j|) = -\frac{\theta_j^2}{2\gamma} \cdot \mathbb{1}_{\{|\theta_j| < \lambda\gamma\}} + \frac{\lambda^2\gamma - 2\lambda|\theta_j|}{2} \cdot \mathbb{1}_{\{|\theta_j| \geq \lambda\gamma\}} \quad (1.4)$$

We present several examples of the MCP regularizer in Figure 1. Fan and Li (2001), Zhang (2010) show that the nonconvex regularizer effectively reduces the estimation bias, and achieve better performance than the ℓ_1 regularizer in both parameter estimation and support recovery. Particularly, given a suitable chosen $\gamma < \infty$, they show that there exists a local optimum to (1.1), which attains the oracle properties under much weaker conditions. However, they cannot provide specific algorithms that guarantee such a local optimum in polynomial time due to the nonconvexity.

Typical algorithms for solving (1.1) developed in existing optimization literature include proximal gradient algorithms [Nesterov (2013)] and coordinate optimization algorithms [Luo and Tseng (1992), Shalev-Shwartz and Tewari (2011), Richtárik and Takáč (2014), Lu and Xiao (2015)]. The proximal gradient algorithms need to access all entries of the design matrix X in each iteration for computing a full gradient and a sophisticated line search step. Thus, they are often not scalable

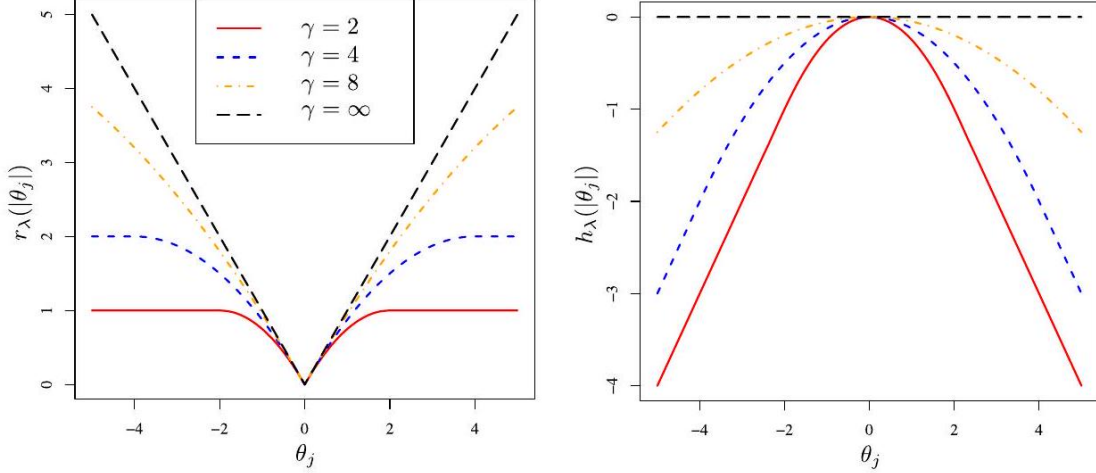


Figure 1. Several examples of the MCP regularizer with $\lambda = 1$ and $\gamma = 2, 4, 8$ and ∞ (Lasso). The MCP regularizer reduces the estimation bias and achieve better performance than the ℓ_1 regularizer in both parameter estimation and support recovery, but imposes great computational challenge.

and efficient in practice when d is large. To address this issue, many researchers resort to the coordinate optimization algorithms for better computational efficiency and scalability.

The classical coordinate optimization algorithm is straightforward and much simpler than the proximal gradient algorithms in each iteration: Given $\theta^{(t)}$ at the t th iteration, we select a coordinate j , and then take an exact coordinate minimization step

$$\theta_j^{(t+1)} = \underset{\theta_j}{\operatorname{argmin}} \mathcal{F}_\lambda(\theta_j, \theta_{\setminus j}^{(t)}) \quad (1.5)$$

where $\theta_{\setminus j}$ is a subvector of θ with the j th entry removed. For the ℓ_1 , SCAD, and MCP regularizers, (1.5) admits a closed-form solution. For notational simplicity, we denote $\theta_j^{(t+1)} = \mathcal{T}_{\lambda,j}(\theta^{(t)})$. Then (1.5) can be rewritten as

$$\theta_j^{(t+1)} = \mathcal{T}_{\lambda,j}(\theta^{(t)}) = \underset{\theta_j}{\operatorname{argmin}} \frac{1}{2n} \|z^{(t)} - X_{*j}\theta_j\|_2^2 + r_\lambda(\theta_j) \quad (1.6)$$

where X_{*j} denotes the j th column of X and $z^{(t)} = y - X\theta^{(t)} + X_{*j}\theta_j^{(t)}$ is the partial residual. Without loss of generality, we assume that X satisfies the column normalization condition $\|X_{*j}\|_2 = \sqrt{n}$ for all $j = 1, \dots, d$. Let $\tilde{\theta}_j^{(t)} = \frac{1}{n} X_{*j}^\top z^{(t)}$. Then for MCP, we obtain $\theta_j^{(t+1)}$ by

$$\theta_j^{(t+1)} = \tilde{\theta}_j^{(t)} \cdot \mathbb{1}_{\{|\tilde{\theta}_j^{(t)}| \geq \gamma\lambda\}} + \frac{\mathcal{S}_\lambda(\tilde{\theta}_j^{(t)})}{1 - 1/\gamma} \cdot \mathbb{1}_{\{|\tilde{\theta}_j^{(t)}| < \gamma\lambda\}} \quad (1.7)$$

where $\mathcal{S}_\lambda(a) = \operatorname{sign}(a) \cdot \max\{|a| - \lambda, 0\}$. As shown in the supplementary materials [Zhao, Liu and Zhang (2018)], (1.7) can be efficiently calculated by a simple partial residual update trick, which only requires the access to one single column of the design matrix X_{*j} (Recall the proximal gradient algorithms need to access the entire design matrix). Once we obtain $\theta_j^{(t+1)}$, we take $\theta_j^{(t+1)} = \theta_j^{(t)}$. Such a coordinate optimization algorithm, though simple, is not necessarily efficient in theory and practice. Existing optimization theory only shows its sublinear convergence to local

optima in high dimensions if we select coordinates from 1 to d in a cyclic order throughout all iterations [Razaviyayn, Hong and Luo (2013), Li et al. (2016)]. Moreover, no theoretical guarantee has been established on statistical properties of the obtained estimators for nonconvex regularizers in parameter estimation and support recovery. Thus, the coordinate optimization algorithms were almost neglected until recent rediscovery by Friedman et al. (2007), Mazumder, Friedman and Hastie (2011), Tibshirani et al. (2012).¹

REMARK 1.1 (Connection between MCP and Lasso). *Let $\frac{c}{\infty} = 0$ for any constant c . As can be seen from (1.2), for $\gamma = \infty$, MCP is reduced to the ℓ_1 regularizer, that is, $r_\lambda(|\theta_j|) = \lambda|\theta_j|$ with $h_\lambda(|\theta_j|) = 0$. Accordingly, (1.7) is reduced to $\theta_j^{(t+1)} = \mathcal{S}_\lambda(\tilde{\theta}_j^{(t)})$, which is identical to the updating formula of the coordinate optimization algorithm proposed in Fu (1998) for Lasso. Thus, throughout the rest of the paper, we just simply consider the ℓ_1 regularizer as a special case of MCP, unless we clearly specify the difference between $\gamma < \infty$ and $\gamma = \infty$ for MCP.*

As illustrated in Figure 2, Hastie and Tibshirani (1990), Mazumder, Friedman and Hastie (2011), Tibshirani et al. (2012) propose a pathwise coordinate optimization framework with three nested loops, which integrates the warm start initialization, active set updating strategy, and strong rule for coordinate preselection into the classical coordinate optimization.

Particularly, in the outer loop, the warm start initialization optimizes (1.1) with a sequence of decreasing regularization parameters in a multistage manner, and yields solutions from sparse to dense. Within each stage of the warm start initialization (an iteration of the outer loop), the algorithm uses the solution from the previous stage for initialization, and then adopts the active set updating strategy to exploit the solution sparsity to speed up computation. The active set updating strategy contains two consequent nested loops: In the middle loop, the algorithm first divides all coordinates into active ones (active set) and inactive ones (inactive set) based on some heuristic coordinate gradient thresholding rule [strong rule, Tibshirani et al. (2012)]. Then within each iteration of the middle loop, an inner loop is called to conduct coordinate optimization. In general, the algorithm runs an inner loop on the current active coordinates until convergence, with all inactive coordinates remain zero. The algorithm then exploits some heuristic rule to identify a new active set, which further decreases the objective value and repeats the inner loops. The iteration within each stage terminates when the active set in the middle loop no longer changes. In practice, the warm start initialization, active set updating strategy, and strong rule for coordinate preselection encourage the algorithm to iterate over a small active set involving only a small number of coordinates and, therefore, significantly boost the computational efficiency and scalability. Software packages such as GLMNET, SparseNet and HUGE have been developed and widely applied to many research areas [Hastie and Tibshirani (1990), Zhao et al. (2012)].

Despite of the popularity of the pathwise coordinate optimization framework, we are still in lack of adequate theory to justify its superior computational performance due to its complex algorithmic structure. The warm start initialization, active set updating strategy, and strong rule for coordinate preselection are only considered as engineering heuristics in existing literature. On the other hand, many experimental results have shown that the pathwise coordinate optimization framework is effective at finding local optima with good empirical performance, yet no theoretical guarantee has been established. Thus, a gap exists between theory and practice.

To bridge this gap, we propose a new algorithm, named PICASSO (Path-wise CalibrAted Sparse Shooting algOrithm), which improves the existing pathwise coordinate optimization framework. Particularly, we propose a new greedy selection rule for active set updating and a new

¹A brief history on applying coordinate optimization to sparse learning problems is presented in Hastie (2009).

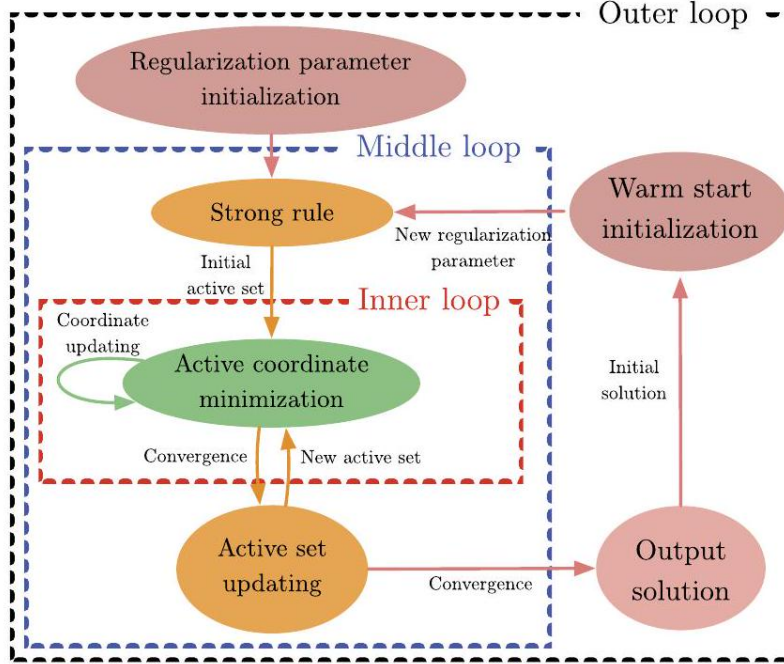


Figure 2. The pathwise coordinate optimization framework contains 3 nested loops: (I) Warm start initialization; (II) Active set updating and strong rule for coordinate preselection; (III) Active coordinate minimization. Many empirical results have corroborated its outstanding performance. Detailed descriptions of the loops are presented in Section 2.

convex relaxation based warm start initialization (for sparse learning problems using general loss functions beyond the least square loss). These modifications though simple, have a profound impact: The solution sparsity and restricted strong convexity can be ensured throughout all iterations, which allows us to establish statistical and computational guarantees of PICASSO in high dimensions [Zhang and Huang (2008), Bickel, Ritov and Tsybakov (2009), Raskutti, Wainwright and Yu (2010)]. Eventually, we prove that PICASSO attains a linear convergence to a unique sparse local optimum with optimal statistical properties in parameter estimation and support recovery (See more details in Section 3). To the best of our knowledge, this is the first result on the computational and statistical guarantees for the pathwise coordinate optimization framework in high dimensions.

Several proximal gradient algorithms are closely related to PICASSO. By exploiting similar sparsity structures of the optimization problem, Wang, Liu and Zhang (2014), Zhao and Liu (2016), Loh and Wainwright (2015) show that these proximal gradient algorithms also attain linear convergence to (approximate) local optima with guaranteed statistical properties. We will compare these algorithms with PICASSO in Section 6.

The rest of this paper is organized as follows: In Section 2, we present the PICASSO algorithm. In Section 3, we present a new theory for analyzing the pathwise coordinate optimization framework, and establish the computational and statistical properties of PICASSO for sparse linear regression. In Section 4, we extend PICASSO to other sparse learning problems with general loss functions, and provide theoretical guarantees; In Section 5, we present thorough numerical experiments to support our theory. In Section 6, we discuss related work. In Section 7, we present the proofs of the theorems. Due to space limit, the proofs of all lemmas are deferred to the supplementary materials

[Zhao, Liu and Zhang (2018)].

NOTATION Given a vector $v = (v_1, \dots, v_d)^\top \in \mathbb{R}^d$, we define vector norms: $\|v\|_1 = \sum_j |v_j|$, $\|v\|_2^2 = \sum_j v_j^2$, and $\|v\|_\infty = \max_j |v_j|$. We denote the number of nonzero entries in v as $\|v\|_0 = \sum_j \mathbb{1}_{\{v_j \neq 0\}}$. We define the soft-thresholding function and operator as $\mathcal{S}_\lambda(v_j) = \text{sign}(v_j) \cdot \max\{|v_j| - \lambda, 0\}$ and $\mathcal{S}_\lambda(v) = (\mathcal{S}_\lambda(v_1), \dots, \mathcal{S}_\lambda(v_d))^\top$. We denote $v_{\setminus j} = (v_1, \dots, v_{j-1}, v_{j+1}, \dots, v_d)^\top \in \mathbb{R}^{d-1}$ as the subvector of v with the j th entry removed. Let $\mathcal{A} \subseteq \{1, \dots, d\}$ be an index set. We use $\overline{\mathcal{A}}$ to denote the complementary set to \mathcal{A} , that is, $\overline{\mathcal{A}} = \{j \mid j \in \{1, \dots, d\}, j \notin \mathcal{A}\}$. We use $v_{\mathcal{A}}$ to denote a subvector of v by extracting all entries of v with indices in \mathcal{A} . Given a matrix $A \in \mathbb{R}^{d \times d}$, we use $A_{*j} = (A_{1j}, \dots, A_{dj})^\top$ to denote the j th column of A , and $A_{k*} = (A_{k1}, \dots, A_{kd})^\top$ to denote the k th row of A . Let $\Lambda_{\max}(A)$ and $\Lambda_{\min}(A)$ be the largest and smallest eigenvalues of A . We define the matrix norms $\|A\|_F^2 = \sum_j \|A_{*j}\|_2^2$ and $\|A\|_2$ as the largest singular value of A . We denote $A_{\setminus i \setminus j}$ as the submatrix of A with the i th row and the j th column removed. We denote $A_{i \setminus j}$ as the i th row of A with its j th entry removed. Let $\mathcal{A} \subseteq \{1, \dots, d\}$ be an index set. We use $A_{\mathcal{A}\mathcal{A}}$ to denote a submatrix of A by extracting all entries of A with both row and column indices in \mathcal{A} .

2 Pathwise calibrated sparse shooting algorithm

We introduce the PICASSO algorithm for sparse linear regression. PICASSO is a pathwise coordinate optimization algorithm and contains three nested loops (as illustrated in Figure 2). For simplicity, we first introduce its inner loop, then its middle loop and at last its outer loop.

2.1 Inner loop: Iterates over coordinates within an active set

We start with the inner loop of PICASSO, which is the active coordinate minimization (ActCooMin) algorithm. The iteration index for the inner loop is (t) , where $t = 0, 1, 2, \dots$. As illustrated in Algorithm 1, the ActCooMin algorithm solves (1.1) by iteratively conducting exact coordinate minimization, but it is only allowed to iterate over a subset of all coordinates, which is called "the active set." Accordingly, the complementary set to the active set is called "the inactive set," because the values of these coordinates do not change throughout all iterations of the inner loop. Since the active set usually contains a very small number of coordinates, the active set coordinate minimization algorithm is very scalable and efficient.

For notational simplicity, we denote the active and inactive sets by \mathcal{A} and $\overline{\mathcal{A}}$, respectively. Here, we select \mathcal{A} and $\overline{\mathcal{A}}$ based on the sparsity pattern of the initial solution of the inner loop $\theta^{(0)}$,

$$\mathcal{A} = \{j \mid \theta_j^{(0)} \neq 0\} \quad \text{and} \quad \overline{\mathcal{A}} = \{j \mid \theta_j^{(0)} = 0\}$$

The ActCooMin algorithm then minimizes (1.1) with all coordinates of $\overline{\mathcal{A}}$ staying at zero values,

$$\min_{\theta \in \mathbb{R}^d} \mathcal{F}_\lambda(\theta) \quad \text{subject to } \theta_{\overline{\mathcal{A}}} = 0 \quad (2.1)$$

The ActCooMin algorithm iterates over all active coordinates in a cyclic order at each iteration. Without loss of generality, we assume

$$|\mathcal{A}| = s \quad \text{and} \quad \mathcal{A} = \{j_1, \dots, j_s\} \subseteq \{1, \dots, d\}$$

Algorithm 1 The active coordinate minimization algorithm (ActCooMin) is the inner loop of PICASSO. It iterates over only a small subset of all coordinates in a cyclic order. Thus, its computation is scalable and efficient. Without loss of generality, we assume $|\mathcal{A}| = s$ and $\mathcal{A} = \{j_1, \dots, j_s\} \subseteq \{1, \dots, d\}$, where $j_1 \leq j_2 \leq \dots \leq j_s$.

Algorithm: $\hat{\theta} \leftarrow \text{ActCooMin}(\lambda, \theta^{(0)}, \mathcal{A}, \tau)$
Initialize: $t \leftarrow 0$
repeat
 $w^{(t+1,0)} \leftarrow \theta^{(t)}$
 for $k \leftarrow 1$ to s **do**
 $w_{j_k}^{(t+1,k)} \leftarrow \mathcal{T}_{\lambda, j_k}(w^{(t+1,k-1)}), w_{j_k}^{(t+1,k)} \leftarrow w_{j_k}^{(t+1,k-1)}$
 end for
 $\theta^{(t+1)} \leftarrow w^{(t+1,s)}$
 $t \leftarrow t + 1$
until $\|\theta^{(t+1)} - \theta^{(t)}\|_2 \leq \tau\lambda$
return $\hat{\theta} \leftarrow \theta^{(t)}$

where $j_1 \leq j_2 \leq \dots \leq j_s$. Given a solution $\theta^{(t)}$ at the t th iteration, we construct a sequence of auxiliary solutions $\{w^{(t+1,k)}\}_{k=0}^s$ to obtain $\theta^{(t+1)}$. Particularly, for $k = 0$, we take $w^{(t+1,0)} = \theta^{(t)}$; For $k = 1, \dots, s$, we take

$$w_{j_k}^{(t+1,k)} = \mathcal{T}_{\lambda, j_k}(w^{(t+1,k-1)}) \quad \text{and} \quad w_{\setminus j_k}^{(t+1,k)} = w_{\setminus j_k}^{(t+1,k-1)}$$

where $\mathcal{T}_{\lambda, j_k}(\cdot)$ is defined in (1.6). We then set $\theta^{(t+1)} = w^{(t+1,s)}$ for the next iteration. Given τ as a small convergence parameter (e.g., 10^{-5}), we terminate the ActCooMin algorithm when

$$\|\theta^{(t+1)} - \theta^{(t)}\|_2 \leq \tau\lambda \quad (2.2)$$

We then take the output solution as $\hat{\theta} = \theta^{(t+1)}$.

The ActCooMin algorithm only converges to a local optimum of (2.1), which is not necessarily a local optimum of (1.1). Thus, PICASSO needs to combine this inner loop with some active set updating scheme, which allows the active set to change. This leads to the middle loop of PICASSO.

2.2 Middle loop: Iteratively updates active sets

We then introduce the middle loop of PICASSO, which is the iterative active set updating (IteActUpd) algorithm. The iteration index of the middle loop is $[m]$, where $m = 0, 1, 2, \dots$. As illustrated in Algorithm 2, the IteActUpd algorithm simultaneously decreases the objective value and iteratively changes the active set to ensure convergence to a local optimum to (1.1). For notational simplicity, we denote the least square loss function and its gradient as $\mathcal{L}(\theta) = \frac{1}{2n} \|y - X\theta\|_2^2$ and $\nabla \mathcal{L}(\theta) = \frac{1}{n} X^\top (X\theta - y)$.

(I) Active Set Initialization by Strong Rule: We first introduce how PICASSO initializes the active set for each middle loop. Suppose an initial solution $\theta^{[0]}$ is supplied to the middle loop of PICASSO. Friedman et al. (2007) suggest a straightforward "simple rule" to initialize the active set based on the sparsity pattern of $\theta^{[0]}$,

$$\mathcal{A}_0 = \{j \mid \theta_j^{[0]} \neq 0\} \quad \text{and} \quad \bar{\mathcal{A}}_0 = \{j \mid \theta_j^{[0]} = 0\} \quad (2.3)$$

Algorithm 2 The iterative active set updating (IteActUpd) algorithm is the middle loop of PICASSO. It simultaneously decreases the objective value and iteratively changes the active set. To encourage the sparsity of the active set, the greedy selection rule moves only one inactive coordinate to the active set in each iteration.

Algorithm: $\hat{\theta} \leftarrow \text{IteActUpd}(\lambda, \theta^{[0]}, \delta, \tau, \varphi)$

Initialize: $m \leftarrow 0, \mathcal{A}_0 \leftarrow \{j \mid \theta_j^{[0]} = 0, |\nabla_j \mathcal{L}(\theta^{[0]})| \geq (1 - \varphi)\lambda\} \cup \{j \mid \theta_j^{[0]} \neq 0\}$

repeat

$\theta^{[m+0.5]} \leftarrow \text{ActCooMin}(\lambda, \theta^{[m]}, \mathcal{A}_m, \tau)$

$\mathcal{A}_{m+0.5} \leftarrow \{j \mid \theta_j^{[m+0.5]} \neq 0\}, \bar{\mathcal{A}}_{m+0.5} \leftarrow \{j \mid \theta_j^{[m+0.5]} = 0\}$

$k_m \leftarrow \arg\max_{k \in \bar{\mathcal{A}}_{m+0.5}} |\nabla_k \mathcal{L}(\theta^{[m+0.5]})|$

$\theta_{k_m}^{[m+1]} \leftarrow \mathcal{T}_{\lambda, k_m}(\theta^{[m+0.5]}), \theta_{\setminus k_m}^{[m+1]} \leftarrow \theta_{\setminus k_m}^{[m+0.5]}$

$\mathcal{A}_{m+1} \leftarrow \mathcal{A}_{m+0.5} \cup \{k_m\}, \bar{\mathcal{A}}_{m+1} \leftarrow \bar{\mathcal{A}}_{m+0.5} \setminus \{k_m\}$

$m \leftarrow m + 1$

until $|\nabla_{k_m} \mathcal{L}(\theta^{[m+0.5]})| \leq (1 + \delta)\lambda$

Return: $\hat{\theta} \leftarrow \theta^{[m]}$

Tibshirani et al. (2012) further show that (2.3) is sometimes too conservative, and suggest a more aggressive active set initialization procedure using a "strong rule," which often leads to better computational performance in practice. Specifically, given an active set initialization parameter $\varphi \in (0, 1)$, the strong rule² for PICASSO initializes \mathcal{A}_0 and $\bar{\mathcal{A}}_0$ as

$$\mathcal{A}_0 = \left\{j \mid \theta_j^{[0]} = 0, |\nabla_j \mathcal{L}(\theta^{[0]})| \geq (1 - \varphi)\lambda\right\} \cup \left\{j \mid \theta_j^{[0]} \neq 0\right\} \quad (2.4)$$

$$\bar{\mathcal{A}}_0 = \left\{j \mid \theta_j^{[0]} = 0, |\nabla_j \mathcal{L}(\theta^{[0]})| < (1 - \varphi)\lambda\right\} \quad (2.5)$$

where $\nabla_j \mathcal{L}(\theta^{[0]})$ denotes the j th entry of $\nabla \mathcal{L}(\theta^{[0]})$. As can be seen from (2.4), the strong rule yields an active set, which is no smaller than the simple rule. Note that we need the initialization parameter φ to be a reasonably small value (e.g., 0.1). Otherwise, the strong rule may select too many active coordinates and compromise the solution sparsity.

(II) Active Set Updating Strategy: We then introduce how PICASSO updates the active set at each iteration of the middle loop. Suppose at the m th iteration ($m \geq 1$), we are supplied with a solution $\theta^{[m]}$ with a pair of active and inactive sets defined as

$$\mathcal{A}_m = \left\{j \mid \theta_j^{[m]} \neq 0\right\} \quad \text{and} \quad \bar{\mathcal{A}}_m = \left\{j \mid \theta_j^{[m]} = 0\right\}$$

Each iteration of the IteActUpd algorithm contains two stages. The first stage conducts the active coordinate minimization algorithm over the active set \mathcal{A}_m until convergence, and returns a solution $\theta^{[m+0.5]}$. Note that the active coordinate minimization algorithm may yield zero values for some active coordinates. Accordingly, we remove those coordinates from the active set, and obtain a new pair of active and inactive sets as

$$\mathcal{A}_{m+0.5} = \left\{j \mid \theta_j^{[m+0.5]} \neq 0\right\} \quad \text{and} \quad \bar{\mathcal{A}}_{m+0.5} = \left\{j \mid \theta_j^{[m+0.5]} = 0\right\}$$

²Our proposed strong rule for PICASSO is slightly different from the sequential strong rule proposed in Tibshirani et al. (2012). See more details in Remark 2.2.

The second stage checks which inactive coordinates of $\bar{\mathcal{A}}_{m+0.5}$ should be added into the active set. Existing pathwise coordinate optimization algorithms usually add inactive coordinates into the active set based on a cyclic selection rule [Friedman et al. (2007), Mazumder, Friedman and Hastie (2011)]. Particularly, they conduct the exact coordinate minimization over all coordinates of $\bar{\mathcal{A}}_{m+0.5}$ in a cyclic order. Accordingly, an inactive coordinate is added into the active set if the corresponding exact coordinate minimization step yields a nonzero value. Such a cyclic selection rule, however, has no control over the solution sparsity. It may add too many inactive coordinates into the active set, and compromise the solution sparsity.

To address this issue, we propose a new greedy selection rule for updating the active set. Particularly, let $\nabla_j \mathcal{L}(\theta^{[m+0.5]})$ denote the j th entry of $\nabla \mathcal{L}(\theta^{[m+0.5]})$. We select a coordinate by

$$k_m = \operatorname{argmax}_{k \in \bar{\mathcal{A}}_{m+0.5}} \left| \nabla_k \mathcal{L}(\theta^{[m+0.5]}) \right|$$

We then terminate the `IteActUpd` algorithm if

$$\left| \nabla_{k_m} \mathcal{L}(\theta^{[m+0.5]}) \right| \leq (1 + \delta) \lambda \quad (2.6)$$

where δ is a small convergence parameter (e.g., 10^{-5}). Otherwise, we take

$$\theta_{k_m}^{[m+1]} = \mathcal{T}_{\lambda, k_m}(\theta^{[m+0.5]}) \quad \text{and} \quad \theta_{\setminus k_m}^{[m+1]} = \theta_{\setminus k_m}^{[m+0.5]}$$

and set the new active and inactive sets as

$$\mathcal{A}_{m+1} = \mathcal{A}_{m+0.5} \cup \{k_m\} \quad \text{and} \quad \bar{\mathcal{A}}_{m+1} = \bar{\mathcal{A}}_{m+0.5} \setminus \{k_m\}$$

REMARK 2.1. *Beside the greedy selection rule, we also propose a randomized selection rule and a truncated cyclic selection rule for updating the active set. Due to space limit, we defer the details to the supplementary materials [Zhao, Liu and Zhang (2018)].*

The `IteActUpd` algorithm, though equipped with the proposed greedy selection rule and strong rule for coordinate preselection, ensures the solution sparsity throughout iterations only for a sufficiently large regularization parameter.³ Otherwise, given an insufficiently large regularization parameter, the `IteActUpd` algorithm may still overselect the active coordinates. To address this issue, we combine the `IteActUpd` algorithm with a sequence of decreasing regularization parameters, which leads to the outer loop of PICASSO.

2.3 Outer loop: Iterates over regularization parameters

The outer loop of PICASSO is the warm start initialization (`WarmStartInt`). The iteration index of the outer loop is $\{K\}$, where $K = 1, \dots, N$. As illustrated in Algorithm 3, the warm start initialization solves (1.1) indexed by a geometrically decreasing sequence of regularization parameters $\{\lambda_K = \lambda_0 \eta^K\}_{K=0}^N$ with a common ratio $\eta \in (0, 1)$, and outputs a sequence of $N + 1$ solutions $\{\hat{\theta}^{(K)}\}_{K=0}^N$, which is also called the solution path.

³As will be shown in Section 3, the choice of λ is determined by the initial solution of the middle loop.

Algorithm 3 The warm start initialization is the **outer loop** of PICASSO. It solves (1.1) with respect to a decreasing sequence of regularization parameters $\{\lambda_K\}_{K=0}^N$. The leading regularization parameter λ_0 is chosen as $\lambda_0 = \|\nabla \mathcal{L}(0)\|_\infty$, which yields an all zero output solution $\hat{\theta}^{\{0\}} = 0$. For $K = 1, \dots, N$, we solve (1.1) for λ_K using $\hat{\theta}^{\{K-1\}}$ as an initial solution. $\{\tau_K\}_{K=1}^N$ and $\{\delta_K\}_{K=1}^N$ are two sequences of small convergence parameters, where τ_K and δ_K correspond to the K th outer loop iteration with the regularization parameter λ_K .

Algorithm: $\{\hat{\theta}^{\{K\}}\}_{K=0}^N \leftarrow \text{WarmStartInt}(\{\lambda_K\}_{K=0}^N)$

Parameters: $\eta, \varphi, \{\tau_K\}_{K=1}^N, \{\delta_K\}_{K=1}^N$

Initialize: $\lambda_0 \leftarrow \|\nabla \mathcal{L}(0)\|_\infty, \hat{\theta}^{\{0\}} \leftarrow 0$

for $K \leftarrow 1, 2, \dots, N$ **do**

$\lambda_K \leftarrow \eta \lambda_{K-1}$

$\hat{\theta}^{\{K\}} \leftarrow \text{IteActUpd}(\lambda_K, \hat{\theta}^{\{K-1\}}, \delta_K, \tau_K, \varphi)$

end for

return $\{\hat{\theta}^{\{K\}}\}_{K=0}^N$

For sparse linear regression,⁴ the warm start initialization chooses the leading regularization parameter λ_0 as $\lambda_0 = \|\nabla \mathcal{L}(0)\|_\infty = \left\| \frac{1}{n} X^\top y \right\|_\infty$. Recall $\mathcal{H}_\lambda(\theta)$ is defined in (1.3). By verifying the KKT condition, we have

$$\min_{\xi \in \partial \|0\|_1} \|\nabla \mathcal{L}(0) + \nabla \mathcal{H}_{\lambda_0}(0) + \lambda_0 \xi\|_\infty = \min_{\xi \in \partial \|0\|_1} \|\nabla \mathcal{L}(0) + \lambda_0 \xi\|_\infty = 0$$

where the first equality comes from $\nabla \mathcal{H}_{\lambda_0}(0) = 0$ for the MCP regularizer (See more details in the supplementary materials [Zhao, Liu and Zhang (2018)]). This indicates that 0 is a local optimum of (1.1). Accordingly, we set $\hat{\theta}^{\{0\}} = 0$. Then for $K = 1, 2, \dots, N$, we solve (1.1) for λ_K using $\hat{\theta}^{\{K-1\}}$ as initialization.

The warm start initialization starts with large regularization parameters to suppress the overselection of the irrelevant coordinates $\{j \mid \theta_j^* = 0\}$ (in conjunction with the IteActUpd algorithm). Thus, the solution sparsity ensures the restricted convexity throughout all iterations, making the algorithm behave as if minimizing a strongly convex function. Though large regularization parameters may also yield zero values for many relevant coordinates $\{j \mid \theta_j^* \neq 0\}$ and result in larger estimation errors, this can be compensated by the decreasing regularization sequence. Eventually, PICASSO gradually recovers the relevant coordinates, reduces the estimation error of each output solution and attains a sparse output solution with optimal statistical properties in parameter estimation and support recovery.

REMARK 2.2 (Connection to the sequential strong rule). *Tibshirani et al. (2012) propose a sequential strong rule for coordinate preselection, which initializes the active set for λ_K as*

$$\mathcal{A}_0 = \left\{ j \mid \theta_j^{[0]} = 0, |\nabla_j \mathcal{L}(\theta^{[0]})| \geq 2\lambda_K - \lambda_{K-1} \right\} \cup \left\{ j \mid \theta_j^{[0]} \neq 0 \right\} \quad (2.7)$$

$$\overline{\mathcal{A}}_0 = \left\{ j \mid \theta_j^{[0]} = 0, |\nabla_j \mathcal{L}(\theta^{[0]})| < 2\lambda_K - \lambda_{K-1} \right\} \quad (2.8)$$

Recall $\lambda_K = \eta \lambda_{K-1}$. Then we have $2\lambda_K - \lambda_{K-1} = (1 - (1 - \eta)/\eta)\lambda_K$. This indicates that the sequential strong rule is a special case of our strong rule for PICASSO with $\varphi = (1 - \eta)/\eta$.

⁴When dealing with general loss functions, we need a new convex relaxation based warm start initialization approach, which will be introduced in Section 4.2.

3 Computational and statistical theory

We develop a new theory to analyze the pathwise coordinate optimization framework, and establish the computational and statistical properties of PICASSO for sparse linear regression. Recall our linear model assumption is $y = X\theta^* + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2 I_n)$ ⁵. Moreover, in (1.3), we rewrite the nonconvex regularizer as $\mathcal{R}_\lambda(\theta) = \lambda\|\theta\|_1 + \mathcal{H}_\lambda(\theta)$, where $\mathcal{H}_\lambda(\theta) = \sum_{j=1}^d h_\lambda(|\theta_j|)$ is a smooth, concave and coordinate decomposable function. For notational simplicity, we define $\tilde{\mathcal{L}}_\lambda(\theta) = \mathcal{L}(\theta) + \mathcal{H}_\lambda(\theta)$. Accordingly, we rewrite $\mathcal{F}_\lambda(\theta)$ as

$$\mathcal{F}_\lambda(\theta) = \mathcal{L}(\theta) + \mathcal{R}_\lambda(\theta) = \tilde{\mathcal{L}}_\lambda(\theta) + \lambda\|\theta\|_1$$

3.1 Computational theory

We first introduce three assumptions. The first assumption requires λ_N to be sufficiently large.

ASSUMPTION 3.1. *We require that the regularization sequence satisfies*

$$\lambda_N = 8\sigma\sqrt{\frac{\log d}{n}} \geq 4\|\nabla\mathcal{L}(\theta^*)\|_\infty = \frac{4}{n}\|X^\top\varepsilon\|_\infty \quad (3.1)$$

Moreover, we require $\eta \in [0.96, 1)$.

Assumption 3.1 ensures that all regularization parameters are sufficiently large to eliminate the irrelevant coordinates for PICASSO.

REMARK 3.2. *Note that Assumption 3.1 is a deterministic bound for our chosen λ_N . As will be shown in Lemma 3.13, since $\|X^\top\varepsilon\|_\infty$ is random, we need to verify that (3.1) holds with high probability when applying PICASSO to sparse linear regression.*

Before we present the second assumption, we define the largest and smallest s sparse eigenvalues of the Hessian matrix $\nabla^2\mathcal{L}(\theta) = \frac{1}{n}X^\top X$ as follows.

DEFINITION 3.3. *Given an integer $s \geq 1$, we define the largest and smallest s sparse eigenvalues as*

$$\rho_+(s) = \sup_{\|v\|_0 \leq s} \frac{v^\top \nabla^2 \mathcal{L}(\theta) v}{\|v\|_2^2} \quad \text{and} \quad \rho_-(s) = \inf_{\|v\|_0 \leq s} \frac{v^\top \nabla^2 \mathcal{L}(\theta) v}{\|v\|_2^2}$$

The next lemma connects the largest and smallest s sparse eigenvalues to the restricted strong convexity and smoothness.

Lemma 3.4. *Suppose there exists an integer s such that $0 < \rho_-(s) \leq \rho_+(s) < \infty$. For any $\theta, \theta' \in \mathbb{R}^d$ satisfying $\|\theta - \theta'\|_0 \leq s$, $\mathcal{L}(\theta)$ is restricted strongly convex and smooth,*

$$\frac{\rho_-(s)}{2}\|\theta' - \theta\|_2^2 \leq \mathcal{L}(\theta') - \mathcal{L}(\theta) - (\theta' - \theta)^\top \nabla \mathcal{L}(\theta) \leq \frac{\rho_+(s)}{2}\|\theta' - \theta\|_2^2 \quad (3.2)$$

Moreover, given $\alpha = 1/\gamma \leq \rho_-(s)$ and $\tilde{\rho}_-(s) = \rho_-(s) - \alpha > 0$, where γ is the concavity parameter of MCP defined in (1.2), for any $\theta, \theta' \in \mathbb{R}^d$ satisfying $\|\theta - \theta'\|_0 \leq s$, $\tilde{\mathcal{L}}_\lambda(\theta)$ is restricted strongly convex and smooth,

$$\frac{\tilde{\rho}_-(s)}{2}\|\theta' - \theta\|_2^2 \leq \tilde{\mathcal{L}}_\lambda(\theta') - \tilde{\mathcal{L}}_\lambda(\theta) - (\theta' - \theta)^\top \nabla \tilde{\mathcal{L}}_\lambda(\theta) \leq \frac{\rho_+(s)}{2}\|\theta' - \theta\|_2^2$$

⁵For simplicity, we only consider the Gaussian noise setting, but it is straightforward to extend our analysis to the sub-Gaussian noise setting.

Meanwhile, for any $\xi \in \partial\|\theta\|_1$, $\mathcal{F}_\lambda(\theta)$ is restricted strongly convex,

$$\frac{\tilde{\rho}_-(s)}{2} \|\theta' - \theta\|_2^2 \leq \mathcal{F}_\lambda(\theta') - \mathcal{F}_\lambda(\theta) - (\theta' - \theta)^\top (\nabla \tilde{\mathcal{L}}_\lambda(\theta) + \lambda \xi)$$

Lemma 3.4 indicates the importance of the solution sparsity: When θ is sufficiently sparse, the restricted strong convexity of $\mathcal{L}(\theta)$ dominates the concavity of $\mathcal{H}_\lambda(\theta)$. Thus, if an algorithm ensures the solution sparsity throughout all iterations, it will behave like minimizing a strongly convex optimization problem. Accordingly, a linear convergence can be established. Note that Lemma 3.4 is also applicable to Lasso, since Lasso satisfies $\alpha = 1/\gamma = 1/\infty = 0$. Now we introduce the second assumption.

ASSUMPTION 3.5. *Given $\|\theta^*\|_0 \leq s^*$, there exists an integer \tilde{s} such that*

$$\tilde{s} \geq (484\kappa^2 + 100\kappa)s^*, \quad \rho_+(s^* + 2\tilde{s}) < \infty, \quad \text{and} \quad \tilde{\rho}_-(s^* + 2\tilde{s}) > 0$$

where κ is defined as $\kappa = \rho_+(s^* + 2\tilde{s})/\tilde{\rho}_-(s^* + 2\tilde{s})$.

Assumption 3.5 guarantees that the optimization problem satisfies the restricted strong convexity as long as the number of active irrelevant coordinates never exceeds \tilde{s} throughout all iterations.

REMARK 3.6. *Assumptions 3.1 and 3.5 are closely related to high dimensional statistical theories for sparse linear regression in existing literature. See more details in Zhang and Huang (2008), Bickel, Ritov and Tsybakov (2009), Zhang (2010), Negahban et al. (2012).*

Now we introduce the last assumption on the computational parameters.

ASSUMPTION 3.7. Recall the convergence parameters δ_K 's and τ_K 's are defined in Algorithm 3, and the active set initialization parameter φ is defined in (2.4). We require for all $K = 1, \dots, N$,

$$\delta_K \leq \frac{1}{8}, \quad \tau_K \leq \frac{\delta_K}{\rho_+(s^* + 2\tilde{s})} \sqrt{\frac{\tilde{\rho}_-(1)}{\rho_+(1)(s^* + 2\tilde{s})}} \quad \text{and} \quad \varphi \leq \frac{1}{8}$$

Assumption 3.7 guarantees that all middle and inner loops of PICASSO attain adequate precisions such that their output solutions satisfy the desired computational and statistical properties.

REMARK 3.8. *All constants in our technical assumptions and proof are for providing insights of PICASSO. We do not make efforts on optimizing any of these constants. Taking Assumption 3.1 as an example, we choose $\eta \in [0.96, 1)$ just for easing our analysis. We can also choose η to be any other constant, for example, 0.95, as long as it is sufficiently close to 1. Such a change in η only affects the required sample complexity, iteration complexity and statistical rates of convergence up to small constant factors.*

Now, we start with the convergence analysis for the inner loop of PICASSO. The following theorem presents the convergence rate in terms of the objective value. For notational simplicity, we omit the outer loop index K , and denote λ_K and τ_K by λ and τ , respectively.

THEOREM 3.9 (Inner loop). *Suppose Assumption 3.5 holds. If the initial active set satisfies $|\mathcal{A}| = s \leq s^* + 2\tilde{s}$, then (2.1) is essentially strongly convex. For $t = 1, 2, \dots$, we have*

$$\mathcal{F}_\lambda(\theta^{(t)}) - \mathcal{F}_\lambda(\bar{\theta}) \leq \left(\frac{s\rho_+^2(s)}{s\rho_+^2(s) + \tilde{\rho}_-(s)\tilde{\rho}_-(1)} \right)^t \left[\mathcal{F}_\lambda(\theta^{(0)}) - \mathcal{F}_\lambda(\bar{\theta}) \right]$$

where $\bar{\theta}$ is a unique global optimum to (2.1). Moreover, we need at most

$$\left(1 + \frac{s\rho_+^2(s)}{\tilde{\rho}_-(s)\tilde{\rho}_-(1)}\right) \cdot \log \left(\frac{2[\mathcal{F}_\lambda(\theta^{(0)}) - \mathcal{F}_\lambda(\bar{\theta})]}{\tilde{\rho}_-(1)\tau^2\lambda^2} \right)$$

iterations to terminate the ActCooMin algorithm, where τ is defined in (2.2).

Theorem 3.9 guarantees that given a sufficiently sparse active set, Algorithm 1 essentially minimizes a strongly convex optimization problem, though (1.1) is globally nonconvex. Thus, it attains a linear convergence to a unique global optimum.

Then we proceed with the convergence analysis for the middle loop of PICASSO. The following theorem presents the convergence rate in terms of the objective value. For notational simplicity, we omit the outer loop index K , and denote λ_K and δ_K by λ and δ , respectively. Moreover, we define

$$\Delta_\lambda = \frac{4\lambda^2 s^*}{\tilde{\rho}_-(s^* + \tilde{s})}, \quad \mathcal{S} = \{j \mid \theta_j^* \neq 0\} \quad \text{and} \quad \bar{\mathcal{S}} = \{j \mid \theta_j^* = 0\} \quad (3.3)$$

THEOREM 3.10 (Middle loop). *Suppose Assumptions 3.1, 3.5 and 3.7 hold. For any $\lambda \geq \lambda_N$, if the initial solution $\theta^{[0]}$ satisfies $\|\theta_{\mathcal{S}}^{[0]}\|_0 \leq \tilde{s}$ and $\mathcal{F}_\lambda(\theta^{[0]}) \leq \mathcal{F}_\lambda(\theta^*) + \Delta_\lambda$, then regardless the active set initialized by either the strong rule or simple rule, we have $|\mathcal{A}_0 \cap \bar{\mathcal{S}}| \leq \tilde{s}$. Meanwhile, for $m = 0, 1, 2, \dots$, we also have $\|\theta_{\mathcal{S}}^{[m]}\|_0 \leq \tilde{s} + 1$, $\|\theta_{\bar{\mathcal{S}}}^{[m+0.5]}\|_0 \leq \tilde{s}$, and*

$$\mathcal{F}_\lambda(\theta^{[m]}) - \mathcal{F}_\lambda(\bar{\theta}^\lambda) \leq \left(1 - \frac{\tilde{\rho}_-(s^* + 2\tilde{s})}{(s^* + 2\tilde{s})\rho_+(1)}\right)^m [\mathcal{F}_\lambda(\theta^{(0)}) - \mathcal{F}_\lambda(\bar{\theta}^\lambda)]$$

where $\bar{\theta}^\lambda$ is a unique sparse local optimum of (1.1) satisfying

$$\mathcal{K}_\lambda(\bar{\theta}^\lambda) = \min_{\xi \in \partial \|\bar{\theta}^\lambda\|_1} \|\nabla \tilde{\mathcal{L}}_\lambda(\bar{\theta}^\lambda) + \lambda \xi\|_\infty = 0 \quad \text{and} \quad \|\bar{\theta}_{\bar{\mathcal{S}}}^\lambda\|_0 \leq \tilde{f} \quad (3.4)$$

Moreover, recall δ is defined in (2.6), we need at most

$$\frac{(s^* + 2\tilde{s})\rho_+(1)}{\tilde{\rho}_-(s^* + 2\tilde{s})} \cdot \log \left(\frac{3\rho_+(1) [\mathcal{F}_\lambda(\theta^{[0]}) - \mathcal{F}_\lambda(\bar{\theta}^\lambda)]}{\delta^2\lambda^2} \right)$$

active set updating iterations to terminate the IteActUpd algorithm. Meanwhile, we have the output solution $\hat{\theta}^\lambda$ satisfying $\mathcal{K}_\lambda(\hat{\theta}^\lambda) \leq \delta\lambda$.

Theorem 3.10 guarantees that when supplied a proper initial solution, the middle loop of PICASSO attains a linear convergence to a unique sparse local optimum. Moreover, Theorem 3.10 has three important implications:

(I) The greedy rule is conservative and only select one coordinate each time. This mechanism prevents the overselection of the irrelevant coordinates and encourages the solution sparsity. In contrast, the cyclic selection rule in Mazumder, Friedman and Hastie (2011) may overselect the irrelevant coordinates and compromise the restricted convexity. An illustration is provided in Figure 3.

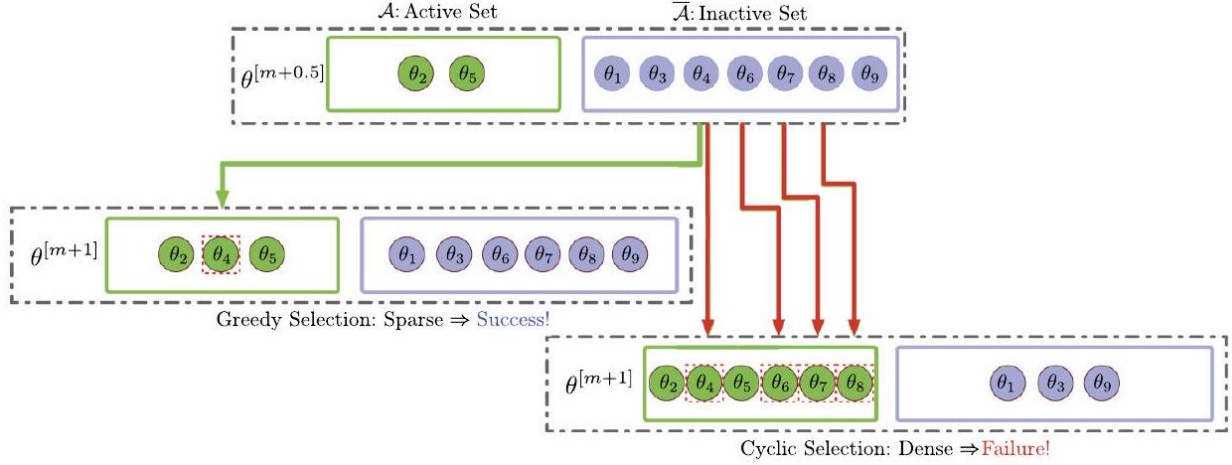


Figure 3. An illustration of the failure of the cyclic selection rule. The green and blue circles denote the active and inactive coordinates, respectively. Suppose we have 9 coordinates and the maximum number of active coordinates we can tolerate is 4. The greedy selection rule is conservative, and only add one coordinate to the active set each time. Thus, it eventually increases the number of active coordinates from 2 to 3, and prevents from overselecting coordinates. In contrast, the cyclic selection rule used in Friedman et al. (2007), Mazumder, Friedman and Hastie (2011) leads to overselecting coordinates, which eventually increases the number of active coordinates to 6. Thus, it fails to preserve the restricted strong convexity.

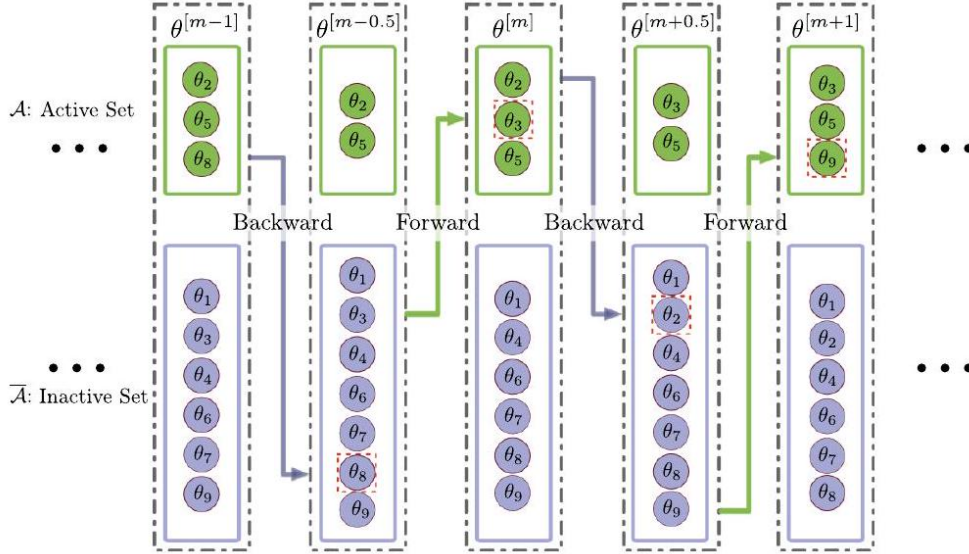


Figure 4. An illustration of the active set updating algorithm. The green and blue circles denote the active and inactive coordinates respectively. Suppose we have 9 coordinates, and the maximum number of active coordinates we can tolerate is 4. The active set updating iteration first removes some active coordinates from the active set, then add some inactive coordinates into the active set. Thus, the number of active coordinates is ensured to never exceed 4 throughout all iterations. To the best of our knowledge, such a "forward-backward" phenomenon has not been discovered and rigorously characterized in existing literature.

(II) Besides decreasing the objective value, the active coordinate minimization algorithm can remove some irrelevant coordinates from the active set. Thus, in conjunction with the greedy selection

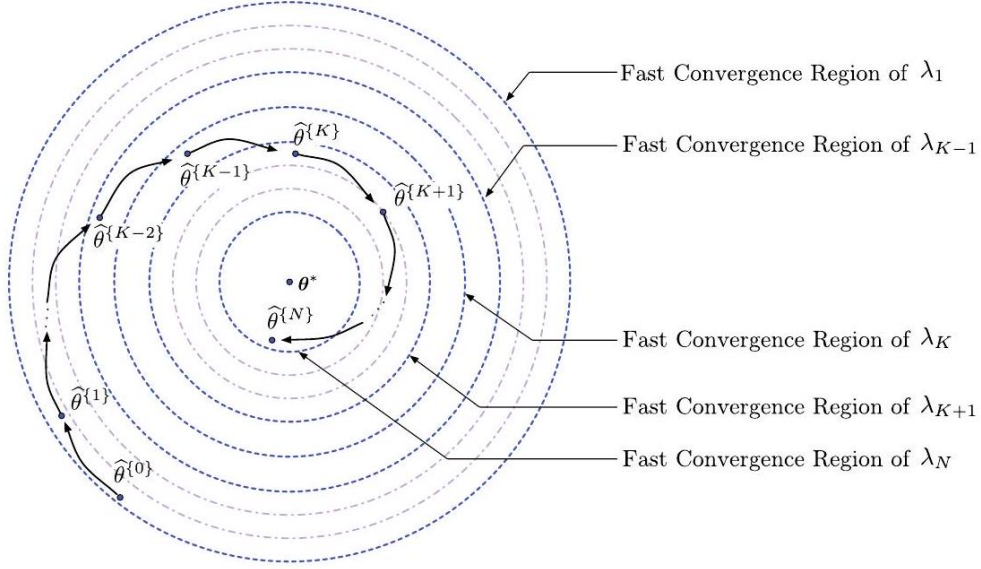


Figure 5. An illustration of the warm start initialization (the outer loop). From an intuitive geometric perspective, the warm start initialization yields a sequence of nested fast convergence regions. We start with large regularization parameters. This suppresses the overselection of the irrelevant coordinates $\{j \mid \theta_j^* = 0\}$ and yields highly sparse solutions. With the decrease of the regularization parameter, PICASSO gradually recovers the relevant coordinates, and eventually obtains a sparse estimator $\hat{\theta}^{(N)}$ with optimal statistical properties in both parameter estimation and support recovery.

rule, the solution sparsity is ensured throughout all iterations. An illustration is provided in Figure 4. To the best of our knowledge, such a "forward-backward" phenomenon has not been discovered and rigorously characterized in existing literature.

(III) The strong rule for coordinate preselection in PICASSO put some coordinates with zero values to the active set, only when their corresponding coordinate gradients have sufficiently large magnitudes. Thus, it can also prevent the overselection of the irrelevant coordinates and ensure the solution sparsity.

Next, we proceed with the convergence analysis for the outer loop of PICASSO. As has been shown in Theorem 3.10, each middle loop of PICASSO requires a proper initialization. Since θ^* and \mathcal{S} are unknown in practice, it is difficult to manually pick such an initial solution. The next theorem shows that the warm start initialization guides PICASSO to attain such a proper initialization for every middle loop without any prior knowledge.

Lemma 3.11 (Outer loop). *Recall Δ_{λ_K} and $\mathcal{K}_{\lambda_K}(\theta)$ are defined in (3.3) and (3.4), respectively. Suppose Assumptions 3.1, 3.5 and 3.7 hold. If θ satisfies $\|\theta_{\mathcal{S}}\|_0 \leq \tilde{s}$ and $\mathcal{K}_{\lambda_{K-1}}(\theta) \leq \delta_{K-1}\lambda_{K-1}$, then we have*

$$\begin{aligned} \|\hat{\Delta}\|_1 &\leq 11\|\hat{\Delta}_{\mathcal{S}}\|_1 \leq 11\sqrt{s^*}\|\hat{\Delta}\|_2 \\ \mathcal{K}_{\lambda_K}(\theta) &\leq \frac{\lambda_K}{4} \\ \mathcal{F}_{\lambda_K}(\theta) &\leq \mathcal{F}_{\lambda_K}(\theta^*) + \Delta_{\lambda_K} \end{aligned}$$

The warm start initialization starts with an all zero local optimum and a sufficiently large λ_0 , which naturally satisfy all requirements:

$$\|0_{\tilde{S}}\|_0 \leq \tilde{s} \quad \text{and} \quad \mathcal{K}_{\lambda_0}(0) = 0$$

Thus, $\theta^{[0]} = 0$ is a proper initial solution for λ_1 . Then combining Theorems 3.10 and 3.11, we show by induction that the output solution of each middle loop is always a proper initial solution for the next middle loop. The warm start initialization is illustrated in Figure 5.

Combining Theorems 3.9 and 3.10 with Lemma 3.11, we establish the global convergence in terms of the objective value for PICASSO.

THEOREM 3.12 (Main theorem). *Suppose Assumptions 3.1, 3.5 and 3.7 hold. Recall $\alpha = 1/\gamma$ and γ is the concavity parameter defined in (1.2), δ_K 's and τ_K 's are the convergence parameters for the middle and inner loops within the K th iteration of the outer loop, and κ and \tilde{s} are defined in Assumption 3.5. For $K = 1, \dots, N$, we have:*

(I) *At the K th iteration of the outer loop, the number of exact coordinate minimization iterations within each inner loop is at most*

$$\left(s^* + 2\tilde{s} + \frac{(s^* + 2\tilde{s})^2 \rho_+^2(s^* + 2\tilde{s})}{\tilde{\rho}_-(s^* + 2\tilde{s})\tilde{\rho}_-(1)} \right) \cdot \log \left(\frac{50s^*}{\tilde{\rho}_-(1)\tau_K^2 \tilde{\rho}_-(s^* + \tilde{s})} \right)$$

(II) *At the K th iteration of the outer loop, the number of active set updating iterations is at most*

$$\frac{(s^* + 2\tilde{s})\rho_+(1)}{\tilde{\rho}_-(s^* + 2\tilde{s})} \cdot \log \left(\frac{75s^*\rho_+(1)}{\delta_K^2 \tilde{\rho}_-(s^* + \tilde{s})} \right)$$

(III) *At the K th iteration of the outer loop, we have*

$$\mathcal{F}_{\lambda_N}(\hat{\theta}^{\{K\}}) - \mathcal{F}_{\lambda_N}(\bar{\theta}^{\lambda_N}) \leq [\mathbb{1}_{\{K < N\}} + \mathbb{1}_{\{K = N\}} \cdot \delta_N] \frac{50\lambda_K^2 s^*}{\tilde{\rho}_-(s^* + \tilde{s})}$$

Theorem 3.12 guarantees that PICASSO attains a linear convergence to a unique sparse local optimum, which is a significant improvement over sublinear convergence of the randomized coordinate minimization algorithms established in existing literature. To the best of our knowledge, this is the first result establishing the convergence properties of the pathwise coordinate optimization framework in high dimensions.

3.2 Statistical theory

Finally, we analyze the statistical properties of the estimator obtained by PICASSO for sparse linear regression. We assume $\|\theta^*\|_0 \leq s^*$, and for any $v \neq 0$, the design matrix X satisfies

$$\psi_\ell \|v\|_2^2 - \gamma_\ell \cdot \frac{\log d}{n} \|v\|_1^2 \leq \frac{\|Xv\|_2^2}{n} \leq \psi_u \|v\|_2^2 + \gamma_u \cdot \frac{\log d}{n} \|v\|_1^2 \quad (3.5)$$

where $\gamma_\ell, \gamma_u, \psi_\ell$ and ψ_u are positive constants, and do not scale with (s^*, n, d) . Existing literature has shown that (3.5) is satisfied by many common examples of sub-Gaussian random design with high probability [Raskutti, Wainwright and Yu (2010), Negahban et al. (2012)].

We then verify Assumptions 3.1 and 3.5 by the following lemma.

LEMMA 3.13. Suppose $\varepsilon \sim N(0, \sigma^2 I_n)$ and (3.5) holds. Given $\lambda_N = 8\sigma \times \sqrt{\log d/n}$, we have

$$\mathbb{P} \left(\lambda_N \geq 4 \|\nabla \mathcal{L}(\theta^*)\|_\infty = \frac{4}{n} \|X^\top \varepsilon\|_\infty \right) \geq 1 - 2d^{-2}$$

Moreover, given $\|\frac{1}{n} X^\top X\|_1 = \mathcal{O}(d)$, $\|\theta^*\|_\infty = \mathcal{O}(d)$, $\gamma \geq 4/\psi_\ell$, and large enough n , there exists a generic constant C_1 such that we have $N = \mathcal{O}_P(\log d)$,

$$\tilde{s} = C_1 s^* \geq [484\kappa^2 + 100\kappa] \cdot s^*$$

$$\tilde{\rho}_-(s^* + 2\tilde{s}) \geq \frac{\psi_\ell}{4} \quad \text{and}$$

$$\rho_+(s^* + 2\tilde{s}) \leq \frac{5\psi_u}{4}$$

Lemma 3.13 guarantees that the regularization sequence satisfies Assumption 3.1 with high probability, and Assumption 3.5 holds when the design matrix satisfies (3.5). Thus, by Theorem 3.12, we know that with high probability, PICASSO attains a linear convergence to a unique sparse local optimum for sparse linear regression. Moreover, Lemma 3.13 also implies that the number of regularization parameters only needs to be the order of $\log d$. Thus, solving the optimization problem with a sequence of regularization parameters does not require much additional efforts.

We then characterize the statistical rate of convergence in parameter estimation for the estimator obtained by PICASSO.

THEOREM 3.14 (Parameter estimation). Suppose $\varepsilon \sim N(0, \sigma^2 I_n)$ and (3.5) holds. Given $\gamma \geq 4/\psi_\ell$ and $\lambda_N = 8\sigma \sqrt{\log d/n}$, for small enough δ_N and large enough n such that $n \geq C_2 s^* \log d$ for a generic constant C_2 , we have

$$\|\hat{\theta}^{\{N\}} - \theta^*\|_2 = \mathcal{O}_P \left(\sigma \sqrt{\frac{s_1^*}{n}} + \sigma \sqrt{\frac{s_2^* \log d}{n}} \right)$$

where $s_1^* = |\{j \mid |\theta_j^*| \geq \gamma \lambda_N\}|$ and $s_2^* = |\{j \mid 0 < |\theta_j^*| < \gamma \lambda_N\}|$.

By dividing all nonzero θ_j^* 's into strong signals and weak signals by their magnitudes, Theorem 3.14 shows that the MCP regularizer reduces the estimation error for strong signal with magnitudes larger than $\gamma \lambda_N$ and, therefore, attains a faster statistical rate of convergence than Lasso.

REMARK 3.15 (Parameter estimation for Lasso). Theorem 3.14 is also applicable to Lasso with $\gamma = \infty$. As a result, all nonzero θ_j^* 's are considered as weak signals $|\theta_j^*| < \infty$ for all $j = 1, \dots, d$, that is, $s_1^* = 0$ and $s_2^* = s^*$. Theorem 3.14 only guarantees a slower statistical rate of convergence for Lasso,

$$\|\hat{\theta}^{\{N\}} - \theta^*\|_2 = \mathcal{O}_P \left(\sigma \sqrt{\frac{s_2^* \log d}{n}} \right) = \mathcal{O}_P \left(\sigma \sqrt{\frac{s^* \log d}{n}} \right) \quad \text{for } \gamma = \infty$$

We then proceed to show that the statistical rate of convergence in Theorem 3.14 is minimax optimal in parameter estimation for a suitably chosen $\gamma < \infty$. Particularly, we consider a class of sparse vectors:

$$\Theta(s_1^*, s_2^*, d) = \left\{ \theta^* \mid \theta^* \in \mathbb{R}^d, \sum_{j=1}^d \mathbb{1}_{\{|\theta_j^*| \geq \theta_{\min}\}} \leq s_1^*, \sum_{j=1}^d \mathbb{1}_{\{0 < |\theta_j^*| < \theta_{\min}\}} \leq s_2^* \right\} \quad (3.6)$$

where $\theta_{\min} = \frac{8\gamma\sigma}{\sqrt{C_2(s_1^* + s_2^*)}}$ is the threshold between strong and weak signals for some generic constant C_2 and $\gamma < \infty$. Given $s^* = s_1^* + s_2^*$ and $n \geq C_2 s^* \log d$, we have

$$\theta_{\min} = \frac{8\gamma\sigma}{\sqrt{C_2(s_1^* + s_2^*)}} \geq 8\gamma\sigma \sqrt{\frac{\log d}{n}} = \gamma\lambda_N$$

which matches the threshold for dividing signals in Theorem 3.14. The next theorem establishes a lower bound for parameter estimation.

THEOREM 3.16 (Lower bound). *Let $\hat{\theta}$ denote any estimator of θ^* based on $y \sim N(X\theta^*, \sigma^2 I_n)$, where $\theta^* \in \Theta(s_1^*, s_2^*, d)$. Then there exists a generic constant C_4 such that*

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta(s_1^*, s_2^*, d)} \mathbb{E} \|\hat{\theta} - \theta^*\|_2 \geq C_4 \left(\sigma \sqrt{\frac{s_1^*}{n}} + \sigma \sqrt{\frac{s_2^* \log d}{n}} \right)$$

Theorem 3.16 guarantees that the estimator obtained by PICASSO attains the minimax optimal rates of convergence over $\Theta(s_1^*, s_2^*, d)$. The convex ℓ_1 regularizer, however, only attains a suboptimal statistical rate of convergence due to the universal estimation bias regardless the signal strength. See more details in Zhang and Huang (2008), Bickel, Ritov and Tsybakov (2009).

To analyze the support recovery performance for the estimator obtained by PICASSO, we define the oracle least square estimator $\hat{\theta}^o$ as

$$\hat{\theta}_S^o = \underset{\theta_S}{\operatorname{argmin}} \frac{1}{2n} \|y - X_{*S} \theta_S\|_2^2 \quad \text{and} \quad \hat{\theta}_S^o = 0 \quad (3.7)$$

where S and \bar{S} are defined in (3.3). Recall $\bar{\theta}^{\lambda_N}$ is the unique sparse local minimizer to (1.1) with λ_N . The following theorem shows that $\bar{\theta}^{\lambda_N}$ is identical to the oracle least square estimator $\hat{\theta}^o$ with high probability.

THEOREM 3.17 (Support recovery). *Suppose (3.5) holds,*

$$\varepsilon \sim N(0, \sigma^2 I_n), \quad \text{and} \quad \min_{j \in S} |\theta_j^*| \geq C_5 \gamma \sigma \sqrt{\frac{\log d}{n}} \quad (3.8)$$

for a generic constant C_5 . Given $4/\psi_\ell \leq \gamma < \infty$ and $\lambda_N = 8\sigma \sqrt{\log d/n}$, for large enough n , there exists a generic constant C_3 such that $\mathbb{P}(\bar{\theta}^{\lambda_N} = \hat{\theta}^o) \geq 1 - 4d^{-2}$. Meanwhile, with probability at least $1 - 4d^{-2}$, we also have

$$\|\hat{\theta}^{\{N\}} - \theta^*\|_2 \leq C_3 \sigma \sqrt{\frac{s^*}{n}} \quad \text{and} \quad \operatorname{supp}(\hat{\theta}^{\{N\}}) = \operatorname{supp}(\theta^*)$$

Theorem 3.17 guarantees that PICASSO converges to $\hat{\theta}^o$ with high probability, which is often referred to the oracle property in existing literature [Fan and Li (2001)]. Besides, we also guarantee that the estimator $\hat{\theta}^{\{N\}}$ obtained by PICASSO is nearly unbiased and correctly identifies the true support with high probability. Although the ℓ_1 regularizer can be viewed as a special case of MCP, such an oracle property does not hold Lasso. This is because we require $\gamma < \infty$ such that the estimation bias can be eliminated for strong signals. Thus, Lasso cannot guarantee the correct support recovery [unless the design matrix satisfies a restrictive irrepresentable condition; see more details in Zhao and Yu (2006), Meinshausen and Bühlmann (2006), Zou (2006)]. We present an illustration of Theorems 3.14 and 3.17 in Figure 6.

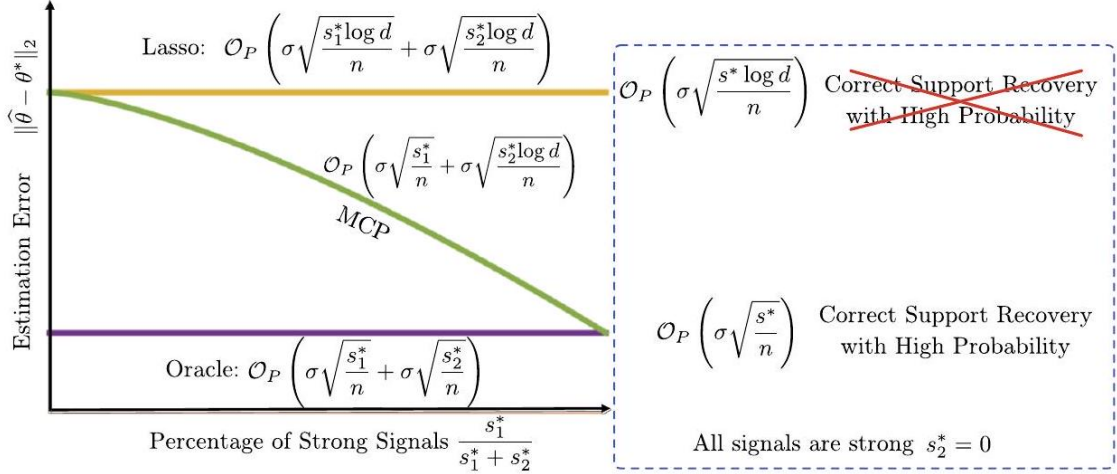


Figure 6. An illustration of the statistical rates of convergence in parameter estimation and support recovery for the Lasso, MCP and oracle estimators. Recall s_1^* and s_2^* are defined in (3.6), and $s^* = s_1^* + s_2^*$. When all the signals are weak ($s_1^* = 0, s^* = s_2^*$), both the Lasso and MCP estimators attain the same estimation error bound $\mathcal{O}_P\left(\sigma\sqrt{s^*\log d/n}\right)$. When some signals are strong, the MCP-regularized estimator attains a better estimation error bound $\mathcal{O}_P\left(\sigma\sqrt{s_1^*/n} + \sigma\sqrt{s_2^*\log d/n}\right)$ than Lasso, because it reduces the estimation bias for the strong signals. Eventually, when all the signals are strong ($s_2^* = 0, s^* = s_1^*$), the MCP estimator attains the same estimation error bound as the oracle estimator $\mathcal{O}_P\left(\sigma\sqrt{s^*/n}\right)$, and also correctly identify the support true support with high probability.

4 Extension to general loss functions

PICASSO can be further extended to other regularized M-estimation problems. Taking sparse logistic regression as an example,⁶ we denote the binary response vector by $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$, and the design matrix by $X \in \mathbb{R}^{n \times d}$. We consider a logistic model with $\mathbb{P}(y_i = 1) = \pi_i(\theta^*)$ and $\mathbb{P}(y_i = -1) = 1 - \pi_i(\theta^*)$, where

$$\pi_i(\theta) = \frac{1}{1 + \exp(-X_{i*}^\top \theta)} \quad \text{for } i = 1, \dots, n \quad (4.1)$$

When θ^* is sparse, we consider the optimization problem

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) + \mathcal{R}_\lambda(\theta) \quad \text{where } \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n \left[\log(1 + \exp(-y_i X_{i*}^\top \theta)) \right] \quad (4.2)$$

For notational simplicity, we denote the logistic loss function in (4.2) as $\mathcal{L}(\theta)$, and define $\tilde{\mathcal{L}}_\lambda(\theta) = \mathcal{L}(\theta) + \mathcal{H}_\lambda(\theta)$. Then similar to sparse linear regression, we write $\mathcal{F}_\lambda(\theta)$ as

$$\mathcal{F}_\lambda(\theta) = \mathcal{L}(\theta) + \mathcal{R}_\lambda(\theta) = \tilde{\mathcal{L}}_\lambda(\theta) + \lambda \|\theta\|_1$$

The logistic loss function is twice differentiable with

$$\nabla \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n [1 - \pi_i(\theta)] y_i X_{i*} \quad \text{and} \quad \nabla^2 \mathcal{L}(\theta) = \frac{1}{n} X^\top P X$$

⁶Due to space limit, we only present sparse logistic regression as an example. Please see more details on sparse robust regression using the huber loss function in the supplementary materials [Zhao, Liu and Zhang (2018)].

where $P = \text{diag}([1 - \pi_1(\theta)]\pi_1(\theta), \dots, [1 - \pi_n(\theta)]\pi_n(\theta)) \in \mathbb{R}^{n \times n}$. Similar to sparse linear regression, we also assume that the design matrix X satisfies the column normalization condition $\|X_{*j}\|_2 = \sqrt{n}$ for all $j = 1, \dots, d$.

4.1 Proximal coordinate gradient descent

For sparse logistic regression, directly taking the minimum with respect to a selected coordinate does not admit a closed-form solution and, therefore, may involve some sophisticated algorithm such as the root-finding method.

To address this issue, Razaviyayn, Hong and Luo (2013) suggest a more convenient approach, which takes a proximal coordinate gradient descent iteration. For example, we select a coordinate j at the t th iteration and consider a quadratic approximation of $\mathcal{F}_\lambda(\theta_j; \theta_j^{(t)})$,

$$\mathcal{Q}_{\lambda,j,L}(\theta_j; \theta_j^{(t)}) = \mathcal{V}_{\lambda,j,L}(\theta_j; \theta_j^{(t)}) + \lambda|\theta_j| + \lambda\|\theta_j^{(t)}\|_1$$

where $L > 0$ is a step size parameter, and $\mathcal{V}_{\lambda,j,L}(\theta_j; \theta_j^{(t)})$ is defined as

$$\mathcal{V}_{\lambda,j,L}(\theta_j; \theta_j^{(t)}) = \tilde{\mathcal{L}}_\lambda(\theta^{(t)}) + (\theta_j - \theta_j^{(t)})\nabla_j \tilde{\mathcal{L}}_\lambda(\theta^{(t)}) + \frac{L}{2}(\theta_j - \theta_j^{(t)})^2$$

Here, we choose the step size parameter L such that $\mathcal{Q}_{\lambda,j,L}(\theta_j; \theta_j^{(t)}) \geq \mathcal{F}_\lambda(\theta_j, \theta_j^{(t)})$ for all $j = 1, \dots, d$. We then take

$$\theta_j^{(t+1)} = \underset{\theta_j}{\operatorname{argmin}} \mathcal{Q}_{\lambda,j,L}(\theta_j; \theta_j^{(t)}) = \underset{\theta_j}{\operatorname{argmin}} \mathcal{V}_{\lambda,j,L}(\theta_j; \theta_j^{(t)}) + \lambda|\theta_j| \quad (4.3)$$

Different from the exact coordinate minimization, (4.3) always has a closedform solution obtained by soft thresholding. Particularly, we define $\tilde{\theta}_j^{(t)} = \theta_j^{(t)} - \nabla_j \tilde{\mathcal{L}}_\lambda(\theta^{(t)})/L$. Then we have

$$\theta_j^{(t+1)} = \underset{\theta_j}{\operatorname{argmin}} \frac{1}{2}(\theta_j - \tilde{\theta}_j^{(t)})^2 + \frac{\lambda}{L}|\theta_j| = \mathcal{S}_{\lambda/L}(\tilde{\theta}_j^{(t)}) \quad \text{and} \quad \theta_{\setminus j}^{(t+1)} = \theta_{\setminus j}^{(t)}$$

For notational convenience, we write $\theta_j^{(t+1)} = \mathcal{T}_{\lambda,j,L}(\theta^{(t)})$. When applying PICASSO to solve sparse logistic regression, we only need to replace $\mathcal{T}_{\lambda,j}(\cdot)$ with $\mathcal{T}_{\lambda,j,L}(\cdot)$ in Algorithms 1-3.

REMARK 4.1. For sparse logistic regression, we have $\nabla_{jj}^2 \mathcal{L}(\theta) = \frac{1}{n} X_{*j}^\top P X_{*j}$. Since P is a diagonal matrix, and $\pi_i(\theta) \in (0, 1)$ for any $\theta \in \mathbb{R}^d$, we have $\|P\|_2 = \max_i P_{ii} \in (0, 1/4]$ for all $i = 1, \dots, n$. Then we have $X_{*j}^\top P X_{*j} \leq \|P\|_2 \|X_{*j}\|_2^2 \leq n/4$, where the last inequality comes from the column normalization condition of X . Thus, we choose $L = \sup_\theta \max_j \nabla_{jj}^2 \mathcal{L}(\theta) = 1/4$.

We then analyze the computational and statistical properties of the estimator obtained by PICASSO for sparse logistic regression.

4.2 Convex relaxation based warm start initialization

We assume that $\|\theta^*\|_0 \leq s^*$, and for any $v \neq 0$ and any θ such that $\|\theta - \theta^*\|_2 \leq R$, we have

$$\psi_\ell \|v\|_2^2 - \gamma_\ell \sqrt{\frac{\log d}{n}} \|v\|_1^2 \leq v^\top \nabla^2 \mathcal{L}(\theta) v \leq \psi_u \|v\|_2^2 + \gamma_u \sqrt{\frac{\log d}{n}} \|v\|_1^2 \quad (4.4)$$

where $\gamma_\ell, \gamma_u, \psi_\ell, \psi_u$ and R are positive constants, and do not scale with (s^*, n, d) . Existing literature has shown that many common examples of sub-Gaussian random design satisfy (4.4) with high

probability [Raskutti, Wainwright and Yu (2010), Negahban et al. (2012), Loh and Wainwright (2015)].

Similar to sparse linear regression, we need to verify Assumptions 3.1 and 3.5 for sparse logistic regression by the following lemma.

LEMMA 4.2. *Suppose (4.4) holds. Given $\lambda_N = 16\sqrt{\log d/n}$, we have*

$$\mathbb{P}\left(\lambda_N \geq 4\|\nabla\mathcal{L}(\theta^*)\|_\infty = \frac{4}{n}\|X^\top w\|_\infty\right) \geq 1 - d^{-7}$$

where $w = ([1 - \pi_1(\theta^*)]y_1, \dots, [1 - \pi_n(\theta^*)]y_n)^\top$ with $\pi_i(\theta)$'s defined in (4.1). Moreover, given $\gamma \geq 4/\psi_\ell$ and $\|\theta - \theta^*\|_2 \leq R$, there exists some generic constant C_1 such that for large enough n , we have

$$\tilde{s} = C_1 s^* \geq [484\kappa^2 + 100\kappa] s^* \quad \tilde{\rho}_-(s^* + 2\tilde{s}) \geq \frac{\psi_\ell}{2} \quad \rho_+(s^* + 2\tilde{s}) \leq \frac{5\psi_u}{4}$$

The proof of Lemma 4.2 directly follows the proof of Lemma 3.13 and Loh and Wainwright (2015) and, therefore, is omitted. Lemma 4.2 guarantees that the regularization sequence satisfies Assumption 3.1 with high probability, and Assumption 3.5 holds when the design matrix satisfies (4.4).

Different from sparse linear regression, however, the restricted convexity and smoothness only hold over an ℓ_2 ball centered at θ^* for sparse logistic regression. Thus, directly choosing $\hat{\theta}^{(0)} = 0$ may violate the restricted strong convexity. A simple counterexample is $\|\theta^*\|_2 > R$, which results in $\|0 - \theta^*\|_2 > R$. To address this issue, we propose a new convex relaxation based warm start initialization to obtain an initial solution for λ_0 . Particularly, we solve the following convex relaxation of (1.1):

$$\min_{\theta \in \mathbb{R}^d} \tilde{\mathcal{F}}_{\lambda_0}(\theta) \quad \text{where } \tilde{\mathcal{F}}_{\lambda_0}(\theta) = \mathcal{L}(\theta) + \lambda_0 \|\theta\|_1 \quad (4.5)$$

up to an adequate precision. For example, we choose θ^{relax} satisfying the approximate KKT condition of (4.5) as follows:

$$\min_{\xi \in \partial \|\theta^{\text{relax}}\|_1} \|\nabla \mathcal{L}(\theta^{\text{relax}}) + \lambda_0 \xi\|_\infty \leq \delta_0 \lambda_0 \quad (4.6)$$

where $\delta_0 \in (0, 1)$ is the initial precision parameter for λ_0 . Since δ_0 in (4.6) can be chosen as a sufficiently large value (e.g., $\delta_0 = 1/8$), computing θ^{relax} becomes very efficient even for algorithms with only sublinear rates of convergence to global optima, for example, classical coordinate minimization and proximal gradient algorithms. For notational convenience, we call the above initialization procedure the convex relaxation based warm initialization.

LEMMA 4.3. *Suppose Assumption 3.5 holds only for $\|\theta - \theta^*\|_2 \leq R$. Given $\rho_-(s^* + \tilde{s})R \geq 9\lambda_0\sqrt{s^*} \geq 18\lambda_N\sqrt{s^*}$ and $\delta_0 = 1/8$, we have*

$$\|\theta_{\tilde{S}}^{\text{relax}}\|_0 \leq \tilde{s} \quad \|\theta^{\text{relax}} - \theta^*\|_2 \leq R \quad \text{and} \quad \mathcal{F}_{\lambda_0}(\theta^{\text{relax}}) \leq \mathcal{F}_{\lambda_0}(\theta^*) + \Delta_{\lambda_0}$$

Lemma 4.3 guarantees that θ^{relax} is a proper initial solution for λ_0 . Thus, all convergence analysis in Theorem 3.12 directly follows, and PICASSO attains a linear convergence to a unique sparse local optimum with high probability. The statistical properties can also be established accordingly. An illustration of the convex relaxation based warm start initialization is provided in Figure 7.

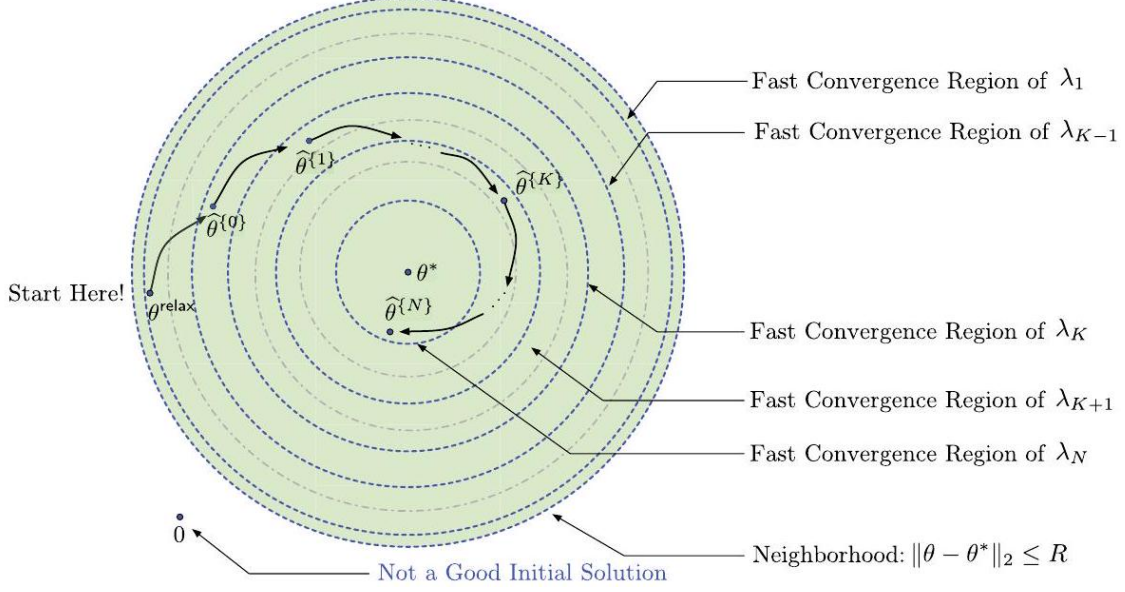


Figure 7. An illustration of the convex relaxation based warm start initialization. When the restricted convexity and smoothness only hold over a neighborhood around θ^* (green region). Directly choosing 0 as the initial solution may violate the restricted strong convexity. Thus, we adopt a convex relaxation approach to obtain an initial solution, which is ensured to be sparse and belong to the desired neighborhood.

5 Numerical experiments

We evaluate the computational and statistical performance of PICASSO through numerical simulations. We compare PICASSO with five competitors: (1) SparseNet [Mazumder, Friedman and Hastie (2011)]; (2) Path-following Iterative Shrinkage Thresholding Algorithm [PISTA, Wang, Liu and Zhang (2014)]; (3) Accelerated PISTA [A-PISTA, Zhao and Liu (2016)]; (4) Multistage Convex Relaxation Method [Mcvx, Zhang (2013)]; (5) Local Linear Approximation [LLA, Zou and Li (2008)]. Note that each subproblem of Mcvx and LLA is solved by proximal gradient algorithms with backtracking line search.

All experiments are conducted on a PC with Intel Core i5 3.3 GHz and 16 GB memory. All programs are coded in double precision C, called from an R wrapper. We optimize the computation by exploiting the vector and matrix sparsity, which gains a significant speedup in vector and matrix manipulations (e.g., computing the gradient and evaluating the objective value). We apply PICASSO to sparse linear regression with the MCP regularizer.

We generate each row of the design matrix X_{i*} independently from a d dimensional Gaussian distribution with mean 0 and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$, where $\Sigma_{kj} = 0.75$ and $\Sigma_{kk} = 1$ for all $j, k = 1, \dots, d$ and $k \neq j$. We then normalize each column of the design matrix X_{*j} such that $\|X_{*j}\|_2 = \sqrt{n}$. The response vector is generated from the linear model $y = X\theta^* + \varepsilon$, where $\theta^* \in \mathbb{R}^d$ is the regression coefficient vector, and ε is generated from a n -dimensional Gaussian distribution with mean 0 and covariance matrix $\sigma^2 I_n$. We set $n = 300, d = 18,000, s^* = 18$ and $\sigma^2 = 4$. θ^* has 18 nonzero entries, which are $\theta_{1000}^* = \theta_{7000}^* = \theta_{13,000}^* = 3, \theta_{2000}^* = \theta_{8000}^* = \theta_{14,000}^* = 2, \theta_{3000}^* = \theta_{9000}^* = \theta_{15,000}^* = 1.5, \theta_{4000}^* = \theta_{10,000}^* = \theta_{16,000}^* = -3, \theta_{5000}^* = \theta_{11,000}^* = \theta_{17,000}^* = -2$ and $\theta_{6000}^* = \theta_{12,000}^* = \theta_{18,000}^* = -1.5$ for $k = 0, \dots, 2$. We then set $\gamma = 1.25, N = 70, \lambda_N = 0.25\sigma\sqrt{\log d/n}, \varphi = 0.05, \delta_K = 10^{-3}$ and $\tau_K = 10^{-6}$ for all $1 \leq K \leq N$.

Method	$\ \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\ _2$	$\ \hat{\boldsymbol{\theta}}_{\mathcal{S}}\ _0$	$\ \hat{\boldsymbol{\theta}}_{\mathcal{S}^c}\ _0$	Correct	Timing
PICASSO	1.258(0.515)	17.79(0.54)	0.48(0.52)	616/1000	1.062(0.084)
SparseNet	1.602(0.791)	17.64(0.85)	2.07(1.41)	248/1000	1.109(0.088)
PISTA	1.267(0.528)	17.76(0.54)	0.55(0.51)	614/1000	52.358(5.920)
A-PISTA	1.276(0.530)	17.76(0.54)	0.57(0.57)	613/1000	6.358(0.865)
Mcvx	1.293(0.529)	17.76(0.52)	0.58(0.52)	615/1000	67.247(7.128)
LLA	1.517(0.949)	17.50(0.61)	1.28(0.85)	365/1000	31.247(3.870)

Table 1. Quantitative comparison on the simulated data set ($n = 300, d = 18,000, s^* = 18, \sigma^2 = 4$). In terms of timing performance, PICASSO slightly outperforms SparseNet, outperforms A-PISTA and greatly outperforms PISTA, LLA and Mcvx. In terms of support recovery and parameter estimation, PICASSO slightly outperforms A-PISTA, PISTA and Mcvx, and greatly outperforms SparseNet and LLA

We present the numerical results averaged over 1000 simulations. Specifically, we create a validation set using the same design matrix as the training set for regularization parameter selection. We then tune the regularization parameter over the selected regularization sequence. We denote the response vector of the validation set as $\tilde{\mathbf{y}} \in \mathbb{R}^n$. Let $\hat{\boldsymbol{\theta}}^\lambda$ denote the obtained estimator using the regularization parameter λ . We then choose the optimal regularization parameter $\hat{\lambda}$ by

$$\hat{\lambda} = \underset{\lambda \in \{\lambda_1, \dots, \lambda_N\}}{\operatorname{argmin}} \quad \|\tilde{\mathbf{y}} - X\hat{\boldsymbol{\theta}}^\lambda\|_2^2$$

We repeat the simulation for 1000 times and summarize the averaged results in Table 1. In terms of timing performance, PICASSO slightly outperforms SparseNet, outperforms A-PISTA, and greatly outperforms PISTA, LLA and Mcvx. In terms of support recovery and parameter estimation, PICASSO slightly outperforms A-PISTA, PISTA and Mcvx, and greatly outperforms SparseNet and LLA.

To further demonstrate the superiority of PICASSO, we present a typical failure example of SparseNet using the heuristic cyclic selection rule. This example is chosen from our 1000 simulations, and illustrated in Figure 8. We see that the heuristic cyclic selection rule in SparseNet always needs to iterate over many irrelevant variables before getting to the relevant variable when identifying a new active set. Since these irrelevant variables are highly correlated with the relevant variables in our experiment, the heuristic cyclic selection rule tends to overselect the irrelevant variables and miss some relevant variables. In contrast, PICASSO, PISTA and A-PISTA have mechanisms to prevent from overselecting the irrelevant coordinates when identifying active sets. This eventually makes them outperform SparseNet in both parameter estimation and support recovery. Moreover, we also see that PISTA is much slower than other algorithms, because PISTA needs to calculate a full gradient and conduct a sophisticated line search in every iteration, which are computationally expensive. Though A-PISTA adopts the coordinate minimization to further accelerate PISTA, it still suffers from the computationally expensive line search when identifying active sets. This eventually leads to less competitive timing performance than PICASSO.

6 Discussions and future work

Here, we discuss several existing methods related to PICASSO, including the multistage convex relaxation method (Mcvx), local linear approximation method (LLA), path-following iterative shrinkage thresholding algorithm (PISTA), accelerated path-following iterative shrinkage thresholding algorithm (A-PISTA) and proximal gradient algorithm.

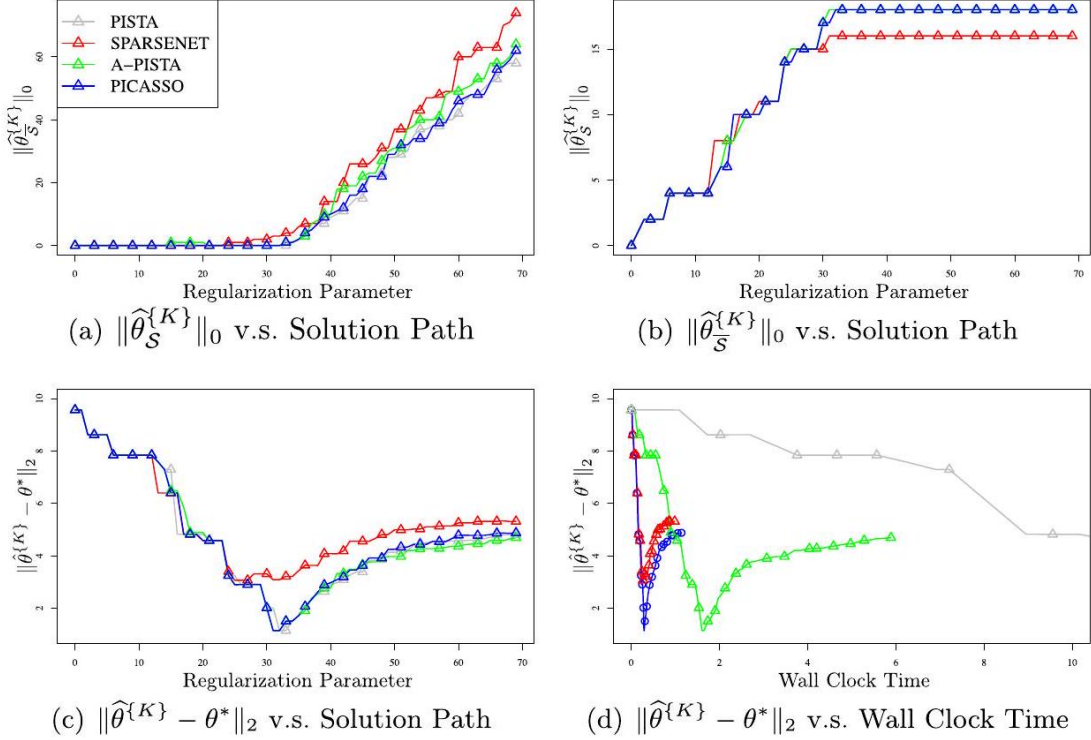


Figure 8. A typical failure example of SparseNet using the heuristic cyclic selection rule, which is chosen from our 1000 simulations. We see that cyclic selection rule tends to overselect the irrelevant coordinates and miss some relevant coordinates when updating the active set. Thus, SparseNet eventually yields denser solutions with worse performance in parameter estimation and support recovery than PICASSO, PISTA and A-PISTA.

The multistage convex relaxation method is proposed in Zhang (2013). It solves a sequence of convex relaxation problems of (1.1). Zhang (2013) show that the obtained estimator enjoys similar statistical guarantees to those of PICASSO for sparse linear regression. However, there is only sublinear guarantee on its convergence rate to a local optimum. Moreover, since each relaxed problem is still lack of strong convexity, the multistage convex relaxation method needs to be combined with some efficient computational algorithms such as PICASSO.

The local linear approximation method is proposed in Zou and Li (2008), Wang, Kim and Li (2013), Fan, Xue and Zou (2014). It is essentially a special case of the multistage convex relaxation with only two iterations. Similar to the multistage convex relaxation method, it also needs an efficient computational algorithm to solve each relaxed problem. Moreover, in order to obtain the variable selection consistency, the local linear approximation method requires a stronger minimum signal strength. Taking sparse linear regression as an example, Wang, Kim and Li (2013), Fan, Xue and Zou (2014) requires a minimum signal strength of order of $\sigma\sqrt{s^*\log d/n}$, while PICASSO only requires a minimum signal strength of order of $\sigma\sqrt{\log d/n}$.

The path-following iterative shrinkage thresholding algorithm (PISTA) is proposed in Wang, Liu and Zhang (2014). PISTA is essentially a proximal gradient algorithm combined with the warm start initialization. PISTA needs to calculate the entire (d -dimensional) gradient vector and requires a sophisticated backtracking line search procedure in every iteration. Thus, PICASSO is computationally much more efficient and scalable than PISTA in practice, although PISTA and

PICASSO enjoy similar theoretical guarantees. Besides, the implementation of PISTA requires subtle control over the step size, and often yield slow empirical convergence. An accelerated PISTA algorithm (A-PISTA) is proposed in Zhao and Liu (2016), which uses coordinate minimization algorithms to accelerated PISTA. It shows an improved computational performance over PISTA in our numerical simulations, but not as competitive as PICASSO.

Moreover, when extending PISTA to general loss functions, Wang, Liu and Zhang (2014) propose a contained formulation. Particularly, they solve (1.1) with an additional constraint,

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) + \mathcal{R}_\lambda(\theta) \quad \text{subject to } \|\theta\|_2 \leq R/2 \quad (6.1)$$

The additional constraint guarantees that the solution always stays in the restricted strongly convex region (a small neighborhood around θ^*), only under the assumption $\|\theta^*\|_2 \leq R/2$, where R is a constant and cannot scale with (n, s^*, d) . This assumption is very restrictive, and also introduces an additional tuning parameter. In contrast, our proposed convex relaxation based warm start initialization avoids this assumption, and allows $\|\theta^*\|_2$ to be arbitrarily large. Furthermore, we want to emphasize that PISTA exploits an explicit soft-thresholding procedure to directly control the solution sparsity in each iteration, while PICASSO adopts an algorithmic strategy to control the sparsity of the active set.

Other researchers focus on solving (1.1) with an additional constraint,

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) + \mathcal{R}_\lambda(\theta) \quad \text{subject to } \|\theta\|_1 \leq M \quad (6.2)$$

where $M > 0$ is an extra tuning parameter. Loh and Wainwright (2015) show that the proximal gradient algorithm attains a linear convergence to a ball centered at θ^* to (6.2) with a radius approximately equal to the statistical error. However, the analysis of Loh and Wainwright (2015) does not justify the advantage of nonconvex regularization: They only provide a slower statistical rate of convergence than PICASSO in parameter estimation for their obtained estimator, and no support recovery guarantee is established. Besides, their analysis for general loss functions also requires the restrictive assumption: $\|\theta^*\|_2 \leq R/2$, where R is a constant and does not scale with (n, s^*, d) . Nevertheless, PICASSO does not require this assumption.

For future work, we are interested in several possible extensions: (I) Extension to more complicated regularizers such as grouping regularizers for variable clustering; (II) Extension to more complicated (possibly nonconvex) loss functions such as sparse phase retrieval and sparse coding problems; (III) Extension to asynchronous parallel optimization setting with shared memory or communication-efficient distributed optimization setting; (IV) Extension to second-order algorithms such as the regularized iterative reweighed least square optimization algorithm for sparse generalized linear model estimation (proximal Newton). These extensions will lead to more efficient and scalable coordinate optimization algorithms for more sophisticated nonconvex optimization problems.

7 Conclusion

The pathwise coordinate optimization framework is a critical advancement for high-dimensional convex and nonconvex sparse learning problems. It introduces three key innovations—warm start initialization, active set updates, and a strong rule for coordinate preselection—that distinguish it from classical coordinate optimization methods. These features significantly enhance its empirical performance, yet they present challenges for rigorous theoretical analysis.

In this paper, we developed a comprehensive theory that clarifies how these components contribute to both the computational and statistical performance of the framework. We demonstrated that they ensure linear convergence to a unique sparse local optimum, with strong statistical guarantees for parameter estimation and support recovery. Additionally, we introduced several modifications that further refine the pathwise optimization process, offering the first complete set of computational and statistical guarantees for high-dimensional optimization.

Our extensive numerical experiments validate the theoretical results, highlighting the robustness and efficiency of the proposed framework. This work successfully bridges the gap between theory and practice, offering a solid foundation for future advancements and applications of pathwise optimization techniques in various high-dimensional data settings.

References

- [Bickel et al.(2009)] Bickel, P. J., Ritov, Y., & Tsybakov, A. B. (2009). Simultaneous analysis of lasso and Dantzig selector. *Annals of Statistics*, 37, 1705–1732.
- [Duchi(2015)] Duchi, J. (2015). Lecture notes for statistics and information theory. Retrieved from http://stanford.edu/class/stats311/Lectures/full_notes.pdf
- [Eloyan et al.(2012)] Eloyan, A., Muschelli, J., Nebel, M. B., Liu, H., Han, F., Zhao, T., Barber, A. D., Joel, S., Pekar, J. J., Mostofsky, S. H., & Caffo, B. (2012). Automated diagnoses of attention deficit hyperactive disorder using magnetic resonance imaging. *Frontiers in Systems Neuroscience*, 6, 61.
- [Fan & Li(2001)] Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96, 1348–1360.
- [Fan et al.(2014)] Fan, J., Xue, L., & Zou, H. (2014). Strong oracle optimality of folded concave penalized estimation. *Annals of Statistics*, 42, 819–849.
- [Friedman et al.(2007)] Friedman, J., Hastie, T., Höfling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, 1, 302–332.
- [Fu(1998)] Fu, W. J. (1998). Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7, 397–416.
- [Hastie(2009)] Hastie, T. (2009). Fast regularization paths via coordinate descent. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [Hastie & Tibshirani(1990)] Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models*. Monographs on Statistics and Applied Probability, 43. Chapman & Hall.
- [Li et al.(2016)] Li, X., Zhao, T., Arora, R., Liu, H., & Hong, M. (2016). An improved convergence analysis of cyclic block coordinate descent-type methods for strongly convex minimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*.
- [Liu et al.(2015)] Liu, H., Wang, L., & Zhao, T. (2015). Calibrated multivariate regression with application to neural semantic basis discovery. *Journal of Machine Learning Research*, 16, 1579–1606.
- [Loh & Wainwright(2015)] Loh, P.-L., & Wainwright, M. J. (2015). Regularized M -estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16, 559–616.
- [Lu & Xiao(2015)] Lu, Z., & Xiao, L. (2015). On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152, 615–642.
- [Luo & Tseng(1992)] Luo, Z. Q., & Tseng, P. (1992). On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72, 7–35.

- [Mazumder et al.(2011)] Mazumder, R., Friedman, J. H., & Hastie, T. (2011). SparseNet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106, 1125–1138.
- [Meinshausen & Bühlmann(2006)] Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34, 1436–1462.
- [Neale et al.(2012)] Neale, B. M., Kou, Y., Liu, L., Ma’Ayan, A., Samocha, K. E., Sabo, A., Lin, C.-F., Stevens, C., Wang, L.-S., Makarov, V., et al. (2012). Patterns and rates of exonic de novo mutations in autism spectrum disorders. *Nature*, 485, 242–245.
- [Negahban et al.(2012)] Negahban, S. N., Ravikumar, P., Wainwright, M. J., & Yu, B. (2012). A unified framework for high-dimensional analysis of M -estimators with decomposable regularizers. *Statistical Science*, 27, 538–557.
- [Nesterov(2013)] Nesterov, Yu. (2013). Gradient methods for minimizing composite functions. *Mathematical Programming*, 140, 125–161.
- [Raskutti et al.(2010)] Raskutti, G., Wainwright, M. J., & Yu, B. (2010). Restricted eigenvalue properties for correlated Gaussian designs. *Journal of Machine Learning Research*, 11, 2241–2259.
- [Raskutti et al.(2011)] Raskutti, G., Wainwright, M. J., & Yu, B. (2011). Minimax rates of estimation for high-dimensional linear regression over ℓ_q -balls. *IEEE Transactions on Information Theory*, 57, 6976–6994.
- [Razaviyayn et al.(2013)] Razaviyayn, M., Hong, M., & Luo, Z.-Q. (2013). A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23, 1126–1153.
- [Richtárik & Takáč(2014)] Richtárik, P., & Takáč, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144, 1–38.
- [Shalev-Shwartz & Tewari(2011)] Shalev-Shwartz, S., & Tewari, A. (2011). Stochastic methods for ℓ_1 -regularized loss minimization. *Journal of Machine Learning Research*, 12, 1865–1892.
- [Tibshirani(1996)] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58, 267–288.
- [Tibshirani et al.(2012)] Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., & Tibshirani, R. J. (2012). Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B*, 74, 245–266.
- [Wang et al.(2013)] Wang, L., Kim, Y., & Li, R. (2013). Calibrating nonconvex penalized regression in ultra-high
- [Wang et al.(2014)] Wang, Z., Liu, H., & Zhang, T. (2014). Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *Annals of Statistics*, 42, 2164–2201.
- [Zhang(2010)] Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38, 894–942.
- [Zhang(2013)] Zhang, T. (2013). Multi-stage convex relaxation for feature selection. *Bernoulli*, 19, 2277–2293.
- [Zhang & Huang(2008)] Zhang, C.-H., & Huang, J. (2008). The sparsity and bias of the LASSO selection in high-dimensional linear regression. *Annals of Statistics*, 36, 1567–1594.
- [Zhao & Liu(2016)] Zhao, T., & Liu, H. (2016). Accelerated path-following iterative shrinkage thresholding algorithm with application to semiparametric graph estimation. *Journal of Computational and Graphical Statistics*, 25, 1272–1296.

- [Zhao et al.(2018)] Zhao, T., Liu, H., & Zhang, T. (2018). Supplement to "Pathwise coordinate optimization for sparse learning: Algorithm and theory." DOI:10.1214/17-AOS1547SUPP.
- [Zhao & Yu(2006)] Zhao, P., & Yu, B. (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7, 2541–2563.
- [Zhao et al.(2012)] Zhao, T., Liu, H., Roeder, K., Lafferty, J., & Wasserman, L. (2012). The huge package for high-dimensional undirected graph estimation in R. *Journal of Machine Learning Research*, 13, 1059–1062.
- [Zou(2006)] Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101, 1418–1429.
- [Zou & Li(2008)] Zou, H., & Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, 36, 1509–1533.