J. Nonlinear Var. Anal. 8 (2024), No. 3, pp. 433-449
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.8.2024.3.06

# A FAST AND EFFECTIVE ALGORITHM FOR SPARSE LINEAR REGRESSION WITH $\ell_p$-NORM DATA FIDELITY AND ELASTIC NET REGULARIZATION

YUNHAI XIAO[1,2], JIAN SHEN[1], YANYUN DING[3], MENGJIAO SHI[1], PEILI LI[2,*]

[1]*School of Mathematics and Statistics, Henan University, Kaifeng 475000, China*
[2]*Center for Applied Mathematics of Henan Province, Henan University, Kaifeng 475000, China*
[3]*Institute of Applied Mathematics, Shenzhen Polytechnic University, Shenzhen 518055, China*

**Abstract.** Elastic net model is widely used in high-dimensional statistics for parameter regression and variable selection, which has been proved that the performance is often better than the lasso. However, it can only deals with data containing Gaussian noise, so it is not suitable for modern complex high-dimensional data. Fortunately, an adaptive and robust minimization model, which combines the $\ell_p$-norm data fidelity and elastic net regularization, has been proposed to deal with different types of noises and inherit the advantages of the elastic net in prediction accuracy. The double non-smoothness in objective function makes it challenging to minimize the model. After investigation, we find that the optimization algorithm is currently limited to the first-order alternating direction method of multipliers (ADMM), which is relatively lower in the recovered solutions' accuracy and relatively slower in the calculation speed. Therefore, we are committed to developing a fast and effective algorithm based on second-order information. Specifically, we propose a preconditioned proximal point algorithm (abbreviated as P-PPA) to solve the considered model by adding a proximal term. In theory, we analyze the consistency between the solution of the surrogate model and the original model. In addition, a key subproblem in P-PPA is solved by superlinear or even quadratically convergent semismooth Newton methods from the dual perspective. Finally, a large number of numerical experiments on high-dimensional simulated and real examples fully verify that our proposed algorithm is superior to ADMM in terms of calculation accuracy and speed.

**Keywords.** Alternating direction method of multipliers; High-dimensional sparse linear regression; $\ell_p$-$\ell_1$-$\ell_2$ minimization; Preconditioned proximal point algorithm; Semismooth Newton method.

## 1. INTRODUCTION

Let $x^* \in \mathbb{R}^n$ be a compressible sparse signal under a suitable basis (e.g. Fourier or wavelet basis). The main idea of compressive sensing (CS) is to firstly encode $x^*$ via a sensing matrix $A$, i.e, $Ax^* = b$ with $b \in \mathbb{R}^m$ and $m \ll n$, and then decode $x^*$ from the undersampled data $b$ by finding the sparest solution of an underdetermined linear system $Ax = b$. However, during the encoding and decoding process, undersampled data $b$ may be inevitably corrupted by various

*Corresponding author.
E-mail address: yhxiao@henu.edu.cn (Y. Xiao), d.shenjian@henu.edu.cn (J. Shen), dingyanyun@szpu.edu.cn (Y. Ding), smj@henu.edu.cn (M. Shi), lipeili@henu.edu.cn (P. Li)

Received 2 November 2023; Accepted 28 February 2024; Published online 5 April 2024.

types of noise. In this case, recovering $x^*$ is usually described as finding a solution of the $\ell_1$-norm regularized least square; see, e.g., [5, 12, 13],

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2+\lambda\|x\|_1, \tag{1.1}$$

where $\|\cdot\|_1$ is a $\ell_1$-norm function, and $\lambda>0$ is a weighting parameter to balance both terms for minimization. A deterministic result shows that it is possible to recover the original signal precisely by $\ell_1$-norm minimization when the number of nonzeros in $x^*$ is less than $(1+1/\zeta)/2$; see [13, 14], where $\zeta$ is the so-called mutual coherence of matrix $A$.

In the field of statistics, although the $\ell_1$-norm regularization (a.k.s. lasso penalty) has shown its success in many situations for sparse linear regression, it also has some limitations. For example, in the high-dimension case, i.e., $n \gg m$, lasso can select at most $m$ variables before saturation. And the well-posedness of lasso needs that the bound on the $\ell_1$-norm of the coefficients is smaller than a certain value. In addition, lasso tends to select only one variable from a group of highly correlated variables. As a result, Zou and Hastie [37] proposed a novel penalty, called elastic net, which can simultaneously does automatic variable selection and continuous shrinkage, and it also can select groups of correlated variables. If the elastic net is used to replace the $\ell_1$-norm in (1.1), we can obtain the following $\ell_1$-$\ell_2$-norm regularized minimization problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2+\lambda\left(\|x\|_1+\frac{\beta}{2}\|x\|_2^2\right), \tag{1.2}$$

where $\beta\geq 0$ is a tuning parameter. This model has been extensively used in many fields, such as neuroimaging [4], genome-wide association studies of rheumatoid arthritis [6], uncovering consistent networks of functional disconnection in Alzheimer's disease [31], and estimating global bank network connectedness [9].

The appearance of the $\|Ax-b\|_2^2$ makes (1.2) strongly convex. In addition, the $\|x\|_2^2$ makes it less sensitive to the contained noise than (1.1). Nevertheless, the quality of the resolutions of (1.2) also relies on the knowing of the standard deviation of the noise. To resolve this issue, Belloni et al. [3] recommended a square-root-loss $\|Ax-b\|_2$, which was proved to be not knowing the standard deviation and having the minimax optimal rate of convergence under some suitable conditions [1, 8]. On the other hand, observations in the era of modern big data are inevitably affected by various noises, such as heavy-tailed noise, Gaussian noise, uniformly distributed noise, and so on. When encountering heavy-tailed or heterogeneous noises, the data fidelity of $\|Ax-b\|_1$ is robust [2, 22, 32, 33]. In the case of uniformly distributed and quantization error, the data fidelity of $\|Ax-b\|_\infty$ is more suitable [34, 36]. Recently, based on $\ell_p$ $(p=1,2,\infty)$ norm data fidelity and elastic net regularization, Ding et al. [10] proposed a flexible and robust reconstruction model:

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_p+\lambda\left(\|x\|_1+\frac{\beta}{2}\|x\|_2^2\right), \tag{1.3}$$

where $\|\cdot\|_p$ is an $\ell_p$-norm function whose proximal mapping is assumed to be strongly semismooth. Obviously, $p=1,2,\infty$ meets the requirements of model [21, Remark 2]. Model (1.3) has a lot of nice features. For instance, it not only has the ability to produce sparse resolutions, but also can cope with different types of noise if $p$ is chosen dynamically, such as, $p=2$ for Gaussian noise, $p=1$ for log-normal noise and heavy-tailed noise, and $p=\infty$ for uniformly distributed noise. Despite all these advantages, comparing with (1.2), it is more difficult to find

a solution to (1.3) with higher accuracy because of the variability of the $\ell_p$-norm data fidelity term. Ding et al. [10] used the popular alternating direction method of multipliers (ADMM), which is easy to implement, but it is not enough to derive higher precision solutions quickly. Then naturally there is a question: can one propose a fast and effective algorithm to implement it?

The main contribution of this paper is to propose a fast and effective algorithm based on second-order information, which can fully exploit the structures of the $\ell_p$-norm. Specifically, we propose a preconditioned proximal point algorithm (P-PPA) which is inspired by the proximal majorization-minimization algorithm (PMM) of Tang et al. [30] for a square-root regression problem. Similarly, for the need of efficient calculation, we focus on the same model but add a proximal term $\frac{\sigma}{2}\|x-\tilde{x}\|^2+\frac{\tau}{2}\|Ax-A\tilde{x}\|^2$, where $\sigma>0$, $\tau>0$, and $\tilde{x}$ are known quantities in advance. In P-PPA, we employ a semismooth Newton method (SSN) to solve the key subproblem from the perspective of duality so that a superlinearly convergence is achieved. Finally, we implement the proposed algorithm by using a large number of simulation and real data which shows that the SSN based P-PPA performs better than ADMM in sense of recovery qualities and calculation speed.

The remaining parts of this paper are organized as follows. In Section 2, we summarize some basic definitions for subsequent algorithm design and numerical implementations. In Section 3, we propose an SSN based the P-PPA algorithm to improve the performance of the algorithm. Besides, the convergence result for the proposed algorithm is also included in each section. Then, in Section 4, we present some numerical experiments as well as some performance comparisons. Finally, we conclude this paper in Section 5.

## 2. PRELIMINARIES

Let $\mathbb{R}^n$ denote the $n$-dimensional Euclidean space, $\langle\cdot,\cdot\rangle$ denote the standard inner product. Let $f:\mathbb{R}^n\to(-\infty,+\infty]$ be a proper, closed, and convex function. We use $\mathrm{dom}(f)$ to denote the domain of $f$, that is, $\mathrm{dom}(f)=\{x\in\mathbb{R}^n\mid f(x)<\infty\}$. A vector $z$ is said to be a subgradient of $f$ at point $x$ if $f(y)\geq f(x)+\langle z,y-x\rangle$ for all $y\in\mathbb{R}^n$. The set of all subgradients of $f$ at $x$ is called the subdifferential of $f$ at $x$ and is denoted by $\partial f(x)$. Obviously, $\partial f(x)$ is a convex and closed set while it is not empty. The Fenchel conjugate of $f$ is defined as $f^*(z):=\sup_{x\in\mathbb{R}^n}\{\langle x,z\rangle-f(x)\}$. The Moreau envelope function of $f$ with parameter $t>0$, denoted by $\Phi_{tf}(x)$, is defined as [24, 35]

$$\Phi_{tf}(x):=\min_{y\in\mathbb{R}^n}\{f(y)+\frac{1}{2t}\|y-x\|_2^2\}. \tag{2.1}$$

The proximal mapping of $f$ with $t>0$ is defined by

$$\mathrm{Prox}_{tf}(x):=\operatorname*{argmin}_{y\in\mathbb{R}^n}\{f(y)+\frac{1}{2t}\|y-x\|_2^2\}. \tag{2.2}$$

From [15, 17], we know that $\Phi_{tf}(x)$ is continuously differentiable and convex with gradient in the form of

$$\nabla\Phi_{tf}(x)=t^{-1}(x-\mathrm{Prox}_{tf}(x)),\quad \forall x\in\mathbb{R}^n. \tag{2.3}$$

The following Moreau's identity [28, Theorem 35.1] is essential in the subsequent analysis:

$$\mathrm{Prox}_{tf}(x)+t\mathrm{Prox}_{f^*/t}(x/t)=x. \tag{2.4}$$

Next, we state some basic concepts and definitions, which are required at the subsequent arithmetic developments and numerical implementations.

**Definition 2.1.** (Semismoothness [23, 25]). Let $\Psi : \mathscr{O} \subseteq \mathbb{R}^n \to \mathbb{R}^m$ be a locally Lipschitz continuous function and $\mathscr{K} : \mathscr{O} \rightrightarrows \mathbb{R}^{m\times n}$ be a nonempty, compact valued, and upper-semicontinuous set-valued mapping on the open set $\mathscr{O}$. $\Psi$ is said to be semismooth at $\nu \in \mathscr{O}$ with respect to the set-valued mapping $\mathscr{K}$ if $\Psi$ is directionally differentiable at $\nu$ and for any $\Gamma \in \mathscr{K}(\nu+\Delta\nu)$ with $\Delta\nu \to 0$ such that

$$\Psi(\nu+\Delta\nu) - \Psi(\nu) - \Gamma\Delta\nu = o(\|\Delta\nu\|_2).$$

$\Psi$ is said to be $\gamma$-order $(\gamma > 0)$ (strongly, if $\gamma = 1$ ) semismooth at $\nu$ with respect to $\mathscr{K}$ if $\Psi$ is semismooth at $\nu$ and for any $\Gamma \in \mathscr{K}(\nu+\Delta\nu)$ such that

$$\Psi(\nu+\Delta\nu) - \Psi(\nu) - \Gamma\Delta\nu = O(\|\Delta\nu\|_2^{1+\gamma}).$$

$\Psi$ is called a semismooth ($\gamma$-order semismooth, strongly semismooth) function on $\mathscr{O}$ with respect to $\mathscr{K}$ if it is semismooth ($\gamma$-order semismooth, strongly semismooth) at every $\nu \in \mathscr{O}$ with respect to $\mathscr{K}$.

We now quickly review some preliminary results on the P-PPA. For more results, one may refer to the popular papers of Rockafellar [26, 27]. The P-PPA is a generalization of the PPA which was first studied by Li et al. [20]. Consider a closed proper convex function $f : \mathbb{R}^n \to (-\infty, +\infty]$. Give a self-adjoint positive definite matrices sequence $\{\mathscr{M}_k\}$, a nonnegative summable sequence $\{\rho_k\}$, and two positive numbers $0 < \xi_{\min} \le \xi_\infty < +\infty$, such that

$$(1+\rho_k)\mathscr{M}_k \succeq \mathscr{M}_{k+1}, \quad \mathscr{M}_k \succeq \xi_{\min} I_n, \forall k \ge 0, \quad \limsup_{k\to\infty} \lambda_{\max}(\mathscr{M}_k) = \xi_\infty,$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of a matrix. Starting from $x^0 \in \mathbb{R}^n$, the P-PPA generates an approximated sequence $\{x^k\}$ via the following scheme

$$x^{k+1} \approx \bar{x}^{k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^n} \left\{ f(x) + \frac{1}{\sigma_k} \left\| x - x^k \right\|_{\mathscr{M}_k}^2 \right\}, \tag{2.5}$$

where $\{\sigma_k\}$ is a positive numbers sequence such that $0 < \sigma_k \uparrow \sigma_\infty \le +\infty$. Clearly, if $\mathscr{M}_k \equiv \mathscr{I}$, i.e., an identity matrix, the P-PPA may reduce to the classical PPA of Rockafellar [26, 27]. It should be noted that there are two general criteria for the approximation between $x^{k+1}$ and $\bar{x}^{k+1}$, that is,

(A) $\left\|x^{k+1} - \bar{x}^{k+1}\right\|_{\mathscr{M}_k} \le \varepsilon_k$ with $0 \le \varepsilon_k$ and $\sum_{k=0}^\infty \varepsilon_k < +\infty$,

(B) $\left\|x^{k+1} - \bar{x}^{k+1}\right\|_{\mathscr{M}_k} \le \delta_k \left\|x^{k+1} - x^k\right\|_{\mathscr{M}_k}$ with $0 \le \delta_k < 1$ and $\sum_{k=0}^\infty \delta_k < +\infty$.

Under criteria conditions (A) and (B), the convergence properties of (2.5) can be obtained theoretically. For more details, one may refer to [20]. Here, to make this paper completeness, we only list the convergence theorem without proof.

**Theorem 2.1.** *(*[20, Theorem 1]*) Suppose that* $\Omega := \{x \mid 0 \in \partial f(x)\} \ne \emptyset$. *Let* $\{x^k\}$ *be the sequence generated by P-PPA (2.5) under criterion (A). Then* $\{x^k\}$ *is bounded and satisfies*

$$\operatorname{dist}_{\mathscr{M}_{k+1}}\left(x^{k+1}, \Omega\right) \le (1+\rho_k)\operatorname{dist}_{\mathscr{M}_k}\left(x^k, \Omega\right) + (1+\rho_k)\varepsilon_k, \quad \forall k \ge 0.$$

*In addition,* $\{x^k\}$ *converges to a point* $x^\infty$ *such that* $0 \in \partial f(x^\infty)$.

In addition, it was also proved that P-PPA (2.5) has asymptotic suplinear rate when criterion (A) and (B) are used. This is described as follows:

**Assumption 2.1.** The operator $\partial f$ satisfies error bound condition, that is, for any $\gamma > 0$, there exists a $\kappa > 0$ such that

$$\operatorname{dist}\left(x,(\partial f)^{-1}(0)\right) \leq \kappa \operatorname{dist}(0,\partial f(x)), \quad \forall x \in \left\{x \mid \operatorname{dist}\left(x,(\partial f)^{-1}(0)\right) \leq \gamma\right\}.$$

**Theorem 2.2.** *(*[20, Theorem 2]*) Suppose that $\Omega \neq \emptyset$ and Assumption 2.1 holds. Let $t$ be a positive number satisfying $t > \sum_{k=0}^{\infty} \varepsilon_k (1+\rho_k)$ and $x^0$ be an initial point such that*

$$\operatorname{dist}_{\mathscr{M}_0}\left(x^0,\Omega\right) \leq \frac{t - \sum_{k=0}^{\infty} \varepsilon_k (1+\rho_k)}{\prod_{k=0}^{\infty}(1+\rho_k)}.$$

*Let $\{x^k\}$ be a sequence generated by P-PPA (2.5) under criteria (A) or (B). Then*

$$\operatorname{dist}_{\mathscr{M}_{k+1}}\left(x^{k+1},\Omega\right) \leq \theta_k \operatorname{dist}_{\mathscr{M}_k}\left(x^k,\Omega\right),$$

*where*

$$\theta_k := (1+\rho_k)(1-\delta_k)^{-1}\left(\delta_k + \frac{(1+\delta_k)\kappa\lambda_{\max}(\mathscr{M}_k)}{\sqrt{\sigma_k^2 + \kappa^2\lambda_{\max}^2(\mathscr{M}_k)}}\right)$$

*and*

$$\limsup_{k\to\infty} \theta_k = \theta_\infty := \frac{\kappa\lambda_\infty}{\sqrt{\sigma_\infty^2 + \kappa^2\xi_\infty^2}} < 1 \quad \text{with } \theta_\infty = 0 \text{ if } \sigma_\infty \to \infty.$$

Based on the perspective of numerical calculation, we summarize some existing results to implement the algorithm P-PPA and the subalgorithm SSN. We let $\Pi_{B_p^{(r)}}(\cdot)$ be the orthogonal projection onto the $\ell_p$-norm ball with radius $r > 0$. It is known that, for $\ell_p$-norm ball with $p = 1$, 2, and $\infty$, the proximal mapping for $\ell_p$-norm function is easily implemented. Here, for convenience, we summarize these results in the following lemma.

**Lemma 2.1.** *For any given $z \in \mathbb{R}^n$, it holds that:*
*(i) If $f(x) = \mu\|x\|_1$ with $\mu > 0$, then $f^*(z) = \delta_{B_\infty^{(\mu)}}(z)$ with $B_\infty^{(\mu)}(z) := \{z \mid \|z\|_\infty \leq \mu\}$ and*

$$\operatorname{Prox}_f(z) = z - \Pi_{B_\infty^{(\mu)}}(z) \quad \text{with} \quad (\Pi_{B_\infty^{(\mu)}}(z))_i = \begin{cases} z_i, & \text{if } |z_i| \leq \mu, \\ \operatorname{sign}(z_i)\mu, & \text{if } |z_i| > \mu. \end{cases}$$

*(ii) If $f(x) = \mu\|x\|_2$ with $\mu > 0$, then $f^*(z) = \delta_{B_2^{(\mu)}}(z)$ with $B_2^{(\mu)}(z) := \{z \mid \|z\|_2 \leq \mu\}$ and*

$$\operatorname{Prox}_f(z) = z - \Pi_{B_2^{(\mu)}}(z) \quad \text{with} \quad \Pi_{B_2^{(\mu)}}(z) = \begin{cases} z, & \text{if } \|z\|_2 \leq \mu, \\ \mu\frac{z}{\|z\|_2}, & \text{if } \|z\|_2 > \mu. \end{cases}$$

*(iii)* [21] *If $f(x) = \mu\|x\|_\infty$ with $\mu > 0$, then $f^*(z) = \delta_{B_1^{(\mu)}}(z)$ with $B_1^{(\mu)}(z) := \{z \mid \|z\|_1 \leq \mu\}$ and*

$$\operatorname{Prox}_f(z) = z - \Pi_{B_1^{(\mu)}}(z), \quad \text{with} \quad \Pi_{B_1^{(\mu)}}(z) = \begin{cases} z, & \text{if } \|z\|_1 \leq \mu, \\ \mu P_z \Pi_{\Delta_n}(P_z z/\mu), & \text{if } \|z\|_1 > \mu. \end{cases}$$

*where $P_z = \operatorname{Diag}(\operatorname{sign}(z)) \in \mathbb{R}^{n\times n}$ and $\Pi_{\Delta_n}(\cdot)$ denotes the projection onto the simplex $\Delta_n = \{z \in \mathbb{R}^n \mid e_n^T z = 1, z \geq 0\}$.*

The following lemma summarizes the compact forms of the generalized Jacobian of proximal mappings for $\ell_p$-norm functions. The results are fundamental to construct the generalized Hessian $\hat{\mathscr{H}}$ used in the Step 1 of Algorithm SSN.

**Lemma 2.2.** *For any given $\vartheta \in \mathbb{R}^n$, one has the following assertions.*

*(a) The Clarke subdifferential of* $\mathrm{Prox}_{\mu\|\cdot\|_1}(\cdot)$ *at* $\vartheta$ *is given by*

$$\partial \mathrm{Prox}_{\mu\|\cdot\|_1}(\vartheta) = \left\{ \mathrm{Diag}(\theta) \mid \theta \in \mathbb{R}^n, \theta_i \in \left\{ \begin{array}{ll} \{1\}, & \text{if } |\vartheta_i| > \mu, \\ [0,1], & \text{if } |\vartheta_i| = \mu, \\ \{0\}, & \text{if } |\vartheta_i| < \mu, \end{array} \quad i = 1,\ldots,n \right\} \right.$$

*In the numerical experiment, we choose the following element in* $\partial \mathrm{Prox}_{\mu\|\cdot\|_1}(\vartheta)$*:*

$$\hat{\Theta} = \mathrm{Diag}(\theta) \text{ with } \theta_i = \left\{ \begin{array}{ll} 1, & \text{if } |\vartheta_i| > \mu, \\ 0, & \text{if } |\vartheta_i| \le \mu, \end{array} \quad i = 1,\ldots,n. \right.$$

*(b) The Clarke subdifferential of* $\mathrm{Prox}_{\mu\|\cdot\|_2}(\cdot)$ *at* $\vartheta$ *is given by*

$$\partial \mathrm{Prox}_{\mu\|\cdot\|_2}(\vartheta) = \left\{ \begin{array}{ll} \{(1-\frac{\mu}{\|\vartheta\|_2})\mathscr{I}_n + \mu\frac{\vartheta\vartheta^\top}{\|\vartheta\|_2^3}\}, & \text{if } \|\vartheta\|_2 > \mu, \\ \{\kappa\frac{\vartheta\vartheta^\top}{\mu^2} \mid 0 \le \kappa \le 1\}, & \text{if } \|\vartheta\|_2 = \mu, \\ \{\mathbf{0}_n\}, & \text{if } \|\vartheta\|_2 < \mu. \end{array} \right.$$

*In the numerical experiment, we choose the following element in* $\partial \mathrm{Prox}_{\mu\|\cdot\|_2}(\vartheta)$*:*

$$\hat{\Theta} = \left\{ \begin{array}{ll} \{(1-\frac{\mu}{\|\vartheta\|_2})\mathscr{I}_n + \mu\frac{\vartheta\vartheta^\top}{\|\vartheta\|_2^3}\}, & \text{if } \|\vartheta\|_2 > \mu, \\ \{\mathbf{0}_n\}, & \text{if } \|\vartheta\|_2 \le \mu. \end{array} \right.$$

*(c)* [21] *The Clarke subdifferential of* $\mathrm{Prox}_{\mu\|\cdot\|_\infty}(\cdot)$ *at* $\vartheta$ *is given by*

$$\partial \mathrm{Prox}_{\mu\|\cdot\|_\infty}(\vartheta) = \mathscr{I}_n - H, H \in \partial\Pi_{B_1^{(\mu)}}(\vartheta) \quad \text{where} \quad H = \left\{ \begin{array}{ll} P_\vartheta \tilde{H} P_\vartheta, & \text{if } \|\vartheta\|_1 > \mu, \\ \mathscr{I}_n, & \text{if } \|\vartheta\|_1 \le \mu, \end{array} \right.$$

*where* $\widetilde{H} = \mathrm{Diag}(r) - \frac{1}{\mathrm{nnz}(r)} r r^\top \in \partial\Pi_{\Delta_n}(\vartheta)$ *with* $r \in \mathbb{R}^n$ *being defined as* $r_i = 1$ *if* $(\Pi_{\Delta_n}(P_\vartheta \vartheta/\mu))_i \neq 0$*, and* $r_i = 0$ *otherwise. In the numerical experiment, we choose the following element in* $\partial \mathrm{Prox}_{\mu\|\cdot\|_\infty}(\vartheta)$*:*

$$\hat{\Theta} = \left\{ \begin{array}{ll} \{\mathscr{I}_n - P_\vartheta \tilde{H} P_\vartheta\}, & \text{if } \|\vartheta\|_1 > \mu, \\ \{\mathbf{0}_n\}, & \text{if } \|\vartheta\|_1 \le \mu. \end{array} \right.$$

## 3. SSN BASED P-PPA METHOD

3.1. **P-PPA method and some properties.** For convenience, we denote the objective function of (1.3) as $f(x)$, i.e.,

$$f(x) := \|Ax - b\|_p + \lambda\left(\|x\|_1 + \frac{\beta}{2}\|x\|_2^2\right).$$

We let $\tilde{x} \in \mathbb{R}^n$ be a given point and consider the following minimization problem

$$\min_{x\in\mathbb{R}^n}\left\{\tilde{f}(x;\sigma,\tau,\tilde{x}) := f(x) + \frac{\sigma}{2}\|x-\tilde{x}\|_2^2 + \frac{\tau}{2}\|Ax - A\tilde{x}\|_2^2\right\}, \tag{3.1}$$

where $\sigma > 0$ and $\tau > 0$ are given positive scalars and can be determined dynamically. It should be noted that the last term in (3.1) is actually a precondition, which is used to derive a dual

problem with favorable structures. From an initial point $x^0$, the algorithm described below generates a sequence $\{x^k\}$ by using a nonincreasing sequence $\{\sigma_k, \tau_k\}$ such that

$$x^{k+1} \approx \arg\min_{x\in\mathbb{R}^n} \tilde{f}(x;\sigma_k,\tau_k,x^k), \tag{3.2}$$

where the criteria of the approximate equality is from Rockafellar [27]. Let $\mathscr{M}^k := \sigma_k\mathscr{I} + \tau_k A^\top A$ with $\mathscr{I}$ be an identity operator. Then, it is trivial to deduce that (3.2) takes the following rule:

$$x^{k+1} \approx \mathscr{P}^k(x^k) \quad \text{with} \quad \mathscr{P}^k := (\mathscr{M}^k + c^k\partial f^k)^{-1}\mathscr{M}^k, \tag{3.3}$$

where $\partial f^k$ is a subdifferentiable of $f$ at $x^k$ and $\{c^k\}$ is a sequence of positive numbers. If $\tau_k \equiv 0$, the updating scheme (3.3) reduces the traditional PPA considered by Rockafellar [26]. Because $\mathscr{M}^k + c^k\partial f^k$ is a strongly monotone operator, it is known from [29, Proposition 12.54] that $\mathscr{P}^k$ is single-valued and is globally Lipschitz continuous. From (3.3), we see that the iterative framework (3.2) actually fills into the framework of the P-PPA analyzed by Li et al. [20], and then some theoretical properties can be followed directly.

We now establish some relations between problems (3.1) and (1.3) under some certain conditions. The first proposition listed below shows that $\tilde{f}(x;\sigma,\tau,\tilde{x})$ converges to $f(x)$ when the positive numbers $\sigma$ and $\tau$ are sufficiently small. The second proposition shows that the optimal solution $x^*$ of problem (1.3) is actually the minimizer of $\tilde{f}(x;\sigma,\tau,\tilde{x})$ in the case of $\tilde{x} \equiv x^*$. A similar proof of both propositions can be found in [30, Theorem 15] and [11, Theorem 3.1, Lemma 3.2]. Here, we report the proof for the completeness of this paper.

**Proposition 3.1.** *The optimal objective value of problem (3.1) converges to the optimal objective value of problem (1.3) if $\sigma,\tau \downarrow 0$, that is,*

$$\lim_{\sigma,\tau\downarrow 0} \hat{f}(\sigma,\tau) = \min_{x\in\mathbb{R}^n} f(x),$$

*where*

$$\hat{f}(\sigma,\tau) := \min_{x\in\mathbb{R}^n} \tilde{f}(x;\sigma,\tau,\tilde{x}).$$

*Proof.* Firstly, it holds that $\hat{f}(\sigma,\tau) \geq \min_{x\in\mathbb{R}^n} f(x)$. For any $\sigma > 0, \tau > 0$, and $x \in \mathbb{R}^n$, we have that

$$\hat{f}(\sigma,\tau) \leq \|Ax-b\|_p + \lambda(\|x\|_1 + \frac{\beta}{2}\|x\|_2^2) + \frac{\sigma}{2}\|x-\tilde{x}\|_2^2 + \frac{\tau}{2}\|Ax-A\tilde{x}\|_2^2.$$

Taking limits on both hand-sides of this inequality as $\sigma,\tau \to 0$, we obtain that

$$\lim_{\sigma,\tau\downarrow 0} \hat{f}(\sigma,\tau) \leq \|Ax-b\|_p + \lambda(\|x\|_1 + \frac{\beta}{2}\|x\|_2^2),$$

which means

$$\lim_{\sigma,\tau\downarrow 0} \hat{f}(\sigma,\tau) \leq \min_{x\in\mathbb{R}^n} f(x).$$

Therefore, the desired result follows directly. □

**Proposition 3.2.** *A vector $x^*$ is an optimal solution to(1.3) if and only if there exist $\sigma \geq 0$ and $\tau \geq 0$ such that*

$$x^* \in \operatorname{argmin}_{x\in\mathbb{R}^n} \tilde{f}(x;\sigma,\tau,x^*),$$

*i.e, the optimal solution to (1.3) is actually the one of (3.1) if and only if $\tilde{x} \equiv x^*$.*

*Proof.* Noting that $f(\cdot)$ is locally Lipschitz continuous near $x^*$ and convex at $x^*$, one has that $0 \in \partial f(x^*)$ is equivalent to $x^*$ being an optimal solution to $f$. It is not difficult to see that function $\tilde{f}(x;\sigma,\tau,x^*)$ is convex. Thus $0 \in \partial \tilde{f}(x^*;\sigma,\tau,x^*)$ is equivalent to $x^* \in \operatorname{argmin}_{x\in\mathbb{R}^n}\{\tilde{f}(x;\sigma,\tau,x^*)\}$. Combining with $\partial f(x^*) = \partial \tilde{f}(x^*;\sigma,\tau,x^*)$, we can easily obtain the conclusion of this theorem. □

3.2. **The dual problem and semismooth equations.** Despite some nice theoretical properties, finding a solution to (3.1) with higher accuracy is not a trivial task. Because the first-order method, such as ADMM, is only suitable for deriving lower to medium quality solutions. To address this issue, in this subsection, we turn to using a second-order method, named SSN, to find a solution of (3.1) rapidly from a perspective of dual. For our purpose, we now give the Lagrangian dual formulation of problem (3.1). Let $y := Ax - b$. Then (3.1) takes the following equivalent form

$$\begin{aligned} \min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} \quad & \|y\|_p + \lambda\left(\|x\|_1 + \tfrac{\beta}{2}\|x\|_2^2\right) + \tfrac{\sigma}{2}\|x-\tilde{x}\|_2^2 + \tfrac{\tau}{2}\|y+b-A\tilde{x}\|_2^2 \\ \text{s.t.} \quad & Ax - y = b. \end{aligned} \tag{3.4}$$

The Lagrangian function associated with problem (3.4) is given by

$$\mathscr{L}(x,y;u) = \|y\|_p + \lambda\left(\|x\|_1 + \frac{\beta}{2}\|x\|_2^2\right) + \frac{\sigma}{2}\|x-\tilde{x}\|_2^2 + \frac{\tau}{2}\|y+b-A\tilde{x}\|_2^2 + \langle u, Ax-y-b\rangle,$$

where $u \in \mathbb{R}^m$ is a multiplier associated with the constraint. From some basic theories of optimization, we know that the Lagrangian dual function, denoted as $D(u)$ here, is defined as the minimum value of the Lagrangian function over $(x,y)$, that is,

$$\begin{aligned} D(u) &= \inf_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} \mathscr{L}(x,y;u) \\ &= \inf_{x\in\mathbb{R}^n}\left\{\lambda\left(\|x\|_1 + \frac{\beta}{2}\|x\|_2^2\right) + \langle u, Ax\rangle + \frac{\sigma}{2}\|x-\tilde{x}\|_2^2\right\} + \inf_{y\in\mathbb{R}^m}\left\{\|y\|_p + \frac{\tau}{2}\|y+b-A\tilde{x}\|_2^2 - \langle u, y\rangle\right\} - \langle u, b\rangle \\ &= \lambda\Phi_{\gamma^{-1}\lambda\|\cdot\|_1}\left(\gamma^{-1}(\sigma\tilde{x} - A^\top u)\right) - \frac{\gamma}{2}\|\gamma^{-1}(\sigma\tilde{x} - A^\top u)\|_2^2 + \frac{\sigma}{2}\|\tilde{x}\|_2^2 \\ &\quad + \Phi_{\tau^{-1}\|\cdot\|_p}\left(\tau^{-1}u - b + A\tilde{x}\right) - \frac{\tau}{2}\|\tau^{-1}u - b + A\tilde{x}\|_2^2 + \frac{\tau}{2}\|b - A\tilde{x}\|_2^2 - \langle u, b\rangle, \end{aligned}$$

where $\gamma := \lambda\beta + \sigma$ and $\Phi(\cdot)$ is a Moreau envelop function defined in (2.1). From this deduce process, we know that the preconditioned term $\|Ax - A\tilde{x}\|_2^2$ in (3.1) is essential to make $D(u)$ allow favorable structures in sense of using Moreau envelop functions.

We now focus on both inf-operations involved in $D(u)$. By the first-order optimality conditions of the $x$- and $y$-subproblems, we know that its minimizers can be expressed explicitly as

$$x = \arg\min_{x\in\mathbb{R}^n}\left\{\lambda\|x\|_1 + \frac{\lambda\beta}{2}\|x\|_2^2 + \langle u, Ax\rangle + \frac{\sigma}{2}\|x-\tilde{x}\|_2^2\right\} = \operatorname{Prox}_{\gamma^{-1}\lambda\|\cdot\|_1}\left(\gamma^{-1}(\sigma\tilde{x} - A^\top u)\right), \tag{3.5}$$

and

$$y = \arg\min_{y\in\mathbb{R}^m}\left\{\|y\|_p + \frac{\tau}{2}\|y+b-A\tilde{x}\|_2^2 - \langle u, y\rangle\right\} = \operatorname{Prox}_{\tau^{-1}\|\cdot\|_p}\left(\tau^{-1}u - b + A\tilde{x}\right), \tag{3.6}$$

where the symbol 'Prox$(\cdot)$' denotes the proximal mapping defined in (2.2). Equality (3.5) clarifies the relations between primal variable $x$ and dual variable $u$, which means that if $u$ is

known, then $x$ can be obtained in a compact form. In the previous section, we have shown that the proximal mapping operation of $\ell_p$-norm with $p = 1$, 2, and $\infty$ are easily performed so that the $x$ and $y$ can be easily derived.

The Lagrangian dual problem of (3.4) is to maximize the dual function $D(u)$, which can be equivalently formulated as the following optimization problem

$$\min_{u\in\mathbb{R}^m}\Big\{\Theta(u) := \langle u,b\rangle - \mathscr{X}(u) - \mathscr{Y}(u)\Big\}, \tag{3.7}$$

where

$$\mathscr{X}(u) := \lambda\Phi_{\gamma^{-1}\lambda\|\cdot\|_1}\big(\gamma^{-1}(\sigma\tilde{x}-A^\top u)\big) - \frac{\gamma}{2}\|\gamma^{-1}(\sigma\tilde{x}-A^\top u)\|_2^2 + \frac{\sigma}{2}\|\tilde{x}\|_2^2,$$

and

$$\mathscr{Y}(u) := \Phi_{\tau^{-1}\|\cdot\|_p}\big(\tau^{-1}u-b+A\tilde{x}\big) - \frac{\tau}{2}\|\tau^{-1}u-b+A\tilde{x}\|_2^2 + \frac{\tau}{2}\|b-A\tilde{x}\|_2^2.$$

Noting that $\Theta(u)$ is convex and continuously differentiable, one sees that its gradient takes the following compact form by using (2.3), that is,

$$\nabla\Theta(u) = b - A\mathrm{Prox}_{\gamma^{-1}\lambda\|\cdot\|_1}\big(\gamma^{-1}(\sigma\tilde{x}-A^\top u)\big) + \mathrm{Prox}_{\tau^{-1}\|\cdot\|_p}\big(\tau^{-1}u-b+A\tilde{x}\big).$$

Therefore, the optimal solution of problem (3.7) can be obtained by solving the following nonlinear system of equations $\nabla\Theta(u) = 0$. It is known in optimization literature that the proximal mapping $\mathrm{Prox}_{\|\cdot\|_p}(\cdot)$ in the case of $p = 1$, 2, and $\infty$ is strongly semismooth [21], so does $\nabla\Theta(u)$. Therefore, we can utilize the efficient SSN to obtain high-precision solutions.

3.3. **SSN method for solving (3.7).** In this subsection, we focus on employing an efficient SSN (Semismooth Newton) method to find an approximated solution $\bar{u}$ of problem (3.7), and then derive the corresponding approximated solution $\bar{x}$ of problem (3.1) and the optimal solution $x^*$ of problem (1.3) theoretically by Proposition 3.2. From (3.7), we know that $\Theta(u)$ is (but not second-order) continuously differentiable, which indicates that the Hessian matrix $\nabla^2\Theta(u)$ is unavailable. Instead, we use the generalized Hessian, or the generalized Jacobian of $\nabla\Theta(u)$, that is,

$$\tilde{\partial}^2\Theta(u) := \gamma^{-1}A\partial\mathrm{Prox}_{\gamma^{-1}\lambda\|\cdot\|_1}\big(\gamma^{-1}(\sigma\tilde{x}-A^\top u)\big)A^\top + \tau^{-1}\partial\mathrm{Prox}_{\tau^{-1}\|\cdot\|_p}\big(\tau^{-1}u-b+A\tilde{x}\big),$$

where '$\tilde{\partial}^2$' is named as generalized Hessian of $\Theta(\cdot)$, and $\partial(\cdot)$ is the generalized (a.k.s. Clarke) Jacobian [7]. Choose

$$U \in \partial\mathrm{Prox}_{\gamma^{-1}\lambda\|\cdot\|_1}\big(\gamma^{-1}(\sigma\tilde{x}-A^\top u)\big) \quad \text{and} \quad V \in \partial\mathrm{Prox}_{\tau^{-1}\|\cdot\|_p}\big(\tau^{-1}u-b+A\tilde{x}\big),$$

and then set $\mathscr{H} := \gamma^{-1}AUA^\top + \tau^{-1}V$. Thus we have $\mathscr{H} \in \tilde{\partial}^2\Theta(u)$.

It is known in optimization literature that, starting from $u^0$, the SSN method generates a sequence $\{u^i\}$ such that

$$u^{i+1} = u^i + \alpha_i\Delta u^i, \quad \text{where} \quad \mathscr{H}\Delta u^i + \nabla\Theta(u^i) = 0,$$

where $\alpha_i$ is a steplength and $\Delta u^i$ is a search direction. Theoretically, under the condition that $\mathscr{H}$ is nonsingular at $\bar{u}$, $\{u^i\}$ converges to $\bar{u}$ at least local superlinerly. However, this nonsingular assumption may violate which means that the SSN method can not be employed any more. Specially, some additional conditions should be included to ensure the positive definiteness of $\mathscr{H}$, e.g., [11, Assumptions 3.2 and 3.3]. However, these conditions seem to be somewhat strong although they are satisfied in some special cases.

To address this issue, we modify $\mathscr{H}$ by adding a small term, that is,

$$\hat{\mathscr{H}} := \mathscr{H} + \nu^{-1}\mathscr{I},$$

where $\nu > 0$ is a small undetermined scalar, which can be chosen dynamically and make that $\hat{\mathscr{H}}$ be positive definite.

3.4. **Iterative framework of the SSN based P-PPA method.** In view of the analysis above, we are in a position to state the iterative framework of SSN based P-PPA method.

**Algorithm: P-PPA**

**Step 0**: Give $\lambda > 0$ and $\beta > 0$. Chose $\sigma_0 > 0$, $\tau_0 > 0$, $x^0 \in \mathbb{R}^n$, and set $k := 0$. Do the following two steps iteratively:

**Step 1**: Find a solution $x^{k+1}$ such that

$$x^{k+1} \approx \arg\min_{x\in\mathbb{R}^n}\{\tilde{f}(x;\sigma_k,\tau_k,x^k)\}$$

via the following steps:

(i) Set $\sigma := \sigma_k$, $\tau := \tau_k$, and $\gamma := \lambda\beta + \sigma$. Choose $\nu > 0$. Applying SSN to find a solution $u^{k+1}$ such that

$$u^{k+1} \approx \arg\min_{u\in\mathbb{R}^m} \Theta(u).$$

(ii) Set $x^{k+1} := \mathrm{Prox}_{\gamma^{-1}\lambda\|\cdot\|_1}\big(\gamma^{-1}(\sigma x^k - A^\top u^{k+1})\big)$.

**Step 2**: Update $\sigma_{k+1} = \rho\sigma_k$ and $\tau_{k+1} = \rho\tau_k$ with $\rho \in (0,1)$. Let $k := k+1$ and go to **Step 1**.

From the steps of Algorithm P-PPA, we observe that main computational burden lies in Step 1(i) to find an inexact optimizer $u^{k+1}$ of function $\Theta(u)$. Since $\nabla\Theta(u)$ is clearly strongly semismooth, we will employ an efficient SSN method for its solution. The full steps of the SSN method are the following. For more details on the SSN as well as its different applications, one may refer to the papers of Li et al. [19], Tang et al. [30], Ding et al. [11] and the references therein..

**Algorithm: SSN**

**Step 0**: Given $\sigma > 0$, $\tau > 0$, $x^k \in \mathbb{R}^n$, choose $\mu \in (0,1/2)$, $\bar{\eta} \in (0,1)$, $\rho \in (0,1]$, $\delta \in (0,1)$, and $u^0 \in \mathbb{R}^m$. Let $i := 0$. Do the following steps iteratively:

**Step 1**: Select $U^i \in \partial\mathrm{Prox}_{\gamma^{-1}\lambda\|\cdot\|_1}\big(\gamma^{-1}(\sigma x^k - A^\top u^i)\big)$ and $V^i \in \partial\mathrm{Prox}_{\tau^{-1}\|\cdot\|_p}\big(\tau^{-1}u^i - b + Ax^k\big)$, and set $\hat{\mathscr{H}}^i := \gamma^{-1}AU^iA^\top + \tau^{-1}V^i + \nu^{-1}\mathscr{I}$. Employ a numerical method to find an approximate solution $\Delta u^i$ to the linear system

$$\hat{\mathscr{H}}^i\Delta u + \nabla\Theta(u^i) = 0$$

such that

$$\|\hat{\mathscr{H}}^i\Delta u^i + \nabla\Theta(u^i)\|_2 \le \min\{\bar{\eta}, \|\nabla\Theta(u^i)\|_2^{1+\rho}\}.$$

**Step 2**: Find $\alpha_i := \delta^{t_i}$, where $t_i$ is the first nonnegative integer $t$ such that

$$\Theta(u^i + \delta^t\Delta u^i) \le \Theta(u^i) + \mu\delta^t\langle\nabla\Theta(u^i), \Delta u^i\rangle.$$

**Step 3**: Set $u^{i+1} := u^i + \alpha_i \Delta u^i$.

We see from the steps of the SSN method that, starting from $u^0$, SSN generates a sequence $\{u^i\}$, which converges to a minimizer $\bar{u}$ of function $\Theta(u)$ theoretically, i.e., $\nabla\Theta(\bar{u}) = 0$. However, to make the algorithm more practical and with convergence guarantee, it is suitable to use the following stopping criterion, that is,

$$\left\|\nabla\Theta\left(u^{i+1}\right) + \nu^{-1}\left(u^{i+1} - u^i\right)\right\|_2 \leq \frac{\delta_i}{\nu}\left\|u^{i+1} - u^i\right\|,$$

where $\{\delta_i\}$ is sum-able sequence such that $\delta_i \in [0,1)$ and $\sum_i^\infty \delta_i < \infty$. For more details on this stopping criterion, one may refer to Rockafellar [26, 27] and Li et al. [20]. In this case, it is from Theorem 2.2 that $\{x^k\}$ converges globally to problem (1.3). The convergence result is omitted here to avoid repetition.

At the end of this section, we report the local convergence rate of Algorithm SSN. Noting that $\nabla\Theta(u)$ is strongly semismooth, one sees that $\{u^i\}$ converges to $\bar{u}$ superlinearly and even quadratically.

**Theorem 3.1.** *The sequence $\{u^i\}$ generated by the SSN method converges to the unique solution $\bar{u}$ of the strongly semismooth equation $\nabla\Theta(u) = 0$ with*

$$\|u^{i+1} - \bar{u}\|_2 = \mathcal{O}(\|u^i - \bar{u}\|_2^{1+\varsigma}),$$

*where $\varsigma \in (0,1]$*

*Proof.* See [19, Theorem 3.5]. □

## 4. Numerical Experiments

In this section, we test the effectiveness and accuracy of P-PPA on sparse linear regression problems with $\ell_p$-norm data fidelity and elastic net regularization using both simulated and real data. All the computations were performed with Microsoft Windows 11 and MATLAB R2022b, and run on a PC with an Intel Core i7-10710U CPU at 1.10 GHz and 16 GB of memory.

### 4.1. Experiment setup.

4.1.1. *Data generation.* First, we describe the data generation part in detail. In order to fully illustrate the effectiveness of the P-PPA algorithm, we consider two different coefficient matrices $A \in \mathbb{R}^{m\times n}$: random Gaussian matrix and random partial DCT matrix. Specifically,

- **(1):** Random Gaussian matrix: Each entry of random Gaussian matrix follows the standard normal distribution, i.e., zero-mean with standard deviation of one.
- **(2):** Random partial DCT matrix: We randomly select rows from the full DCT matrix to form partial DCT matrix.

All the testing matrices are normalized to have unit (spectral) norms. The true sparse signal $x^*$ is a strict $K$-sparse signal with an active set (indices of nonzero components) denoted by $\mathscr{A}^*$, and its dynamic range $R$ is defined by $R = R_1/R_2$ with $R_1 = \max\left\{|x_i^*| : i \in \mathscr{A}^*\right\}$ and $R_2 = \min\left\{|x_i^*| : i \in \mathscr{A}^*\right\}$. To fully demonstrate the robustness and applicability of model (1.3), we consider three types of noise in this study. In practice, different values of $p$ correspond to different types of noise, as detailed below:

**(1):** $p = 1$ : Log-normal noise.
**(2):** $p = 2$ : Gaussian noise.
**(3):** $p = \infty$ : Uniformly distributed noise.

Then, the observed data $b$ is generated by $b = Ax^* + \kappa * \varepsilon$, where $\varepsilon$ denotes the type of noise, and $\kappa$ is the noise level. Throughout the experiment, we uniformly set $\kappa = 1e-3$, $\rho = 0.9$, $\mu = 0.01$, and $\delta = 0.8$. The values of other parameters will be adaptively determined in each experiment.

4.1.2. *Evaluation metrics.* To evaluate the numerical performance of the considered algorithms from multiple perspectives, we consider three evaluation metrics:

**(1):** $RLNE := \frac{\|\hat{x}-x^*\|_2}{\|x^*\|_2}$, where $\hat{x}$ is the estimated sparse signal. $RLNE$ describes the recovery quality.
**(2):** $ResErr := \frac{\|Ax-b\|_2}{1+\|b\|_2}$. $ResErr$ demonstrates the computational accuracy.
**(3):** $RelErr := \frac{\|x^{k+1}-x^k\|_2}{1+\|x^k\|_2+\|x^{k+1}\|_2}$. $RelErr$ captures the descending trend of the algorithm's iteration sequence.

In the entire numerical experiment, we utilize $RelErr \leq 1e-5$ or iterate steps exceeding 7000 as the termination condition. These metrics provide a comprehensive assessment of the algorithm's performance in terms of recovery quality, computational accuracy, and convergence speed, enabling a thorough evaluation of its effectiveness and reliability.

4.2. **The behavior of P-PPA.** In this subsection, to demonstrate the numerical performance of the P-PPA algorithm, we conducted separate tests on the recovery effect of two coefficient matrices under three types of noise for a fixed sparse signal. We fixed the size of the coefficient matrix $A$ at $500 \times 1000$. We pre-generated a fixed true sparse signal $x^*$ containing 10 non-zero elements, that is, $x_i^* \equiv 0$ except for $x_{125}^* = -1.0000$, $x_{224}^* = -1.0270$, $x_{392}^* = 1.6313$, $x_{533}^* = 3.1623$, $x_{716}^* = 3.0626$, $x_{786}^* = 1.3702$, $x_{820}^* = -3.0201$, $x_{833}^* = -1.3848$, $x_{956}^* = -2.6705$, $x_{961}^* = 1.4610$. Box plots in Figures 1-2 illustrate the estimation performance of the P-PPA algorithm regarding the positions of non-zero elements based on 10 independent experiments. From the figures, it is evident that the P-PPA algorithm consistently and accurately identifies the positions of non-zero elements across all 10 independent experiments, and it can almost perfectly estimate the values of non-zero elements. This result sufficiently indicates that the P-PPA algorithm can successfully accomplish variable selection and parameter estimation.

4.3. **Numerical comparison.** Considering the similar numerical performance of semi-proximal ADMM and directly extended ADMM for model (1.3) in [10], we specifically compare the numerical results of directly extended ADMM (referred to as dADMM) with P-PPA algorithm. The comparison is conducted in terms of computational accuracy and efficiency by both simulated and real data.

4.3.1. *Simulated examples.* In the simulation experiments, we test two different coefficient matrices with three sizes: $200 \times 500$, $500 \times 800$, and $800 \times 1100$. For each size, we also consider three pairs of $(\lambda, \beta)$. The dynamic range $R$ of the true sparse signal $x^*$ is fixed at 1000. We compare P-PPA and dADMM in terms of six aspects: number of outer iterations (Iter), CPU time (Time) in seconds, objective function value (Obj) of (1.3), RLNE, ResErr, and RelErr. The specific numerical results can be found in Table 1-2.

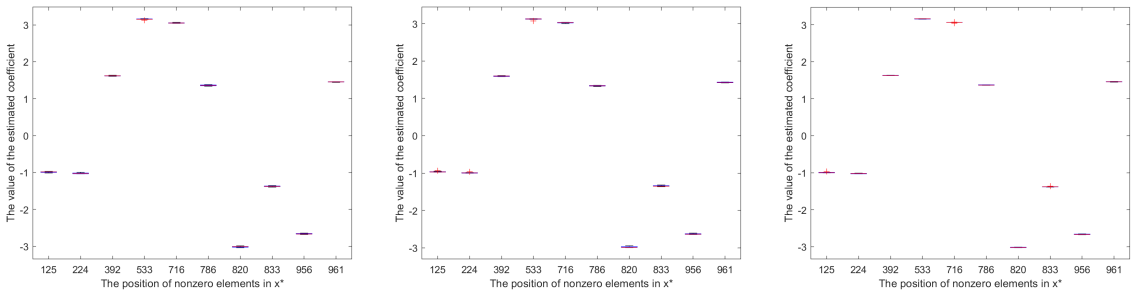


FIGURE 1. Estimation performance of P-PPA algorithm with random Gaussian matrix under different noises (Left to rigth: log-normal noise, Gaussian noise, uniformly distributed noise).

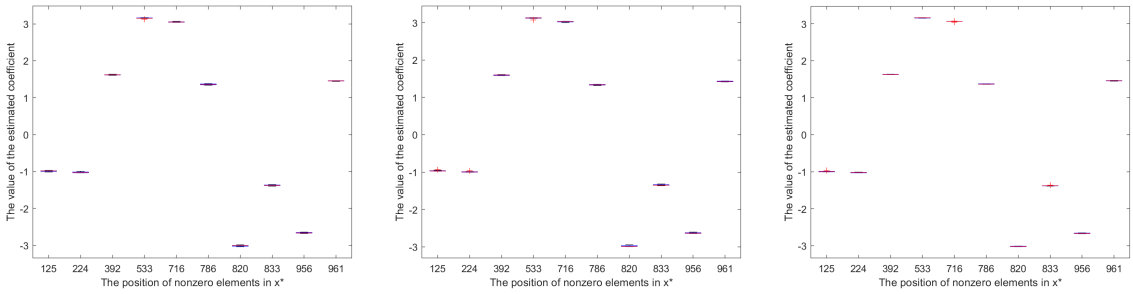


FIGURE 2. Estimation performance of P-PPA algorithm with random partial DCT matrix under different noises (Left to rigth: log-normal noise, Gaussian noise, uniformly distributed noise).

From the results in the tables, it can be observed that in the current high-dimensional scenario, P-PPA consistently requires significantly fewer iterations and less computation time compared to dADMM. This phenomenon clearly demonstrates that P-PPA possesses faster computational speed. We also find that both algorithms experience a substantial increase in number of iterations and computation time when dealing with uniformly noisy interference. This is attributed to the fact that the proximal mapping of the $\ell_\infty$-norm function and its subdifferential involve a complex task of projecting onto the simplex, which undoubtedly requires a significant amount of time for computation, especially in the SSN algorithm of P-PPA. Otherwise, if both algorithms reach the termination condition within the maximum iteration steps, the optimal objective function values are nearly equal. However, for cases where dADMM reaches the maximum iteration steps, the optimal objective function value of P-PPA is consistently lower than that of dADMM.

For the two indicators of RLNE and ResErr, which characterize the recovery quality and computational accuracy of the algorithms, the corresponding values of these two indicators for P-PPA are almost at the magnitude of $-4$. This clearly demonstrates that P-PPA algorithm exhibits high computational precision. However, in the case of uniform distribution, the corresponding values of these two indicators for dADMM are only at the magnitude of $-1$, which indicates poorer recovery performance of dADMM under uniform noise interference. Furthermore, the values of RelErr for P-PPA is consistently lower than that of dADMM, and in most

TABLE 1. The performance of P-PPA and dADMM with random Gaussian matrix.

| | | | P-PPA | | | | | | dADMM | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dim | $\lambda$ | $\beta$ | Iter | Time | Obj | RLNE | ResErr | RelErr | Iter | Time | Obj | RLNE | ResErr | RelErr |
| **Log-normal noise** | | | | | | | | | | | | | | |
| | 0.3 | 0.02 | 3 | 0.0682 | 2.39e+02 | 6.49e-04 | 6.44e-04 | 0.00e+00 | 323 | 0.3358 | 2.39e+02 | 7.03e-04 | 2.63e-04 | 9.87e-06 |
| $200\times500$ | 0.2 | 0.02 | 3 | 0.0402 | 1.60e+02 | 7.94e-04 | 7.07e-04 | 0.00e+00 | 367 | 0.1648 | 1.60e+02 | 4.61e-04 | 1.52e-04 | 9.99e-06 |
| | 0.2 | 0.03 | 3 | 0.0405 | 1.90e+02 | 9.65e-04 | 7.48e-04 | 0.00e+00 | 385 | 0.1678 | 1.90e+02 | 6.47e-04 | 1.21e-04 | 9.92e-06 |
| | 0.4 | 0.06 | 4 | 0.3907 | 5.55e+02 | 9.38e-04 | 8.20e-04 | 0.00e+00 | 351 | 0.7937 | 5.55e+02 | 9.60e-04 | 4.14e-04 | 9.68e-06 |
| $500\times800$ | 0.3 | 0.05 | 4 | 0.3493 | 3.72e+02 | 8.90e-04 | 7.86e-04 | 0.00e+00 | 405 | 0.4617 | 3.72e+02 | 4.98e-04 | 3.04e-04 | 9.63e-06 |
| | 0.35 | 0.055 | 5 | 0.3921 | 4.60e+02 | 8.97e-04 | 6.53e-04 | 0.00e+00 | 405 | 0.4030 | 4.60e+02 | 4.78e-04 | 3.50e-04 | 9.96e-06 |
| | 0.5 | 0.05 | 4 | 0.9609 | 5.98e+02 | 9.68e-04 | 1.32e-03 | 0.00e+00 | 303 | 2.1072 | 5.98e+02 | 6.84e-04 | 9.89e-04 | 9.83e-06 |
| $800\times1100$ | 0.3 | 0.06 | 3 | 0.6599 | 4.02e+02 | 1.10e-03 | 1.24e-03 | 0.00e+00 | 456 | 3.4826 | 4.01e+02 | 1.14e-03 | 8.22e-04 | 9.94e-06 |
| | 0.1 | 0.08 | 3 | 0.7989 | 1.64e+02 | 1.25e-03 | 1.28e-03 | 0.00e+00 | 455 | 3.4434 | 1.62e+02 | 2.25e-03 | 2.34e-04 | 9.97e-06 |
| **Gaussian noise** | | | | | | | | | | | | | | |
| | 0.04 | 0.01 | 15 | 0.3115 | 2.59e+01 | 1.88e-04 | 1.69e-04 | 0.00e+00 | 835 | 0.8726 | 2.59e+01 | 2.68e-04 | 6.98e-05 | 9.65e-06 |
| $200\times500$ | 0.06 | 0.001 | 30 | 0.6463 | 3.07e+01 | 2.07e-04 | 2.49e-04 | 2.71e-07 | 688 | 0.6726 | 3.07e+01 | 1.81e-04 | 1.16e-04 | 9.64e-06 |
| | 0.05 | 0.001 | 16 | 0.367 | 2.56e+01 | 2.15e-04 | 2.17e-04 | 0.00e+00 | 704 | 0.6811 | 2.56e+01 | 2.41e-04 | 7.21e-05 | 9.67e-06 |
| | 0.02 | 0.05 | 10 | 0.632 | 2.48e+01 | 3.21e-04 | 2.81e-04 | 0.00e+00 | 833 | 2.0274 | 2.48e+01 | 4.60e-04 | 8.06e-05 | 9.77e-06 |
| $500\times800$ | 0.02 | 0.06 | 11 | 0.7959 | 2.77e+01 | 3.56e-04 | 2.84e-04 | 0.00e+00 | 889 | 2.2063 | 2.77e+01 | 5.00e-04 | 6.89e-05 | 9.56e-06 |
| | 0.02 | 0.07 | 12 | 0.8641 | 3.07e+01 | 5.40e-04 | 2.72e-04 | 0.00e+00 | 945 | 2.3762 | 3.07e+01 | 5.39e-04 | 6.42e-05 | 9.92e-06 |
| | 0.05 | 0.005 | 12 | 2.5118 | 2.81e+01 | 2.85e-04 | 4.53e-04 | 3.87e-08 | 510 | 3.8881 | 2.81e+01 | 1.34e-04 | 3.36e-04 | 8.68e-06 |
| $800\times1100$ | 0.04 | 0.01 | 12 | 2.9009 | 2.53e+01 | 2.47e-04 | 4.28e-04 | 3.97e-06 | 503 | 3.8336 | 2.53e+01 | 2.07e-04 | 3.16e-04 | 9.22e-06 |
| | 0.02 | 0.04 | 14 | 4.3229 | 2.11e+01 | 3.01e-04 | 3.11e-04 | 2.12e-07 | 623 | 4.7185 | 2.11e+01 | 3.72e-04 | 1.67e-04 | 9.80e-06 |
| **Uniformly distributed noise** | | | | | | | | | | | | | | |
| | 0.001 | 0.03 | 25 | 0.7280 | 9.48e-01 | 1.71e-04 | 1.31e-04 | 0.00e+00 | 7000 | 5.1968 | 1.48e+00 | 5.26e-01 | 9.53e-03 | 4.21e-05 |
| $200\times500$ | 0.001 | 0.04 | 24 | 0.8543 | 1.10e+00 | 2.43e-04 | 1.16e-04 | 0.00e+00 | 7000 | 5.1693 | 1.58e+00 | 5.31e-01 | 1.69e-02 | 4.23e-05 |
| | 0.001 | 0.02 | 24 | 1.0687 | 9.23e-01 | 1.50e-02 | 9.37e-03 | 0.00e+00 | 7000 | 6.3822 | 1.38e+00 | 5.21e-01 | 2.54e-03 | 4.18e-05 |
| | 0.001 | 0.03 | 29 | 11.8960 | 9.40e-01 | 1.16e-04 | 1.63e-04 | 0.00e+00 | 7000 | 24.1811 | 1.99e+00 | 4.42e-01 | 2.21e-01 | 5.22e-05 |
| $500\times800$ | 0.001 | 0.02 | 29 | 12.1170 | 7.91e-01 | 1.56e-04 | 1.74e-04 | 0.00e+00 | 7000 | 22.2927 | 1.89e+00 | 4.32e-01 | 2.14e-01 | 5.27e-05 |
| | 0.001 | 0.06 | 34 | 9.8568 | 1.39e+00 | 2.28e-04 | 1.94e-04 | 0.00e+00 | 7000 | 22.3105 | 2.27e+00 | 4.69e-01 | 2.43e-01 | 4.83e-05 |
| | 0.001 | 0.1 | 34 | 33.1954 | 1.90e+00 | 1.93e-04 | 2.30e-04 | 0.00e+00 | 7000 | 53.3976 | 2.58e+00 | 5.33e-01 | 3.69e-01 | 4.82e-05 |
| $800\times1100$ | 0.001 | 0.09 | 33 | 31.9735 | 1.76e+00 | 2.09e-04 | 2.33e-04 | 0.00e+00 | 7000 | 53.1046 | 2.51e+00 | 5.25e-01 | 3.61e-01 | 4.93e-05 |
| | 0.001 | 0.05 | 33 | 38.5035 | 1.20e+00 | 2.27e-04 | 2.75e-04 | 0.00e+00 | 7000 | 53.3089 | 2.24e+00 | 4.92e-01 | 3.32e-01 | 5.43e-05 |

cases, it even approaches to 0. This indicates that the iteration sequence of P-PPA has reached a stable stage where further descent is no longer observed.

Overall, P-PPA consistently achieves satisfactory and accurate recovery within the maximum iteration steps, while dADMM fails to do so in most cases, especially under uniform distribution (i.e., $p=\infty$). The extensive numerical experiments fully demonstrate the applicability, effectiveness and precision of the P-PPA algorithm in high-dimensional sparse linear regression problems.

4.3.2. *Real examples.* In this section, we perform numerical comparisons using the instances "mpg", "housing", and "bodyfat" from the LIBSVM dataset, which are commonly used for regression. These datasets can be obtained from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets. To accommodate the high-dimensional setting of this paper, we follow the approach in [16, 18] to expand the original features by using polynomial basis functions. For example, the suffix "7" in "mpg7" indicates the use of a 7th-degree polynomial to generate the basis functions. This naming convention is also employed in "housing4" and "bodyfat4". It should be noted that the termination condition is modified to "$RelErr \leq 1e-4$" in the real examples, while other parameters remain consistent with the simulated examples. The computational results of P-PPA and dADMM under different noise interferences are given in Table 3. Since the true sparse coefficients are not known in advance for the real examples, the calculation results of RLNE are not provided in Table 3.

From the computational results of the real examples, it can be observed that P-PPA always efficiently solves all instances to achieve the desired accuracy, and sometimes the values of RelErr

TABLE 2. The performance of P-PPA and dADMM with random partial DCT matrix.

| | | | P-PPA | | | | | | dADMM | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dim | $\lambda$ | $\beta$ | Iter | Time | Obj | RLNE | ResErr | RelErr | Iter | Time | Obj | RLNE | ResErr | RelErr |
| Log-normal noise | | | | | | | | | | | | | | |
| | 0.25 | 0.05 | 5 | 0.0639 | 3.10e+02 | 8.97e-02 | 1.87e-04 | 0.00e+00 | 630 | 0.1649 | 3.10e+02 | 9.12e-02 | 9.26e-06 | 9.93e-06 |
| $200\times500$ | 0.35 | 0.045 | 9 | 0.1163 | 4.10e+02 | 5.29e-02 | 4.76e-03 | 0.00e+00 | 513 | 0.1254 | 4.10e+02 | 4.51e-02 | 1.10e-05 | 9.95e-06 |
| | 0.35 | 0.05 | 8 | 0.1024 | 4.33e+02 | 9.61e-02 | 1.31e-02 | 0.00e+00 | 509 | 0.1323 | 4.33e+02 | 9.04e-02 | 7.33e-06 | 9.80e-06 |
| | 0.085 | 0.15 | 4 | 0.3494 | 2.43e+02 | 8.54e-02 | 9.82e-06 | 0.00e+00 | 566 | 0.5221 | 2.43e+02 | 8.73e-02 | 2.64e-06 | 9.96e-06 |
| $500\times800$ | 0.09 | 0.15 | 4 | 0.36 | 2.57e+02 | 8.54e-02 | 5.36e-06 | 0.00e+00 | 540 | 0.4681 | 2.57e+02 | 8.72e-02 | 2.58e-06 | 9.98e-06 |
| | 0.1 | 0.15 | 4 | 0.3263 | 2.86e+02 | 8.54e-02 | 1.85e-06 | 0.00e+00 | 496 | 0.4081 | 2.86e+02 | 8.70e-02 | 3.31e-06 | 9.94e-06 |
| | 0.1 | 0.3 | 4 | 0.6224 | 4.57e+02 | 1.79e-01 | 1.50e-06 | 0.00e+00 | 293 | 0.7837 | 4.57e+02 | 1.79e-01 | 2.80e-06 | 9.87e-06 |
| $800\times1100$ | 0.09 | 0.3 | 4 | 0.57 | 4.11e+02 | 1.79e-01 | 2.20e-06 | 0.00e+00 | 318 | 0.859 | 4.11e+02 | 1.80e-01 | 2.69e-06 | 9.98e-06 |
| | 0.08 | 0.3 | 4 | 0.5812 | 3.66e+02 | 1.79e-01 | 7.32e-07 | 0.00e+00 | 350 | 0.9904 | 3.66e+02 | 1.80e-01 | 1.86e-06 | 9.90e-06 |
| Gaussian noise | | | | | | | | | | | | | | |
| | 0.06 | 0.01 | 13 | 0.2072 | 3.88e+01 | 1.29e-04 | 1.11e-04 | 0.00e+00 | 675 | 0.4561 | 3.88e+01 | 1.15e-04 | 4.72e-05 | 9.74e-06 |
| $200\times500$ | 0.1 | 0.001 | 16 | 0.2915 | 5.12e+01 | 1.35e-04 | 1.73e-04 | 9.87e-09 | 444 | 0.3286 | 5.12e+01 | 9.12e-05 | 5.55e-05 | 9.64e-06 |
| | 0.09 | 0.005 | 16 | 0.2634 | 5.15e+01 | 1.37e-04 | 1.67e-04 | 6.05e-07 | 475 | 0.3431 | 5.15e+01 | 1.03e-04 | 5.28e-05 | 9.20e-06 |
| | 0.02 | 0.08 | 12 | 0.5906 | 3.37e+01 | 2.89e-04 | 1.19e-04 | 0.00e+00 | 820 | 0.7894 | 3.37e+01 | 1.31e-04 | 2.15e-05 | 9.33e-06 |
| $500\times800$ | 0.02 | 0.09 | 13 | 0.6118 | 3.67e+01 | 3.02e-04 | 1.15e-04 | 0.00e+00 | 887 | 0.8195 | 3.67e+01 | 1.22e-04 | 1.87e-05 | 9.93e-06 |
| | 0.025 | 0.07 | 13 | 0.5984 | 3.84e+01 | 1.98e-04 | 1.11e-04 | 0.00e+00 | 619 | 0.6281 | 3.84e+01 | 1.34e-04 | 3.46e-05 | 9.56e-06 |
| | 0.02 | 0.08 | 12 | 2.0702 | 3.23e+01 | 2.42e-04 | 1.34e-04 | 0.00e+00 | 531 | 2.5491 | 3.23e+01 | 2.21e-04 | 3.87e-05 | 9.64e-06 |
| $800\times1100$ | 0.01 | 0.07 | 11 | 2.5686 | 1.48e+01 | 2.51e-04 | 1.45e-04 | 0.00e+00 | 960 | 4.5532 | 1.48e+01 | 2.22e-04 | 2.21e-05 | 9.28e-06 |
| | 0.02 | 0.07 | 12 | 2.1154 | 2.95e+01 | 2.60e-04 | 1.38e-04 | 0.00e+00 | 516 | 2.5243 | 2.95e+01 | 2.17e-04 | 3.62e-05 | 9.61e-06 |
| Uniformly distributed noise | | | | | | | | | | | | | | |
| | 0.002 | 0.04 | 29 | 1.0794 | 2.19e+00 | 3.96e-04 | 1.13e-04 | 0.00e+00 | 7000 | 4.2859 | 2.63e+00 | 4.79e-01 | 6.68e-07 | 2.85e-05 |
| $200\times500$ | 0.003 | 0.04 | 30 | 1.1406 | 3.29e+00 | 3.35e-04 | 9.06e-05 | 0.00e+00 | 7000 | 4.7184 | 3.71e+00 | 3.85e-01 | 6.78e-07 | 3.18e-05 |
| | 0.001 | 0.04 | 28 | 1.0192 | 1.10e+00 | 5.09e-04 | 7.77e-05 | 0.00e+00 | 7000 | 4.1154 | 1.47e+00 | 6.08e-01 | 1.52e-07 | 2.25e-05 |
| | 0.005 | 0.04 | 39 | 12.7576 | 5.44e+00 | 1.81e-04 | 1.81e-04 | 0.00e+00 | 7000 | 19.2919 | 6.55e+00 | 3.20e-01 | 3.03e-01 | 4.81e-05 |
| $500\times800$ | 0.001 | 0.1 | 35 | 14.1634 | 1.98e+00 | 2.11e-04 | 1.22e-04 | 0.00e+00 | 7000 | 20.9787 | 3.14e+00 | 3.68e-01 | 2.03e-01 | 5.16e-05 |
| | 0.01 | 0.001 | 43 | 19.3941 | 5.33e+00 | 2.13e-02 | 2.08e-02 | 0.00e+00 | 7000 | 19.8943 | 6.48e+00 | 2.05e-01 | 2.01e-01 | 6.47e-05 |
| | 0.01 | 0.005 | 41 | 45.6143 | 5.62e+00 | 5.67e-05 | 1.33e-04 | 0.00e+00 | 7000 | 54.9824 | 7.28e+00 | 3.64e-01 | 3.61e-01 | 5.91e-05 |
| $800\times1100$ | 0.005 | 0.01 | 36 | 44.4422 | 3.16e+00 | 6.57e-05 | 1.32e-04 | 0.00e+00 | 7000 | 58.7052 | 4.79e+00 | 2.90e-01 | 2.84e-01 | 6.09e-05 |
| | 0.01 | 0.01 | 41 | 43.3003 | 6.32e+00 | 8.68e-05 | 1.47e-04 | 0.00e+00 | 7000 | 58.7794 | 7.77e+00 | 4.03e-01 | 3.99e-01 | 5.61e-05 |

TABLE 3. The performance of P-PPA and dADMM with real data.

| | | P-PPA | | | | dADMM | | | |
|---|---|---|---|---|---|---|---|---|---|
| Data name | M,N | Iter | Time | ResErr | RelErr | Iter | Time | ResErr | RelErr |
| Log-normal noise | | | | | | | | | |
| mpg7 | 392,3432 | 19 | 3.8149 | 9.97e-02 | 0.00e+00 | 7000 | 1028.4981 | 2.34e-01 | 1.71e-02 |
| housing4 | 506,2380 | 67 | 15.2100 | 2.63e-02 | 0.00e+00 | 629 | 36.4358 | 1.20e-01 | 9.10e-05 |
| bodyfat4 | 252,3060 | 5 | 0.3657 | 4.15e-03 | 0.00e+00 | 338 | 37.7413 | 2.97e-03 | 9.93e-05 |
| Gaussian noise | | | | | | | | | |
| mpg7 | 392,3432 | 25 | 49.0618 | 7.93e-02 | 4.92e-05 | 1913 | 218.1492 | 9.87e-02 | 9.96e-05 |
| housing4 | 506,2380 | 26 | 39.3100 | 7.81e-02 | 5.08e-05 | 4689 | 327.9067 | 9.72e-01 | 9.68e-05 |
| bodyfat4 | 252,3060 | 6 | 4.1721 | 2.68e-03 | 4.40e-07 | 558 | 42.5721 | 7.91e-01 | 9.74e-05 |
| Uniformly distributed noise | | | | | | | | | |
| mpg7 | 392,3432 | 65 | 34.3475 | 4.32e-01 | 5.68e-05 | 7000 | 1035.6185 | 5.14e-01 | 2.00e-02 |
| housing4 | 506,2380 | 48 | 29.0035 | 3.25e-01 | 3.97e-05 | 7000 | 495.0260 | 7.80e-01 | 1.05e-02 |
| bodyfat4 | 252,3060 | 57 | 35.5436 | 7.32e-03 | 0.00e+00 | 103 | 11.3445 | 4.32e-03 | 7.55e-05 |

are even close to 0. In contrast, dADMM only satisfies the termination condition in a few cases. More specifically, P-PPA consistently requires fewer iteration steps and less computation time than dADMM to achieve higher computational accuracy. Based on the above analysis, it can be concluded that the P-PPA algorithm possesses better robustness, accuracy, and computational efficiency than dADMM.

## 5. CONCLUSIONS

This paper focuses on the high-dimensional sparse linear regression problem with $\ell_p$-norm data fidelity and elastic net regularization. Based on the second-order information, an efficient and stable P-PPA algorithm is proposed, where the subproblem is solved by superlinear or even quadratically convergent semismooth Newton method from the dual perspective. This key step significantly reduces the computational cost of the proposed algorithm. The global convergence of the algorithm is theoretically analyzed. Numerical experiments based on extensive simulated and real examples fully demonstrate the numerical advantages of the proposed P-PPA algorithm over the first-order ADMM in terms of robustness and computational efficiency. In conclusion, the P-PPA algorithm proposed in this paper is highly suitable for high-dimensional sparse linear regression problems.

**Acknowledgements**
The work of Y. Xiao was supported by National Natural Science Foundation of China (Grant No. 11971149 and 12271217), and National Natural Science Foundation of Henan Province (Grant No. 232300421018). The work of P. Li was supported by National Natural Science Foundation of China (Grant No. 12301420).

## REFERENCES

[1] P.C. Bellec, G. Lecué, A.B. Tsybakov, Slope meets lasso: improved oracle bounds and optimality, The Annals of Statistics, 2 46 (2018), 3603-3642.
[2] A. Belloni, V. Chernozhukov, $\ell_1$-penalized quantile regression in high-dimensional sparse models, The Annals of Statistics, 39 (2011), 82-130.
[3] A. Belloni, V. Chernozhukov, L. Wang, Square-root lasso: pivotal recovery of sparse signals via conic programming, Biometrika, 98 (2011), 791-806.
[4] F. Bunea, Y. She, H. Ombao, et al., Penalized least squares regression methods and applications to neuroimaging, Neuroimage, 55 (2011), 1519-1527.
[5] E. Candes, T. Tao, Decoding by linear programming. IEEE Transactions on Information Theory, 51 (2005), 4203-4215.
[6] S. Cho, H. Kim, S. Oh, et al. Elastic-net regularization approaches for genome-wide association studies of rheumatoid arthritis, BMC proceedings. BioMed Central, 3 (2009): 1-6.
[7] F. Clarke, Optimization and Nonsmooth Analysis, Wiley, 1983.
[8] Y. Cui, J. Pang, B. Sen, Composite difference-max programs for modern statistical estimation problems, SIAM Journal on Optimization, 28 (2018), 3344-3374.
[9] M. Demirer, F. Diebold, L. Liu, et al., Estimating global bank network connectedness, Journal of Applied Econometrics, 33 (2018), 1-15.
[10] Y. Ding, Z. Yue, H. Zhang, An adaptive $\ell_1$-$\ell_2$-type model with hierarchies for sparse signal reconstruction problem, Pacific Journal of Optimization, 18 (2022), 695-712.
[11] Y. Ding, H. Zhang, P. Li, et al., An efficient semismooth Newton method for adaptive sparse signal recovery problems, Optimization Methods and Software, 38 (2023), 262-288.
[12] D. Donoho, Compressed sensing. IEEE Transactions on Information Theory, 52 (2006), 1289-1306.
[13] D. Donoho, M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell_1$ minimization, Proceedings of the National Academy of Sciences, 100 (2003), 2197-2202.
[14] R. Gribonval, M. Nielsen, Sparse representations in unions of bases, IEEE Transactions on Information Theory, 49 (2003), 3320-3325.
[15] J. Hiriart-Urruty, C. Lemaréchal, Convex Analysis and Minimization Algorithms I: Fundamentals, Springer 2013.

[16] L. Huang, J. Jia, B. Yu, et al. Predicting execution time of computer programs using sparse polynomial regression, Advances in Neural Information Processing Systems, 23 (2010), 883-891.
[17] C. Lemaréchal, C. Sagastizábal, Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries, SIAM journal on optimization, 1997, 7(2): 367-385.
[18] P. Li, M. Liu, Z. Yu, A global two-stage algorithm for non-convex penalized high-dimensional linear regression problems, Computational Statistics, 38 (2023), 871-898.
[19] X. Li, D. Sun, K.-C. Toh, A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems, SIAM Journal on Optimization, 28 (2018): 433-458.
[20] X. Li, D. Sun, K.-C. Toh, An asymptotically superlinearly convergent semismooth Newton augmented Lagrangian method for linear programming, SIAM Journal on Optimization, 30 (2020), 2410-2440.
[21] M. Lin , D. Sun, K.-C. Toh, Efficient algorithms for multivariate shape-constrained convex regression problems, arXiv preprint arXiv:2002.11410, 2020.
[22] Z. Lu, Iterative reweighted minimization methods for $l_p$ regularized unconstrained nonlinear programming, Mathematical Programming, 147 (2014), 277-307.
[23] R. Mifflin, Semismooth and semiconvex functions in constrained optimization, SIAM Journal on Control and Optimization, 15 (1977), 959-972.
[24] J. Moreau, Proximité et dualité dans un espace hilbertien, Bulletin de la société mathématique de France, 93 (1965), 273-299.
[25] L. Qi, J. Sun, A nonsmooth version of Newton's method, Mathematical Programming, 58 (1993), 353-367.
[26] R. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, Mathematics of Operations Research, 1 (1976), 97-116.
[27] R. Rockafellar, Monotone operators and the proximal point algorithm, SIAM Journal on Control and Optimization, 14 (1976)m 877-898.
[28] R. Rockafellar, Convex Analysis, Princeton University Press, Princeton, 2015.
[29] R. Rockafellar, R. Wets, Variational Analysis, Springer, New York, 1998.
[30] P. Tang, C. Wang, D. Sun, et al., A sparse semismooth Newton based proximal majorization-minimization algorithm for nonconvex square-root-loss regression problems, The Journal of Machine Learning Research, 21 (2020), 9253-9290.
[31] S. Teipel, M. Grothe, C. Metzger, et al. Robust detection of impaired resting state functional connectivity networks in Alzheimer's disease using elastic net regularized regression, Frontiers in Aging Neuroscience, 8 (2017), 318.
[32] L. Wang, The $L_1$ penalized LAD estimator for high dimensional linear regression, Journal of Multivariate Analysis, 120 (2013), 135-151.
[33] X. Xiu, L. Kong, Y. Li, et al., Iterative reweighted methods for $\ell_1$-$\ell_p$ minimization, Computational Optimization and Applications, 70 (2018), 201-219.
[34] Y. Xue, Y. Feng, C. Wu, An efficient and globally convergent algorithm for $\ell_{p,q}$-$\ell_r$ model in group sparse optimization, Communications in Mathematical Sciences, 18 (2020), 227-258.
[35] K. Yosida, Functional Analysis, Springer, Berlin, 1964.
[36] Z. Zhang, W. Wei, Primal-Dual approach for uniform noise removal,. First International Conference on Information Science and Electronic Technology (ISET 2015), pp. 103-106, Atlantis Press, 2015
[37] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society Series B: Statistical Methodology, 67 (2005), 301-320.