Fast projection onto the ordered weighted $\ell_1$ norm ball

Qinzhen Li$^{1}$ & Xudong Li$^{1,2,*}$

$^1$School of Data Science, Fudan University, Shanghai 200433, China; $^2$Shanghai Center for Mathematical Sciences, Fudan University, Shanghai 200433, China

Email: 18210980031@fudan.edu.cn, lixudong@fudan.edu.cn

Received February 12, 2020; accepted June 16, 2020; published online July 29, 2021

Abstract In this paper, we provide a finitely terminated yet efficient approach to compute the Euclidean projection onto the ordered weighted $\ell_1$ (OWL1) norm ball. In particular, an efficient semismooth Newton method is proposed for solving the dual of a reformulation of the original projection problem. Global and local quadratic convergence results, as well as the finite termination property, of the algorithm are proved. Numerical comparisons with the two best-known methods demonstrate the efficiency of our method. In addition, we derive the generalized Jacobian of the studied projector which, we believe, is crucial for the future designing of fast second-order nonsmooth methods for solving general OWL1 norm constrained problems.

Keywords Euclidean projector, ordered weighted $\ell_1$ norm ball, HS-Jacobian, semismooth Newton method

MSC(2020) 90C06, 90C20, 90C25

Citation: Li Q Z, Li X D. Fast projection onto the ordered weighted $\ell_1$ norm ball. Sci China Math, 2022, 65: 869–886, https://doi.org/10.1007/s11425-020-1743-9

1 Introduction

Given a nonzero vector $\lambda \in \mathbb{R}^n$ satisfying $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$, the ordered weighted $\ell_1$ (OWL1) norm $\kappa_\lambda$ is defined as

$$\kappa_\lambda(x) = \langle |x|^\lambda, \lambda \rangle = \sum_{i=1}^n \lambda_i |x_i|^\lambda, \quad \forall x \in \mathbb{R}^n,$$

where $|x| \in \mathbb{R}^n$ denotes the vector whose $i$-th entry is $|x_i|$ and for any given vector $z \in \mathbb{R}^n$, $z^\downarrow$ denotes the vector of entries of $z$ being arranged in the non-increasing order $z_1^\downarrow \geq \cdots \geq z_n^\downarrow$. It can be shown that $\kappa_\lambda$ is a norm function if and only if $\lambda_1 > 0$ (see [5, Exercise IV.1.19]). In fact, both the $\ell_1$ norm and the $\ell_\infty$ norm can be obtained from $\kappa_\lambda$ via choosing different weighted vectors $\lambda$. Given $\tau > 0$, the $\tau$ level-set of $\kappa_\lambda$ is defined as $\Delta_\tau := \{ x \in \mathbb{R}^n | \kappa_\lambda(x) \leq \tau \}$. In this paper, we focus on the problem of projecting a vector $b$ onto $\Delta_\lambda$, i.e., solving the following optimization problem:

$$\Pi_{\Delta_\tau}(b) := \arg\min_x \left\{ \frac{1}{2} \| x - b \|^2 \right\} \kappa_\lambda(x) \leq \tau \right\}. \quad (1.1)$$

*Corresponding author
Obviously, if \( \kappa_\lambda(b) \leq \tau \), we have trivially that \( \Pi_{\Delta,}(b) = b \). To avoid this special case, throughout this paper, we make the blanket assumption that \( \kappa_\lambda(b) > \tau \).

Recently, the OWL1 norm function \( \kappa_\lambda \) receives significant academic attention in both optimization and statistical communities. Coined as the weighted Ky Fan norm function, \( \kappa_\lambda \) appeared in the optimization literature [42], where the first- and second-order properties of the Moreau-Yosida regularization of \( \kappa_\lambda \) were carefully studied. In [42], Wu et al. also emphasized the important roles played by the class of Ky Fan norm functions in matrix optimization problems. Later, as a generalization of the octagonal shrinkage and clustering algorithm for regression (OSCAR) regularizer, the OWL1 norm was studied as a group sparsity promoted regularizer in [43]. Meanwhile, an OWL1 norm regularized least square model, termed as the sorted L-one penalized estimation (SLOPE), was proposed in [6]. Bogdan et al. [6] showed that the SLOPE estimator enjoys nice statistical properties. In particular, it has a controllable false discovery rate, a measure of the Type-I error for multiple testing. Recently, Su and Candès [38] further argued that under certain mild assumptions, the SLOPE estimator achieves asymptotic minimaxity over large sparsity classes for Gaussian designs. Hence, as a fundamental subroutine for solving the OWL1 norm constrained estimator which is a provable statistically well-behaved estimator, the computation of the projector \( \Pi_{\Delta,} \) in (1.1) deserves more research efforts.

As an important regularizer, the computation of the proximal mapping of \( \kappa_\lambda \), defined as follows:

\[
\text{Prox}_{\kappa_\lambda}(b) := \arg\min_x \left\{ \kappa_\lambda(x) + \frac{1}{2}\|x - b\|^2 \right\}, \quad \forall b \in \mathbb{R}^n,
\]

was studied in [6, 43, 44]. These studies revealed that the computation of \( \text{Prox}_{\kappa_\lambda} \) can be reduced to the Euclidean projection problem over the monotone nonnegative cone\(^1\) and can be efficiently solved by the famous pool adjacent violators algorithm (PAVA) [2, 4, 17]. With this effective subroutine, the accelerated proximal gradient method [25] was used to solve the SLOPE model in [6]. Recently, the computational efficiency for the SLOPE model, especially for large-scale difficult problems, was greatly enhanced by Luo et al. [22] in which the authors carefully exploited the special low-rank and sparsity structures in the generalized Jacobian of \( \text{Prox}_{\kappa_\lambda} \) and designed a highly efficient sparse semismooth Newton based augmented Lagrangian method.

Unlike the proximal mapping \( \text{Prox}_{\kappa_\lambda} \), the Euclidean projector \( \Pi_{\Delta,} \) is less understood. In fact, given the discussions in the literature of classic level-set methods [1], one could argue that the computations of \( \text{Prox}_{\kappa_\lambda} \) and \( \Pi_{\Delta,} \) are mathematically equivalent. However, as far as we know, there is no closed-form expression in the literature to explicitly characterize this equivalence. Nevertheless, this equivalence can be used to design algorithms to compute \( \Pi_{\Delta,} \) from \( \text{Prox}_{\kappa_\lambda} \). For example, in [44], Zeng and Figueiredo advocated a root-finding iterative scheme based on the equivalence. Later, Davis [10] noticed that the root-finding scheme cannot obtain \( \Pi_{\Delta,} \) in a finite number of steps and proposed a finitely terminated approach for computing the projector. By partitioning non-decreasing entries of the preprocessed input into groups, Davis’ approach conducts a series of tests, and partial sortings and averagings. However, as shown in Section 5, it can be time-consuming in solving high-dimensional problems. We also note that neither [44] nor [10] studied the generalized Jacobian of \( \Pi_{\Delta,} \), which can be essential for the algorithmic design for solving general OWL1 norm constrained optimization problems as demonstrated in [19–22, 45].

In this paper, we aim to design a finitely terminated yet efficient algorithm for the large-scale projection problem (1.1). Our proposed algorithm starts with an observation that the problem (1.1) can be reduced to the problem of finding the projector onto the intersection of the monotone nonnegative cone and an affine subspace. Building upon the HanSun (HS)-Jacobian [14, 20] of the projector onto the monotone nonnegative cone, we are able to design an efficient semismooth Newton (SSN) method to solve the dual of the reduced problem. The global and local quadratic convergence results as well as the finite termination properties of the SSN are provided. Numerical experiments demonstrate that our algorithm is faster than the root-finding approach and outperforms Davis’ approach by a large margin. In addition to these, we conduct a careful first-order variational analysis on \( \Pi_{\Delta,} \) and construct its generalized Jacobian which

\(^1\) See Subsection 3.1 for the definition.
lays the foundation for the subsequent designing of second-order nonsmooth methods for solving general OWL1 norm constrained problems.

The rest of this paper is organized as follows. In Section 2, we review some definitions and results which will be used later for designing and analyzing our algorithm. In Section 3, we recast the original problem as the problem of projecting onto the intersection of the monotone nonnegative cone and the affine subspace and derive the dual of the reformulated problem. Then a semismooth Newton method is proposed for this dual problem. Theoretical convergence results as well as the finite termination property of our algorithm are also provided in this section. Section 4 is dedicated to the generalized Jacobian of $\Pi_{\Delta}$. In Section 5, we conduct numerical experiments to evaluate the performance of our algorithms against the root-finding and Davis’ approaches. We conclude our paper in Section 6.

Before moving to the next section, we list here some notation to be used later in this paper. For any given vector $y \in \mathbb{R}^n$, we use $\text{Diag}(y)$ to denote the diagonal matrix whose $i$-th diagonal element is $y_i$, and $e_n$ to denote vectors of all ones in $\mathbb{R}^n$. We denote by $O_n$ and $I_n$ the $n \times n$ zero matrix and identity matrix, respectively. For any given matrix $A \in \mathbb{R}^{m \times n}$, we denote by $\text{Null}(A)$ the null space of $A$.

## 2 Preliminaries

We briefly recall in this section some basic definitions and results associated with the set-valued mappings and the semismoothness of vector-valued functions.

Let $K : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a set-valued mapping. The domain of $K$, denoted by $\text{dom} \, K$, is the set

$$\text{dom} \, K \equiv \{ x \in \mathbb{R}^n \mid K(x) \neq \emptyset \}.$$  

The set-valued mapping $K$ is said to be nonempty valued at a given point $p$ if $p \in \text{dom} \, K$. We also recall the following definitions associated with $K$.

**Definition 2.1** (See [13, Definition 2.1.16]). A set-valued mapping $K : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is said to be

1. closed at the point $\bar{x}$ if there are sequences $\{ x^k \}$ and $\{ y^k \} \subseteq \mathbb{R}^n$ such that $x^k \to \bar{x}$ as $k \to \infty$ and for all $k \geq 0$, $y^k \in K(x^k)$ and $y^k \to y$ as $k \to \infty$, $y \in K(\bar{x})$;
2. bounded at the point $\bar{x}$ if $K(\bar{x})$ is bounded;
3. upper semicontinuous at the point $\bar{x}$ if for every open set $V$ containing $K(\bar{x})$, there exists an open neighborhood $N$ of $\bar{x}$ such that for each $x \in N$, $V$ contains $K(x)$;
4. upper semicontinuous on a set $\Omega \subseteq \mathbb{R}^n$ if $K$ is upper semicontinuous at every point in $\Omega$.

**Definition 2.2** (B-subdifferential and Clarke generalized Jacobian [9]). Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a locally Lipschitz continuous function. Define the B-subdifferential of $F$ at the point $x \in \mathbb{R}^n$ to be

$$\partial_B F(x) = \left\{ V \in \mathbb{R}^{n \times n} \mid V = \lim_{x^k \to x} JF(x^k), \, x^k \in \Omega_F \right\},$$

where

$$\Omega_F := \{ x \in \mathbb{R}^n \mid F\text{ is differentiable at } x \}$$

and define the Clarke generalized Jacobian of $F$ at the point $x$ as

$$\partial F(x) = \text{conv} \partial_B F(x),$$

i.e., the convex hull of $\partial_B F(x)$.

By [13, Propositions 7.1.4 and 7.4.11], we see that both the Clarke generalized Jacobian $\partial F$ and the B-subdifferential $\partial_B F$ are nonempty and compact valued, upper-semicontinuous set-valued mappings.

Inspired by the definitions given in [18, 24, 31, 41], we can state the following definition of semismooth functions with respect to any given nonempty and compact valued, upper-semicontinuous set-valued mapping. In fact, this definition has been used in recent papers for solving various high-dimensional statistical optimization problems [19, 21, 45].
Let $O \subseteq \mathbb{R}^n$ be an open set, $K : O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$ be a nonempty and compact valued, upper-semicontinuous set-valued mapping and $F : O \rightarrow \mathbb{R}^n$ be a locally Lipschitz continuous function. $F$ is said to be semismooth at $x \in O$ with respect to the multifunction $K$ if $F$ is directionally differentiable at $x$ and for any $V \in K(x + \Delta x)$ with $\Delta x \rightarrow 0$,

$$F(x + \Delta x) - F(x) - V\Delta x = o(\|\Delta x\|).$$

Let $\gamma$ be a positive constant. $F$ is said to be $\gamma$-order (strongly, if $\gamma = 1$) semismooth at $x$ with respect to $K$ if $F$ is directionally differentiable at $x$ and for any $V \in K(x + \Delta x)$ with $\Delta x \rightarrow 0$,

$$F(x + \Delta x) - F(x) - V\Delta x = O(\|\Delta x\|^1+\gamma).$$

$F$ is said to be a semismooth ($\gamma$-order semismooth, strongly semismooth, respectively) function on $O$ with respect to $K$ if it is semismooth ($\gamma$-order semismooth, strongly semismooth, respectively) everywhere in $O$ with respect to $K$.

Semismooth functions and the corresponding algorithms have been extensively studied and used in the nonsmooth optimization/equation community. Indeed, it is well known that continuous piecewise affine functions and twice continuously differentiable functions are all strongly semismooth everywhere. Please refer to [13] for more examples of semismooth functions. The following second-order limit result is in fact an extension of [28, Theorem 2.1] and has been proved and used in [19]. We will also use it to prove the convergence of our algorithm.

**Proposition 2.4.** Let $\theta : \Omega \rightarrow \mathbb{R}$ be a continuously differentiable function and its gradient $\nabla \theta : \Omega \rightarrow \mathbb{R}^n$ is locally Lipschitz on the open set $\Omega$. If $\nabla \theta$ is semismooth at a point $x \in \Omega$ with respect to a nonempty, compact valued and upper-semicontinuous multifunction $K : \Omega \cong \mathcal{S}^n$, then for any $V \in K(x + d)$ with $d \rightarrow 0$, we have

$$\theta(x + d) - \theta(x) - \langle \nabla \theta(x), d \rangle - \frac{1}{2}(d, Vd) = o(\|d\|^2).$$

## 3 A dual semismooth Newton algorithm for solving (1.1)

In this section, by some well-known reduction techniques, we are able to represent the OWL1 norm constraint in a more explicit form and to transform the original problem (1.1) into a handy reformulation. Then we propose a semismooth Newton method to efficiently solve the reformulated projection problem.

### 3.1 A reformulation of the problem (1.1)

We first focus on the reformulation of the original problem (1.1). Let $\Pi_n^s$ be the set of all the signed permutation matrices in $\mathbb{R}^{n \times n}$. For any given vector $u \in \mathbb{R}^n$, denote by

$$\Pi_n^s(u) := \{P \in \Pi_n^s \mid Pu = |u|^1\}$$

(3.1)

the set of signed permutation matrices associated with $u$. Let $B \in \mathbb{R}^{n \times n}$ be the matrix defined by

$$Bx = [x_1 - x_2, x_2 - x_3, \ldots, x_{n-1} - x_n, x_n]^T$$

for all $x \in \mathbb{R}^n$ and define the monotone nonnegative cone as

$$C := \{x \in \mathbb{R}^n \mid x_1 \geq \cdots \geq x_n \geq 0\} = \{x \in \mathbb{R}^n \mid Bx \geq 0\}.$$

Given $\lambda \in C$ and $\tau > 0$, define

$$C_\lambda^\tau := \{x \in \mathbb{R}^n \mid \langle \lambda, x \rangle = \tau, x \in C\}$$

and

$$\Pi_{C_\lambda^\tau}(b) := \arg \min_x \left\{\frac{1}{2}\|x - b\|^2 \left| x \in C_\lambda^\tau \right\} \right., \quad b \in \mathbb{R}^n.$$  

(3.2)

The following proposition states that the original projection problem (1.1) can be reformulated in a more explicit form via involving the monotone nonnegative cone $C$. 


Proposition 3.1. For any given \( b \in \mathbb{R}^n \), suppose that \( \kappa_\lambda(b) > \tau \). Then it holds that

\[
\Pi_{\Delta_\tau}(b) = P^T \Pi_{C_\tau}(Pb), \quad \forall P \in \Pi^n(b).
\]

Note that Proposition 3.1 can be proved via simple reduction techniques. In fact, although presented in different ways, it has been proved in [10, 44]. For the compactness of the current paper, we provide a simple proof for this proposition in Appendix A.

Proposition 3.1 implies that we can, without loss of generality, assume in the problem (3.2) that \( b \in C \).

As one can observe later, that our algorithm and analysis in fact work for the problem (3.2) with a general vector \( b \in \mathbb{R}^n \). In order to design efficient algorithms for solving the problem (3.2), we first study its constraints system, i.e.,

\[
\langle \lambda, x \rangle = \tau, \quad x \in C.
\]  

(3.3)

A simple observation shows that the Slater’s condition holds, i.e., there exists \( \hat{x} \in \text{int}(C) \) such that \( \langle \lambda, \hat{x} \rangle = \tau \). For example, one can set \( \hat{x}_i = \tau / \|\lambda\|_1 \) for \( i = 1, \ldots, n \). The following nondegeneracy condition associated with (3.3) stems from the seminar works of Robinson [32–34] and has been extensively studied for semidefinite programming problems [8, 39]. It can be regarded as a generalization of the classic linear independent constraint qualification (LICQ) [32, 37].

Definition 3.2. We say that a feasible solution \( \tilde{x} \) to the system (3.3) is constraint nondegenerate if

\[
\lambda^T \text{lin} T_C(\tilde{x}) = \mathbb{R},
\]

(3.4)

where \( T_C(\tilde{x}) \) denotes the tangent cone of \( C \) at the point \( \tilde{x} \) and \( \text{lin} T_C(\tilde{x}) \) represents the lineality space of \( T_C(\tilde{x}) \).

Proposition 3.3. It holds that any feasible solution to the problem (3.2) is constraint nondegenerate in the sense of Definition 3.2.

Proof. For any feasible solution \( \tilde{x} \) to (3.2), it holds that

\[
T_C(\tilde{x}) = \{ d \in \mathbb{R}^n \mid (Bd)_i \geq 0, \ i \in \mathcal{I}(\tilde{x}) \},
\]

where \( \mathcal{I}(\tilde{x}) \) is the set of active indices given as

\[
\mathcal{I}(\tilde{x}) = \{ i \in \{1, \ldots, n\} \mid (B\tilde{x})_i = 0 \}.
\]

Thus, simple calculations show that

\[
\text{lin} T_C(\tilde{x}) = \{ d \in \mathbb{R}^n \mid (Bd)_i = 0, \ i \in \mathcal{I}(\tilde{x}) \} = \text{Null}(B_{\mathcal{I}(\tilde{x})}),
\]

where \( B_{\mathcal{I}(\tilde{x})} \) is the submatrix obtained by extracting rows of \( B \) with indices in \( \mathcal{I}(\tilde{x}) \). We only consider the nontrivial case where \( \mathcal{I}(\tilde{x}) \neq \emptyset \). Since \( \tilde{x} \neq 0 \), we know that \( \mathcal{I}(\tilde{x}) \neq [n] \).

Next, we show that the matrix

\[
\begin{bmatrix}
\lambda^T \\
B_{\mathcal{I}(\tilde{x})}^T
\end{bmatrix}
\]

is of full row rank. Suppose, on the contrary, that there exists a nonzero vector \( \mu \in \mathbb{R}^{\mathcal{I}(\tilde{x})} \) such that

\[
\lambda = \sum_{i \in \mathcal{I}} B_{\mathcal{I}(\tilde{x})}^T \mu_i.
\]

Define

\[
i_0 := \inf\{ i \in [n] \mid i \notin \mathcal{I}(\tilde{x}) \},
\]

i.e., the smallest index which is not in \( \mathcal{I}(\tilde{x}) \). Let \( \tilde{e} \in \mathbb{R}^n \) be the vector whose first \( i_0 \) entries are all ones with the rest entries being zeros. Then we have

\[
0 < \sum_{i=1}^{n-1} \lambda_i = \langle \tilde{e}, \lambda \rangle = \sum_{i \in \mathcal{I}(\tilde{x})} \mu_i \langle \tilde{e}, B_{\mathcal{I}(\tilde{x})}^T \rangle = 0,
\]

which is a contradiction.
where the last equality follows from the fact that $\langle \tilde{e}, B_i^T \rangle = 0$ for all $i \in [n]$. Hence, we arrive at a contradiction.

Now, the above full row rankness implies that for any $u \in \mathbb{R}$, there exists $d \in \mathbb{R}^n$ such that

$$\lambda^T d = u, \quad B_i(d) = 0,$$

i.e.,

$$\lambda^T \text{lin} T_C(\tilde{x}) = \mathbb{R}.$$  

The proof is thus completed. \hfill \Box

Remark 3.4. As can be observed from the proof of Proposition 3.3, the nondegeneracy condition (3.4) is in fact equivalent to the classic linear independent constraint qualification associated with the following system:

$$\langle \lambda, x \rangle = \tau, \quad B x \geq 0.$$  

3.2 A dual semismooth Newton method for (3.2)

We note that the problem (3.2) concerns with computing the Euclidean projector onto the intersection of an affine subspace and a closed convex cone. Various algorithms, including the dual quasi-Newton method [23], the dual semismooth Newton method [30] and the alternating projection method with the Dykstra’s correction [12,16], have been proposed for computing the projector. Extensive numerical comparisons in [30] show that the dual semismooth Newton approach outperforms others significantly. Hence, in this subsection, we focus on designing a semismooth Newton method for solving the dual of (3.2) and study its convergence properties.

The dual of the problem (3.2) in the minimization form can be written as follows:

$$\min_y \left\{ \phi(y) := \frac{1}{2} \| \Pi_C(y \lambda + b) \|^2 - y \tau - \frac{1}{2} \| b \|^2 \right\}.$$  

(3.5)

Since the Slater’s condition associated with the problem (3.2) is satisfied, we know that $\phi$ is coercive, i.e., $\phi(y) \to +\infty$ as $|y| \to +\infty$ (see [36]). Hence, we know that the solution set $\Omega_D \subseteq \mathbb{R}^n$ of the dual problem (3.5) is nonempty. Meanwhile, since $C$ is a closed convex cone, we know that $\| \Pi_C(\cdot) \|^2$ is a continuously differentiable function (see, for example, [35, Theorem 31.5]). Then it is not difficult to see that $\phi$ is continuously differentiable over $\mathbb{R}$ with

$$\phi'(y) = \langle \Pi_C(y \lambda + b), \lambda \rangle - \tau.$$  

(3.6)

Therefore, the optimal solution set $\Omega_D$ of the problem (3.5) is exactly the solution set of the following univariate nonlinear nonsmooth equation:

$$\phi'(y) = 0, \quad y \in \mathbb{R}.$$  

(3.7)

Let $y^* \in \Omega_D$ be a solution to (3.7), i.e., an optimal solution to the problem (3.5). Then we know from the optimality condition that $\Pi_C(y^* \lambda + b)$ is the unique optimal solution to the problem (3.2). Since $C$ is a polyhedral convex set, by [13, Proposition 4.1.4], we know that $\phi'$ is in fact a Lipschitz continuous piecewise affine function on $\mathbb{R}$. Hence, $\phi'$ is semismooth on $\mathbb{R}$ with respect to the classic Clarke generalized Jacobian $\partial \phi'$ and a semismooth Newton method can be used for efficiently solving the above equation.

3.2.1 Computations of $\Pi_C$ and its generalized Jacobian

In order to use the semismooth Newton method to solve (3.7), we first need to compute $\Pi_C$ efficiently. For any $d \in \mathbb{R}^n$, recall from the definition of $C$ that

$$\Pi_C(d) := \arg\min_x \left\{ \frac{1}{2} \| x - d \|^2 \left| B x \geq 0 \right\}.$$  

(3.8)
Due to the special structure of $B$, the problem (3.8) has been extensively studied in the optimization and statistical literature. Indeed, the problem (3.8) is closely related to the so-called isotonic regression problem and can be solved by the famous pool adjacent violators algorithm (PAVA) [2, 4, 17]. In this paper, we use a highly efficient implementation of the PAVA provided in [6].

Next, we shall study the first-order variational properties of $\Pi_C$, i.e., obtaining the certain generalized Jacobian associated with $\Pi_C$ which will be used in the semismooth Newton method. As the Euclidean projector over a polyhedral convex set, the first-order variation properties of $\Pi_C$ have been intensively studied in [14, 15, 20, 27, 29]. It is noted in the above references that the computation of the generalized Jacobian has been widely used in the algorithmic design for solving many high-dimensional machine learning problems, for example, the fused least absolute shrinkage and selection operator (LASSO) problems [19], the SLOPE models [22] and the clustered LASSO problems [21]. The following definition of the HS-Jacobian associated with $\Pi_C$ is based on the study conducted in [14, 20]. Consider the set-valued mapping $\mathcal{H} : \mathbb{R}^n \rightrightarrows \mathbb{R}^{n \times n}$ defined by

$$\mathcal{H}(d) := \{ H \in \mathbb{R}^{n \times n} | \ H = I_n - B_\Gamma^T (B_\Gamma B_\Gamma^T)^{-1} B_\Gamma, \ \Gamma \in \mathcal{K}(d) \},$$

(3.9)

where

$$\mathcal{K}(d) := \{ \Gamma \subseteq \{1, \ldots, n\} | \ \text{Supp}(z(d)) \subseteq \Gamma \subseteq \mathcal{I}(\Pi_C(d)) \}.$$

Here,

$$z(d) := (B^T)^{-1}(d - \Pi_C(d))$$

is the optimal solution to the dual of the projection problem; Supp$(z(d))$ denotes the support of $z(d)$, which is the set of indices $i$ such that $z(d)_i \neq 0$, and

$$\mathcal{I}(\Pi_C(d)) := \{ i \in \{1, \ldots, n\} | (BP\Pi_C(d))_i = 0 \}$$

is the set of active indices at $\Pi_C(d)$; $B_\Gamma$ is the submatrix obtained by extracting the rows of $B$ with indices in $\Gamma$ with the convention that

$$B_\Gamma^T (B_\Gamma B_\Gamma^T)^{-1} B_\Gamma = 0 \in \mathbb{R}^{n \times n}$$

if $\Gamma = \emptyset$. For any given nonempty index set $\Gamma \subseteq \{1, \ldots, n\}$, it is not difficult to observe that $B_\Gamma$ is of full row rank and $B_\Gamma B_\Gamma^T$ is invertible. Hence, the set-valued mapping $\mathcal{H}$ is well defined over $\mathbb{R}^n$. Here, we term $\mathcal{H}(d)$ as the HS-Jacobian associated with $\Pi_C$ at the point $d$. Meanwhile, the mapping $\mathcal{H}$ has the following desirable property.

**Lemma 3.5.** For any $d \in \mathbb{R}^n$, there exists a neighborhood $W$ of $d$ such that for all $d' \in W$,

$$\mathcal{K}(d') \subseteq \mathcal{K}(d), \quad \mathcal{H}(d') \subseteq \mathcal{H}(d).$$

(3.10)

When $\mathcal{H}(d') \subseteq \mathcal{H}(d)$, it holds that

$$\Pi_C(d') - \Pi_C(d) - H(d' - d) = 0, \quad \forall \ H \in \mathcal{H}(d').$$

(3.11)

In fact,

$$\mathcal{H}(d) \equiv \partial_B \Pi_C(d),$$

i.e., $\mathcal{H}(d)$ is exactly the $B$-subdifferential of $\Pi_C$ at the point $d$.

**Proof.** The first part of the lemma (see (3.10) and (3.11)) follows directly from [14, Lemma 2.1]. Since the LICQ associated with the problem (3.8) holds at $\Pi_C(d)$, the last statement holds from [40, Section 2] and [26, Theorem 3.2 and Corollary 3.2.2].

\[\square\]
Given \( d \in \mathbb{R}^n \), although the formula of the HS-Jacobian \( H \in \mathcal{H}(d) \) defined in (3.9) seems complicated, we show in the following discussion that \( H \) in fact enjoys a very simple yet elegant representation which is largely inherited from the special structure of \( B \). The discussions below are based on [19, Proposition 6] and [22, Section 3.4]. For any given nonempty index set \( \Gamma \subseteq \{1, \ldots, n\} \), define the diagonal matrix \( \Sigma_{\Gamma} \in \mathbb{R}^{n \times n} \) by
\[
(\Sigma_{\Gamma})_{ii} = \begin{cases} 1, & \text{if } i \in \Gamma, \\ 0, & \text{otherwise}. \end{cases}
\]
We note that \( \Sigma_{\Gamma} \) can be partitioned into \( N \) blocks so that the diagonal elements of these submatrices are all the identity matrices or zero matrices and any two consecutive blocks are different in their types:
\[
\Sigma_{\Gamma} = \text{Diag}(\Lambda_1, \ldots, \Lambda_N),
\]
where \( \Lambda_j \in \{O_{n_i}, I_{n_i}\} \) and \( \sum_{i=1}^{N} n_i = n \). Let
\[
H = I_n - B_\Gamma^T (B_\Gamma B_\Gamma^T)^{-1} B_\Gamma,
\]
i.e., the orthogonal projection onto the null space of \( B_\Gamma \). Then by some direct calculations, we see that \( H \) is a nonnegative block diagonal matrix, i.e.,
\[
H = \text{Diag}(H_1, \ldots, H_N),
\]
where
\[
H_i = \begin{cases} \frac{1}{n_i + 1} e_{n_i + 1} e_{n_i + 1}^T, & \text{if } i \in J \text{ and } i \neq N, \\ O_{n_i}, & \text{if } i \in J \text{ and } i = N, \\ I_{n_i-1}, & \text{if } i \notin J \text{ and } i \neq 1, \\ I_{n_i}, & \text{if } i \notin J \text{ and } i = 1
\end{cases}
\]
with \( J := \{j \in \{1, \ldots, N\} \mid \Lambda_j = I_{n_j}\} \) and the convention \( I_0 = \emptyset \). In fact, we can further decompose \( H \) into the sum of a diagonal matrix and a low rank matrix, i.e.,
\[
H = D + UU^T,
\]
where \( D = \text{Diag}(D_1, \ldots, D_N) \) with
\[
D_i = \begin{cases} O_{n_i+1}, & \text{if } i \in J \text{ and } i \neq N, \\ O_{n_i}, & \text{if } i \in J \text{ and } i = N, \\ I_{n_i-1}, & \text{if } i \notin J \text{ and } i \neq 1, \\ I_{n_i}, & \text{if } i \notin J \text{ and } i = 1
\end{cases}
\]
and \( U \in \mathbb{R}^{n \times N} \) with its \((k, j)\)-th entry given by
\[
U_{k,j} = \begin{cases} \frac{1}{\sqrt{n_j + 1}}, & \text{if } \sum_{t=1}^{j-1} n_t + 1 \leq k \leq \sum_{t=1}^{j} n_t + 1 \text{ and } j \in J \setminus \{N\}, \\ 0, & \text{otherwise}. \end{cases}
\]

**Lemma 3.6.** For any \( d \in \mathbb{R}^n \) satisfying \( \Pi_C(d) \neq 0 \), it holds that
\[
H_{11} > 0, \quad \forall H \in \mathcal{H}(d).
\]

**Proof.** Since \( \Pi_C(d) \neq 0 \), we know that
\[
\mathcal{I}(\Pi_C(d)) \neq \{1, \ldots, n\}.
\]
Hence, it holds that \( \Gamma \neq \{1, \ldots, n\} \) for all \( \Gamma \in \mathcal{K}(d) \). If \( \Gamma = \emptyset \), we have \( H = I \) and \( H_{11} = 1 \). If \( \Gamma \neq \emptyset \), we know that \( N > 1 \) with \( N \) being the number of blocks in (3.12). The desired result then follows directly from (3.13).
3.2.2 A semismooth Newton method for (3.7) and its convergence

From Lemma 3.5, we obtain the B-subdifferential of $\Pi C$. However, it is still a nontrivial task for computing the B-subdifferential or the Clarke generalized Jacobian of $\phi'$ defined in (3.6). Here, we propose to circumvent this difficulty by designing a set-valued mapping which can be regarded as an alternative surrogate of $\partial \phi'$. Given a nonzero vector $\lambda \in C$ and $b \in \mathbb{R}^n$, define the set-valued mapping $\mathcal{M} : \mathbb{R} \rightrightarrows \mathbb{R}$ by

$$
\mathcal{M}(y) := \{ M \in \mathbb{R} \mid M = \lambda^T H \lambda, \ H \in \mathcal{H}(y \lambda + b) \},
$$

where $\mathcal{H}$ is the set-valued mapping defined in (3.9). We show in the next proposition that the Lipschitz continuous piecewise affine function $\phi'$ is $\gamma$-order semismooth on $\mathbb{R}$ with respect to $\mathcal{M}$ for any given $\gamma > 0$ in the sense of Definition 2.3 and $\mathcal{M}$ is a legitimate replacement of the Clarke generalized Jacobian of $\phi'$.

**Proposition 3.7.** Let $b \in \mathbb{R}^n$, $y \in \mathbb{R}$ and a nonzero vector $\lambda \in C$ be any given data. Then $\mathcal{M}$ defined in (3.14) is a nonempty and compact valued and upper-semicontinuous set-valued mapping and $M \geq 0$ for all $M \in \mathcal{M}(y)$. Moreover, there exists a neighborhood $W$ of $y$ such that for all $y' \in W$,

$$
\mathcal{M}(y') \subseteq \mathcal{M}(y) \quad \text{and} \quad \phi'(y') - \phi'(y) - M(y' - y) = 0, \quad \forall M \in \mathcal{M}(y').
$$

(3.15)

Therefore, $\partial_B \phi'(y) \subseteq \mathcal{M}(y)$.

**Proof.** By Lemma 3.5, we know that $\mathcal{M}$ is a nonempty and compact valued and upper-semicontinuous set-valued mapping. From the definitions of $\mathcal{H}$ and $\mathcal{M}$ in (3.9) and (3.14), it is easy to see that $M \geq 0$ for all $M \in \mathcal{M}(y)$. Meanwhile, for any given $y \in \mathbb{R}$, from Lemma 3.5, we know that there exists a neighborhood $W$ of $y$ such that for all $y' \in W$,

$$
\mathcal{H}(y' \lambda + b) \subseteq \mathcal{H}(y \lambda + b)
$$

and

$$
\Pi C(y' \lambda + b) - \Pi C(y \lambda + b) - P \lambda (y' - y) = 0, \quad \forall P \in \mathcal{H}(y \lambda + b).
$$

(3.16)

Hence, we know from (3.14) that $\mathcal{M}(y') \subseteq \mathcal{M}(y)$ for all $y' \in W$. Let $y'$ be a fixed but arbitrary vector in $W$. Then for any $M \in \mathcal{M}(y')$, there exists $P \in \mathcal{H}(y \lambda + b)$ such that $M = \lambda^T P \lambda$ and

$$
\phi'(y') - \phi'(y) = (\Pi C(y' \lambda + b) - \Pi C(y \lambda + b), \lambda) = \lambda^T P \lambda (y' - y) = M(y' - y),
$$

where the second equation follows from (3.16). Since $y'$ is chosen arbitrarily from $W$, we know that (3.15) holds. Then (3.15), together with the definition of the B-subdifferential of $\phi'$ at $y$, further implies that $\partial_B \phi'(y) \subseteq \mathcal{M}(y)$.

Next, given $y \in \mathbb{R}$, we show how to construct an element in $\mathcal{M}(y)$. Let the index set $\Gamma = \mathcal{I}(\Pi C(y \lambda + b))$, i.e., the set of active indices at $\Pi C(y \lambda + b)$. Then

$$
H = I_n - B^T \Gamma (B_{\Gamma} B_{\Gamma}^T)^{-1} B_{\Gamma} \in \mathcal{H}(y \lambda + b)
$$

and it holds from the definition of $\mathcal{M}$ in (3.14) that

$$
M = \lambda^T H \lambda \in \mathcal{M}(y).
$$

(3.17)

In the following proposition, we show that all the elements in $\mathcal{M}(y)$ are positive for all $y$ such that $\Pi C(y \lambda + b) \neq 0$. Proposition 3.8 will be critical for the algorithmic design and for establishing the convergence properties of the algorithm.

**Proposition 3.8.** Given a nonzero vector $\lambda \in C$ and $b \in \mathbb{R}^n$, let $S = \{ y \in \mathbb{R} \mid \Pi C(y \lambda + b) \neq 0 \}$. Then $S$ is nonempty and for any $y \in S$, it holds that

$$
M > 0, \quad \forall M \in \mathcal{M}(y),
$$

where the set-valued mapping $\mathcal{M}$ is defined in (3.14).
Proof. We first show that $S \neq \emptyset$. Since $\lambda \in C$ is nonzero, it holds that $\lambda_1 > 0$. Hence, there exists $\hat{y} > 0$ large enough such that $\hat{y}\lambda_1 + b_1 > 0$. Consider a vector $\alpha \in \mathbb{R}^n$ with $\alpha_1 = \hat{y}\lambda_1 + b_1$ and $\alpha_i = 0$ for all $i = 2, \ldots, n$. Clearly, $\alpha \in C$. Observe that

$$
||\alpha - (\hat{y}\lambda + b)||^2 = \sum_{i=2}^{n}(\hat{y}\lambda_i + b_i)^2 < \sum_{i=1}^{n}(\hat{y}\lambda_i + b_i)^2 = ||\hat{y}\lambda + b||^2.
$$

Hence, $\hat{y} \in S$, i.e., $S$ is nonempty.

For any $y \in S$, since $\Pi_C(y\lambda + b) \neq 0$, we know that $\mathcal{I}(\Pi_C(y\lambda + b)) \neq \{1, \ldots, n\}$. By Lemma 3.6, we have $H_{11} \geq 0$ for all $H \in \mathcal{H}(y\lambda + b)$. Therefore, simple calculations show that

$$
\lambda^T H \lambda \geq H_{11} \lambda_1^2 > 0, \quad \forall H \in \mathcal{H}(y\lambda + b).
$$

The desired conclusion then follows from the definition of $\mathcal{M}$ in (3.14).

**Corollary 3.9.** For any given nonzero vector $\lambda \in C$, $b \in \mathbb{R}^n$ and $\tau > 0$, the problem (3.5) has a unique optimal solution, denoted by $y^*$, i.e., the optimal solution set $\Omega_D$ is a singleton. Moreover, $M > 0$ for all $M \in \mathcal{M}(y^*)$.

**Proof.** We note that $\Omega_D$ is exactly the solution set of the univariate piecewise affine equation (3.7). Since $\tau > 0$, $\Pi_C(y\lambda + b) \neq 0$ for any $y \in \Omega_D$. Meanwhile, from Proposition 3.7, we have

$$
\partial \phi'(y) = \text{conv} \partial B \phi'(y) \subseteq \text{conv}(\mathcal{M}(y)).
$$

By Proposition 3.8, we know that for any $y \in \Omega_D$, all the elements in $\partial \phi'(y)$ are positive. Thus, the Clarke inverse function theorem [13, Proposition 7.1.19] implies that $\Omega_D$ is a singleton.

With all the above preparations, we are ready to present a semismooth Newton method for solving (3.7).

---

**Algorithm SSN.** A semismooth Newton algorithm for solving (3.7)

Set $\mu \in (0, 1/2)$ and $\delta \in (0, 1)$ and choose $y^0 \in \mathbb{R}$. Iterate the following steps for $j = 0, 1, \ldots$

**Step 1.** Compute $\Pi_C(y^j\lambda + b)$ and set

$$
d^j = \begin{cases} -\phi'(y^j), & \text{if } \Pi_C(y^j\lambda + b) = 0, \\ -\phi'(y^j)/M_j, & \text{otherwise,} \end{cases}
$$

where $M_j \in \mathcal{M}(y^j\lambda + b)$ is constructed as in (3.17).

**Step 2** (Line search). Set $\alpha_j = \delta^m$, where $m_j$ is the first nonnegative integer $m$ for which

$$
\phi(y^j + \delta^m d^j) \leq \phi(y^j) + \mu \delta^m \phi'(y^j) d^j.
$$

**Step 3.** Set $y^{j+1} = y^j + \alpha_j d^j$.

---

**Theorem 3.10.** Let $\{y^j\}$ be the infinite sequence generated by Algorithm SSN. Then $\{y^j\}$ converges to the unique optimal solution $y^*$ to the problem (3.5). Moreover, it holds that for all $j$ sufficiently large, $y^{j+1} = y^j + d^j$ with $d^j = -\phi'(y^j)/M_j$ and $|y^{j+1} - y^*| = O(|y^j - y^*|^2)$.

**Proof.** By Proposition 3.8, we know that $d^j$ is always well defined and a descent direction. Thus, Algorithm SSN is well defined. Since $\phi$ is a coercive function, the sequence $\{y^j\}$ is bounded. Then the standard convergence analysis [3, Proposition 1.2.1] implies that any cluster point of $\{y^j\}$ is a stationary point of the problem (3.5), and hence an optimal solution since $\phi$ is convex. Corollary 3.9 further implies that $y^j$ converges to the unique optimal solution $y^*$ to (3.5).

Since every element in $\mathcal{M}(y^*)$ is positive and $\mathcal{M}$ is upper semicontinuous, from [13, Lemma 7.5.2], Proposition 3.8 and Corollary 3.9, we have that for all $j$ sufficiently large, $\{M_j^{-1}\}$ is uniformly bounded and $d^j = -\phi'(y^j)/M_j$. Since $\phi'$ is strongly semismooth with respect to $\mathcal{M}$, similar to the proof of [46, Theorem 3.5], it can be shown that for all $j$ sufficiently large,

$$
|y^j + d^j - y^*| = O(|y^j - y^*|^2), \quad (3.18)
$$
and for some constant \( \delta > 0 \), \(-\phi'(y) d^j \geq \delta |d|^2 \). Based on (3.18), Proposition 2.4 and [13, Proposition 8.3.18], we know that for \( \mu \in (0, 1/2) \), there exists an integer \( j_0 \) such that for all \( j \geq j_0 \),
\[
\phi(y^j + d^j) \leq \phi(y^j) + \mu \phi'(y^j) d^j,
\]
i.e., \( y^{j+1} = y^j + d^j \) for all \( j \geq j_0 \). This together with (3.18) completes the proof.

The above theorem asserts the local fast convergence of Algorithm SSN. A careful analysis further shows that Algorithm SSN proposes a finite termination property, i.e., if an iterate \( y^j \) is sufficiently close to the optimal solution \( y^* \), Algorithm SSN finds \( y^* \) in one step.

**Theorem 3.11.** Let \( \{y^j\} \) be the infinite sequence generated by Algorithm SSN. Then there exists a positive integer \( j_0 \) such that
\[
y^j = y^*, \quad \forall j > j_0.
\]

**Proof.** By Theorem 3.10 and Proposition 3.7, we know that there exists a positive integer \( j_0 \) such that for all \( j \geq j_0 \), all the elements in \( M(y^j) \) are positive, \( d_j = -\phi'(y^j)/M_j \) and
\[
y^{j+1} = y^j + d^j \quad \text{and} \quad \phi'(y^j) - \phi'(y^*) - M_j(y^j - y^*) = 0, \quad \forall M_j \in M(y^j).
\]
Then it holds that for all \( j \geq j_0 \),
\[
y^{j+1} - y^* = y^j + d^j - y^*
\]
\[
= - (\phi'(y^j) - M_j(y^j - y^*)) / M_j
\]
\[
= - (\phi'(y^j) - \phi'(y^*) - M_j(y^j - y^*)) / M_j = 0.
\]
We thus complete the proof.

## 4 Generalized Jacobian of \( \Pi_{\Delta} \)

Besides the fast computation of \( \Pi_{\Delta} \), another important task is to obtain the certain generalized Jacobian associated with this projector. As is demonstrated in [19–22,45], this concept will be critical to the design of efficient nonsmooth second-order algorithms for solving general OWL1 constrained optimization problems. Similar to our analysis of the projector \( \Pi_C \), the first thought would be trying to derive the HS-Jacobian of \( \Pi_{\Delta} \). However, as far as we know, there is no explicit polyhedral representation for the OWL1 norm constraint \( \{ x \in \mathbb{R}^n \mid \kappa_\lambda(x) \leq \tau \} \) in the literature, which we believe is a nontrivial task. In this section, we propose to overcome this difficulty by exploiting the composition structure of \( \Pi_{\Delta} \), in Proposition 3.1 and investigating the HS-Jacobian of the solution mapping \( \Pi_\Gamma \) associated with the reformulated problem (3.2) in which the polyhedral constraint enjoys a simple representation.

We first study the HS-Jacobian of \( \Pi_\Gamma \). Similar to Subsection 3.2.1, the HS-Jacobian of \( \Pi_\Gamma \) defined in (3.2) at any given point \( w \in \mathbb{R}^n \) can be written as follows:

\[
\mathcal{V}(w) := \left\{ V \in \mathbb{R}^{n \times n} \mid V = I_n - [\lambda B_F^T] \begin{bmatrix} \lambda^T & [\lambda^T B_F^T] \end{bmatrix}^{-1} \begin{bmatrix} \lambda^T \\ B_F \end{bmatrix}, \Gamma \in \mathcal{K}(w) \right\},
\]

where
\[
\mathcal{K}(w) := \left\{ \Gamma \subseteq \{1, \ldots, n\} \mid \text{Supp}(y(w)) \subseteq \Gamma \subseteq \mathcal{I}(\Pi_\Gamma(w)) \right\}
\]
and \( y(w) \) is the unique solution to the dual of (3.2). We remark here that the nonsingularity of the matrix
\[
[\lambda^T \\ B_F]
\]
follows from Proposition 3.3 and Remark 3.4: given \( w \in \mathbb{R}^n \), the uniqueness of the dual optimal solution \( y(w) \) stems from Corollary 3.9. Hence, the set-valued mapping \( \mathcal{V} \) is well defined. Similarly,
the set-valued mapping $\mathcal{V}$ enjoys similar properties to $\mathcal{K}$ in Lemma 3.5. In particular, it is not difficult to show that $\mathcal{V}(w) \equiv \partial_B \Pi_{\mathcal{C}_f}(w)$ for all $w \in \mathbb{R}^n$.

Now, we are ready to define the following set-valued mapping associated with $\Pi_{\Delta_\gamma}$:

$$S(b) := \{ S \in \mathbb{R}^{n \times n} \mid S = P^T V P, \ P \in \Pi^*(b), \ V \in \mathcal{V}(Pb)\}. \quad (4.1)$$

We show in the next proposition that $S$ can be regarded as a computation tractable alternative of the Clarke generalized Jacobian of $\Pi_{\Delta_\gamma}$ and $\Pi_{\Delta_\gamma}$ is $\gamma$-order semismooth with respect to $S$ for any given $\gamma > 0$.

**Proposition 4.1.** Let $\lambda \neq 0 \in C$. Then $S$ defined in (4.1) is a nonempty and compact valued and upper-semicontinuous set-valued mapping, and for any given vector $b \in \mathbb{R}^n$, every $S \in S(b)$ is symmetric positive semidefinite. Moreover, there exists a neighborhood $W$ of $b$ such that for all $b' \in W$,

$$S(b') \subseteq S(b) \quad \text{and} \quad \Pi_{\Delta_\gamma}(b') - \Pi_{\Delta_\gamma}(b) - S(b' - b) = 0, \quad \forall S \in \mathcal{M}(b').$$

Therefore,

$$\partial_B \Pi_{\Delta_\gamma}(b) \subseteq S(b).$$

**Proof.** By using the same arguments in [22, Theorem 1] and noting the properties of $\mathcal{V}$, we can easily obtain the desired results. □

Given any $b \in \mathbb{R}^n$, one can observe that to obtain an element in $S(b)$, an important step is to obtain a matrix $V \in \mathcal{V}(Pb)$ with some $P \in \Pi^*(b)$. We show in the following lemma how this matrix can be efficiently constructed.

**Lemma 4.2.** Let $\Gamma \subseteq \{1, \ldots, n\}$ be a given index set and $\alpha \in \mathbb{R}^n$ be a given nonzero vector. Suppose that $\alpha \notin \text{Range}(B)$. It then holds that

$$I_n - [\alpha B^T_\Gamma] \left( \begin{bmatrix} \alpha^T \\ B^T_\Gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} \alpha^T \\ B^T_\Gamma \end{bmatrix} = H - \frac{1}{\alpha_1^T \alpha} \alpha_1 \alpha^T_1, \quad (4.2)$$

where

$$H = I_n - B^T_\Gamma (B^T_\Gamma B^T_\Gamma)^{-1} B^T_\Gamma$$

and $\alpha_1 \in \mathbb{R}^n$ denotes the orthogonal projection of $\alpha$ onto the null space of $B^T_\Gamma$, i.e.,

$$\alpha_1 = H \alpha = \alpha - B^T_\Gamma (B^T_\Gamma B^T_\Gamma)^{-1} B^T_\Gamma \alpha.$$

**Proof.** Since $\alpha \notin \text{Rang}(B)$, $\alpha_1$, the orthogonal projection of $\alpha$ onto $\text{Null}(B^T_\Gamma)$, is not a zero vector. Moreover, it is not difficult to see that

$$B^T_\Gamma \alpha_1 = 0$$

and

$$\text{Null} \left( \begin{bmatrix} \alpha^T \\ B^T_\Gamma \end{bmatrix} \right) = \text{Null} \left( \begin{bmatrix} \alpha_1^T \\ B^T_\Gamma \end{bmatrix} \right).$$

Hence, by [20, Lemma 1], we know that

$$[\alpha B^T_\Gamma] \left( \begin{bmatrix} \alpha^T \\ B^T_\Gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} \alpha^T \\ B^T_\Gamma \end{bmatrix} = [\alpha_1 B^T_\Gamma] \left( \begin{bmatrix} \alpha_1^T \\ B^T_\Gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} \alpha_1^T \\ B^T_\Gamma \end{bmatrix}$$

$$= [\alpha_1 B^T_\Gamma] \left( \begin{bmatrix} \alpha_1^T \alpha_1 \end{bmatrix}^{-1} (B^T_\Gamma B^T_\Gamma)^{-1} \right) \begin{bmatrix} \alpha_1^T \\ B^T_\Gamma \end{bmatrix}$$

$$= \frac{1}{\alpha_1^T \alpha_1} \alpha_1 \alpha^T_1 + B^T_\Gamma (B^T_\Gamma B^T_\Gamma)^{-1} B^T_\Gamma.$$

Then (4.2) follows directly. □
Lemma 4.2 asserts that an element in $\mathcal{V}(Pb)$ can be easily obtained by performing an additional rank one update to the corresponding matrix $H$. We further note from the discussions in Subsection 3.2.1 that the matrix $H$ can be decomposed into the summation of a diagonal matrix and a low rank matrix. Hence, the elements in $\mathcal{V}(Pb)$ can be constructed in low computational costs and so are the elements in $\mathcal{S}(b)$ as the corresponding matrix $P$ is nothing but a signed permutation matrix.

5 Numerical experiments

In this section, we shall evaluate our algorithm for solving the problem (1.1), i.e., the projection onto the OWL1 norm ball. We compare our algorithm with the two best-known methods, i.e., the root-finding scheme in [44] and the Davis’ approach in [10] which consists of repeated partial sorting and averaging steps. While the explanations of Davis’ approach may require some new definitions and technical jargons, the root-finding scheme can be easily explained. One will see later that this approach is in fact closely related to our method.

We briefly review the root-finding scheme in [44] for solving the problem (1.1). Define the optimal solution mapping $\beta$ as

$$
\beta(\mu) := \arg\min_x \left\{ \frac{1}{2} \|x - b\|^2 + \mu \kappa_\lambda(x) \right\}, \quad \forall \mu \geq 0
$$

and the function $\varphi : \mathbb{R}_+ \rightarrow \mathbb{R}$ as $\varphi(\mu) = \kappa_\lambda(\beta(\mu))$. Then the root-finding algorithm aims at solving the following univariate nonlinear equation:

$$
\varphi(\mu) = \tau, \quad \mu \in [0, \kappa_\lambda^\circ(\lambda(b))]
$$

where $\kappa_\lambda^\circ$ is the dual norm, or equivalently the polar function of $\kappa_\lambda$ defined as:

$$
\kappa_\lambda^\circ(y) := \sup_x \{ \langle x, y \rangle \mid \kappa_\lambda(x) \leq 1 \}, \quad \forall y \in \mathbb{R}^n.
$$

From [44, Lemma 5], we know that there exists a unique solution $\mu^*$ to the equation (5.1) on the interval $[0, \kappa_\lambda^\circ(\lambda(b))]$ and

$$
\beta(\mu^*) = \Pi_{\Delta_\lambda}(b),
$$

i.e., $\beta(\mu^*)$ solves the problem (1.1). In [44], the equation (5.1) is solved via the Dekker-Brent method [7,11] and the evaluation of $\beta(\mu)$ for any given $\mu$ relies again on the PAVA. From the discussions in [44] and Proposition 3.1, we know under the certain assumption that the equations (5.1) and (3.7) are in fact optimality conditions associated with equivalent optimization problems. In particular, one may view the equation (5.1) as a reparameterization of the equation (3.7). Hence, the comparisons presented in this section between our method and the root-finding approach can also be regarded as the comparisons between using the semismooth Newton method and the Dekker-Brent method for solving the univariate equation (3.7).

We implemented our Algorithm SSN and the root-finding approach [44] in MATLAB. A highly efficient implementation of the PAVA\(^2\) provided in [6] is used to compute $\Pi_C(d)$ for given $d \in \mathbb{R}^n$ in SSN and to compute $\beta(\mu)$ for given $\mu \geq 0$ in the root-finding approach. As recommended in [44], the highly optimized implementation of the Dekker-Brent method—the MATLAB built-in function `fzero` with default parameter values is used for computing the solution to (5.1). Meanwhile, for the Davis’ approach, we use Davis’ MATLAB mex rapper of his C++ implementation\(^3\). We also note in order to compute $\Pi_{\Delta_\lambda}(b)$ for given $\lambda \in C$ and $b \in \mathbb{R}^n$, all the three tested algorithms employ the preprocessing step on $b$ to obtain the sorted and nonnegative input $|b|^+$ and also the postprocessing step to recover $\Pi_{\Delta_\lambda}(b)$. See Proposition 3.1 in the current paper, [44, Figure 5] and [10, Algorithm 1]. All our computational results are obtained by running MATLAB (Version 9.5) on a desktop (4-core, Intel Core i7-9700K at 3.60 GHz, 16 G RAM).

\(^2\) https://statweb.stanford.edu/~candes/software/SortedL1/software.html

\(^3\) https://people.orie.cornell.edu/dsd95/OWLBall.html
For our Algorithm SSN, we measure the accuracy of an obtained approximate solution \( \tilde{y} \) to the problem (3.5) with the given data \( \lambda, b \in C \) and \( \tau > 0 \) by using the relative residual of the optimality equation (3.6) as follows:

\[
\eta = \frac{|\langle \Pi_C(\tilde{y}\lambda+b),\lambda \rangle - \tau|}{1+\tau}.
\]

For a given tolerance \( \epsilon > 0 \), we will stop Algorithm SSN when \( \eta < \epsilon \). For all the tests in this section, we set \( \epsilon = 10^{-12} \) to demonstrate the finite termination property of Algorithm SSN. In our numerical experiments, it is observed that all the three algorithms output almost the same approximate optimal objective value, i.e., the relative difference of these values between any two algorithms is around \( 10^{-13} \) to \( 10^{-16} \). Hence, we believe that all the algorithms find the “optimal” solution and compare only the wall-clock time of each method.

We test the algorithms with simulated data. In this case, the vector \( b \in \mathbb{R}^n \) is set to be a random vector whose entries are i.i.d. random Gaussian numbers of mean 0 and standard deviation \( \sigma \) with \( \sigma \in \{10^{-3}, 1, 10^{3}\} \).

Meanwhile, the vector \( \lambda \in C \) is obtained by sorting the absolute value of a randomly generated vector in \( \mathbb{R}^n \) using the MATLAB command:

\[
\text{lambda} = \text{sort(abs(randn(n,1))),'descend'}.
\]

To ensure the blank assumption \( \kappa_\lambda(b) > \tau \), we set the parameter

\[
\tau = \beta \kappa_\lambda(b)
\]

with \( \beta \) choosing from \( \{10^{-3}, 10^{-2}, 10^{-1}, 0.5, 0.8\} \).

In our tests, we vary \( n \) from \( 10^6 \) to \( 10^8 \), i.e., we test algorithms with large-scale problems.

In Table 1, we present our numerical results. The reported results are averaged over \( 10^2 \) (for \( n = 10^6 \) and \( n = 10^7 \)) and \( 10 \) (for \( n = 10^8 \)) random realizations. In the table, the first two columns give the parameter \( \beta \) and the problem dimension \( n \). The number of averaged iterations “iter.” and the relative Karush-Kuhn-Tucker (KKT) residual \( \eta \) of Algorithm SSN, and computation times (in seconds) are listed in the last fifteen columns. In all the cases, the best time is in bold. As one can observe, for all the experiments, Algorithm SSN performs best among all the three tested algorithms. We note that for almost all the test instances, Algorithm SSN only needs 3 to 4 iterations to obtain highly accurate solutions with the relative optimality residual \( \eta \) ranging from \( 10^{-14} \) to \( 10^{-16} \). These results clearly demonstrate the nice convergence properties, e.g., the quadratic convergence and the finite termination properties, of Algorithm SSN. Careful comparisons show that our Algorithm SSN can be 2 times faster than the root-finding approach and 70 times faster than Davis’ approach. Moreover, these results also emphasize that the computational time of our algorithm scales linearly with respect to \( n \).

Overall, one can safely conclude that our algorithm SSN is robust and highly efficient for solving projection problems over the OWL1 norm ball. We also note the good performance of the root-finding approach. A possible guess is that the root-finding approach may also enjoy the fast local convergence and the finite termination properties which will be left as a further research topic.

6 Conclusion

In this paper, we introduced an finitely terminated yet efficient semismooth Newton method to compute the projector over the OWL1 norm ball. Numerical experiments demonstrate the efficiency of the algorithm. We also conducted a thorough first-order variational analysis on the projector \( \Pi_{\Delta} \). In particular, the generalized Jacobian associated with \( \Pi_{\Delta} \) is given and its critical sparse plus low rank structure is uncovered. We believe that these results are critical for designing efficient and robust algorithms for solving large-scale general OWL1 norm constraint problems which will be considered in our future work.
Table 1  The performance of the SSN method, root-finding approach and Davis’ approach on the projection problem (1.1). In the table, $n$ is the dimension of $b$; $\sigma$ is the standard deviation of entries of $b$; $\beta$ is the parameters associated with $\tau$; “a” stands for the SSN method; “b” stands for the root-finding approach; “c” stands for Davis’ approach.

| $\beta$ | $n$ | $\eta$ | iter. | Time (s) | $\eta$ | iter. | Time (s) | $\eta$ | iter. | Time (s) |
|---------|-----|-------|------|---------|-------|------|---------|-------|------|---------|
| 1E−03 | 10^6 | 3.3E−14 | 4.3 | 0.1 | 0.2 | 2.0 | 3.2E−14 | 4.3 | 0.1 | 0.2 | 2.0 |
| 1E−02 | 10^6 | 6.2E−14 | 3.7 | 0.1 | 0.2 | 2.0 | 8.7E−14 | 3.7 | 0.1 | 0.2 | 2.0 |
| 1E−01 | 10^6 | 4.8E−16 | 3.0 | 0.1 | 0.2 | 2.0 | 3.8E−16 | 3.0 | 0.1 | 0.2 | 2.1 |
| 5E−01 | 10^6 | 3.3E−16 | 3.0 | 0.1 | 0.2 | 1.8 | 3.2E−16 | 3.0 | 0.1 | 0.2 | 1.8 |
| 8E−01 | 10^6 | 3.5E−16 | 3.0 | 0.1 | 0.1 | 1.3 | 3.4E−16 | 3.0 | 0.1 | 0.1 | 1.3 |
| 1E−03 | 10^6 | 4.8E−14 | 4.0 | 1.2 | 2.5 | 34.1 | 2.7E−14 | 4.0 | 1.2 | 2.5 | 34.1 |
| 1E−02 | 10^7 | 3.3E−14 | 3.0 | 1.0 | 2.0 | 33.8 | 4.7E−14 | 3.0 | 1.0 | 2.1 | 33.8 |
| 1E−01 | 10^7 | 8.0E−16 | 3.0 | 1.1 | 1.8 | 33.8 | 1.0E−15 | 3.0 | 1.1 | 1.9 | 33.8 |
| 5E−01 | 10^7 | 1.1E−15 | 3.0 | 1.2 | 1.7 | 29.0 | 1.0E−15 | 3.0 | 1.2 | 1.8 | 29.0 |
| 8E−01 | 10^7 | 7.3E−16 | 3.0 | 1.1 | 1.5 | 21.4 | 7.8E−16 | 3.0 | 1.1 | 1.5 | 21.4 |
| 1E−03 | 10^6 | 8.5E−14 | 3.9 | 12.2 | 24.2 | 717.9 | 8.4E−14 | 3.9 | 12.2 | 24.0 | 717.9 |
| 1E−02 | 10^8 | 5.4E−15 | 3.0 | 10.5 | 20.2 | 743.8 | 3.4E−15 | 3.0 | 10.5 | 20.4 | 743.7 |
| 1E−01 | 10^8 | 1.8E−15 | 3.0 | 11.3 | 17.3 | 752.8 | 3.1E−15 | 3.0 | 11.2 | 17.8 | 764.7 |
| 5E−01 | 10^8 | 3.2E−15 | 3.0 | 12.9 | 18.0 | 709.0 | 3.9E−15 | 3.0 | 12.8 | 18.7 | 708.0 |
| 8E−01 | 10^8 | 8.8E−14 | 2.9 | 12.1 | 15.9 | 591.0 | 2.9E−15 | 3.0 | 12.9 | 16.2 | 590.1 |
Acknowledgements The second author was supported by National Natural Science Foundation of China (Grant No. 11901107), the Young Elite Scientists Sponsorship Program by CAST (Grant No. 2019QNRC001), the Shanghai Sailing Program (Grant No. 19YF1402600) and the Science and Technology Commission of Shanghai Municipality Project (Grant No. 19511120700).

References
1. Aravkin A Y, Burke J V, Drusvyatskiy D, et al. Level-set methods for convex optimization. Math Program, 2019, 174: 359–390
2. Barlow R E, Bartholomew D J, Bremner J M, et al. Statistical Inference under Order Restrictions: The Theory and Application of Isotonic Regression. New York: Wiley, 1972
3. Bertsekas D P. Nonlinear Programming. Nashua: Athena Scientific, 1999
4. Best M J, Chakravarti N. Active set algorithms for isotonic regression: A unifying framework. Math Program, 1990, 47: 425–439
5. Bhatia R. Matrix Analysis. New York: Springer-Verlag, 1997
6. Bogdan M, van den Berg E, Sabatti C, et al. SLOPE-adaptive variable selection via convex optimization. Ann Appl Stat, 2015, 9: 1103–1140
7. Brent R P. Algorithms for Minimization without Derivatives. Englewood Cliffs: Prentice-Hall, 1973
8. Chan Z X, Sun D F. Constraint nondegeneracy, strong regularity, and nonsingularity in semidefinite programming. SIAM J Optim, 2008, 19: 370–396
9. Clarke F H. Optimization and Nonsmooth Analysis. New York: Wiley & Sons, 1983
10. Davis D. An \(O(n \log(n))\) algorithm for projecting onto the ordered weighted \(\ell_1\) norm ball. arXiv:1505.00870, 2015
11. Dekker T J. Finding a zero by means of successive linear interpolation. In: Constructive Aspects of the Fundamental Theorem of Algebra. New York: Wiley-Interscience, 1969, 37–48
12. Dykstra R L. An algorithm for restricted least squares regression. J Amer Statist Assoc, 1983, 78: 837–842
13. Facchinei F, Pang J-S. Finite-Dimensional Variational Inequalities and Complementarity Problems. New York: Springer, 2003
14. Han J Y, Sun D F. Newton and quasi-Newton methods for normal maps with polyhedral sets. J Optim Theory Appl, 1997, 94: 659–676
15. Haraux A. How to differentiate the projection on a convex set in Hilbert space: Some applications to variational inequalities. J Math Soc Japan, 1977, 29: 615–631
16. Higham N. Computing the nearest symmetric correlation matrix—a problem from finance. IMA J Numer Anal, 2002, 22: 329–343
17. Kruskal J B. Nonmetric multidimensional scaling: A numerical method. Psychometrika, 1964, 29: 115–129
18. Kummer B. Newton's method for non-differentiable functions. Adv Math Optim, 1977, 15: 959–972
19. Lin M X, Liu Y-J, Sun D F, et al. Efficient sparse semismooth Newton methods for the clustered LASSO problem. SIAM J Optim, 2019, 29: 2026–2052
20. Liu X, Sun D F, Toh K-C. On efficiently solving the subproblems of a level-set method for fused LASSO problems. SIAM J Optim, 2018, 28: 1842–1866
21. Luo Z Y, Sun D F, Toh K-C. On the efficient computation of a generalized Jacobian of the projector over the Birkhoff polytope. Math Program, 2020, 179: 419–446
22. Malick J. A dual approach to semidefinite least-squares problems. SIAM J Matrix Anal Appl, 2004, 26: 272–284
23. Mifflin R. Semismooth and semiconvex functions in constrained optimization. SIAM J Control Optim, 1977, 15: 959–972
24. Nesterov Y. A method of solving a convex programming problem with convergence rate \(O(1/k^2)\). Sov Math Dokl, 1983, 27: 372–376
25. Outrata J V, Zowe J. A numerical approach to optimization problems with variational inequality constraints. Math Program, 1995, 68: 105–130
26. Pang J-S. Newton’s method for B-differentiable equations. Math Oper Res, 1990, 15: 311–341
27. Pang J-S, Qi L. A globally convergent Newton method for convex SC1 minimization problems. J Optim Theory Appl, 1995, 85: 631–648
28. Pang J-S, Ralph D. Piecewise smoothness, local invertibility, and parametric analysis of normal maps. Math Oper Res, 1996, 21: 401–426
29. Qi H, Sun D F. A quadratically convergent Newton method for computing the nearest correlation matrix. SIAM J Matrix Anal Appl, 2006, 28: 360–385
30. Qi L, Sun J. A nonsmooth version of Newton’s method. Math Program, 1993, 58: 353–367
Appendix A  Proof of Proposition 3.1

To start the proof of Proposition 3.1, we need the following two reduction lemmas.

Lemma A.1.  For any $b \in \mathbb{R}^n$, it holds that

$$P \Pi_{\Delta_\kappa}(b) = \Pi_{\Delta_\kappa}(Pb) = \arg\min_x \left\{ \frac{1}{2} \|x - Pb\|^2 \left| \kappa_\lambda(x) \leq \tau \right. \right\} \quad (A.1)$$

for all $P \in \Pi^s(b)$.

Proof.  Recalling the definition of $\Pi^s(b)$ in (3.1), we know that $Pb = |b|^1$ for any $P \in \Pi^s(b)$. Hence, $\kappa_\lambda(P \Pi_{\Delta_\kappa}(b)) = \langle |\Pi_{\Delta_\kappa}(b)|^1, \lambda \rangle = \kappa_\lambda(\Pi_{\Delta_\kappa}(b)) \leq \tau$, i.e., $P \Pi_{\Delta_\kappa}(b)$ is a feasible solution to the problem (A.1).

For any $P \in \Pi^s(b)$ and $y \in \mathbb{R}^n$ satisfying $\kappa_\lambda(y) \leq \tau$, we have

$$\frac{1}{2} \|y - Pb\|^2 = \frac{1}{2} \|P(P^Ty - b)\|^2 = \frac{1}{2} \|P^Ty - b\|^2 \geq \frac{1}{2} \|\Pi_{\Delta_\kappa}(b) - b\|^2 = \frac{1}{2} \|P \Pi_{\Delta_\kappa}(b) - Pb\|^2,$$

i.e., $P \Pi_{\Delta_\kappa}(b)$ is an optimal solution to the problem (A.1) for all $P \in \Pi^s(b)$.

Lemma A.2.  Suppose that $\kappa_\lambda(b) > \tau$. Then it holds that $\Pi_{\Delta_\kappa}(b) = \Pi_{C_{\tau}^m}(b)$ for all $b \in C$.

Proof.  Given any $b \in C$, we first show that $(\Pi_{\Delta_\kappa}(b))_i \geq 0$ for all $1 \leq i \leq n$. Suppose otherwise that there exists $i_0 \in \{1, \ldots, n\}$ such that $(\Pi_{\Delta_\kappa}(b))_{i_0} < 0$. Let $u \in \mathbb{R}^n$ be the vector such that $u_{i_0} = -(\Pi_{\Delta_\kappa}(b))_{i_0}$ and $u_i = (\Pi_{\Delta_\kappa}(b))_i$ for all $i \neq i_0$. Then it holds that $u \neq \Pi_{\Delta_\kappa}(b)$ and $\kappa_\lambda(\Pi_{\Delta_\kappa}(b)) = \kappa_\lambda(u)$. Meanwhile, we have

$$\frac{1}{2} \|\Pi_{\Delta_\kappa}(b) - b\|^2 - \frac{1}{2} \|u - b\|^2 = 2u_{i_0}b_{i_0} \geq 0,$$

which contradicts the fact that $\Pi_{\Delta_\kappa}(b)$ is the unique optimizer of the problem (1.1). Thus, $(\Pi_{\Delta_\kappa}(b))_i \geq 0$ for all $1 \leq i \leq n$.

Next, we show that for any $1 \leq i < j \leq n$, it holds that $(\Pi_{\Delta_\kappa}(b))_{i} \geq (\Pi_{\Delta_\kappa}(b))_{j}$. Suppose otherwise that there exists $1 \leq i_0 < j_0 \leq n$ such that $(\Pi_{\Delta_\kappa}(b))_{i_0} < (\Pi_{\Delta_\kappa}(b))_{j_0}$. Then we can construct a new vector $u \in \mathbb{R}^n$ by switching the $i_0$ and $j_0$ entries of $\Pi_{\Delta_\kappa}(b)$, i.e., $u_{i_0} = (\Pi_{\Delta_\kappa}(b))_{j_0}$, $u_{j_0} = (\Pi_{\Delta_\kappa}(b))_{i_0}$ and...
\[ u_i = (\Pi_{\Delta, i}(b))_i \text{ for all } i \in \{1, \ldots, n\}/\{i_0, j_0\}. \]
Then \( u \neq \Pi_{\Delta, i}(b) \) and \( \kappa_{\lambda}(u) = \kappa_{\lambda}(\Pi_{\Delta, i}(b)) \leq \tau. \) Similarly, one can show by some simple calculations that

\[ \frac{1}{2} \| \Pi_{\Delta, i}(b) - b \|^2 - \frac{1}{2} \| u - b \|^2 \geq 0. \]

We again arrive at a contradiction. Hence, we have \( \Pi_{\Delta, i}(b) \in C. \)

Since \( \Pi_{\Delta, i}(b) \in C \) and \( \kappa(b) > \tau, \) we know that \( \langle \lambda, \Pi_{\Delta, i}(b) \rangle = \kappa_{\lambda}(\Pi_{\Delta, i}(b)) = \tau, \) i.e., \( \Pi_{\Delta, i}(b) \) is a feasible solution to the problem (3.2). Hence,

\[ \frac{1}{2} \| \Pi_{\Delta, i}(b) - b \|^2 \leq \frac{1}{2} \| \Pi_{\Delta, i}(b) - b \|^2 \leq \frac{1}{2} \| \Pi_{\Delta, i}(b) - b \|^2, \]

where the last inequality follows from the fact that \( \Pi_{\Delta, i}(b) \) is a feasible solution to the problem (1.1). Hence, the uniqueness of the optimal solution to the problem (3.2) implies that \( \Pi_{\Delta, i}(b) = \Pi_{\Delta, i}(b). \)

**Proof of Proposition 3.1.** For any \( b \in \mathbb{R}^n \) satisfying \( \kappa_{\lambda}(b) > \tau \) and \( P \in \Pi^+(b), \) we know from Lemmas A.1 and A.2 that

\[ P \Pi_{\Delta, i}(b) = \Pi_{\Delta, i}(b) = \Pi_{\Delta, i}(b). \]

Since \( P \) is a signed permutation matrix, we further have \( \Pi_{\Delta, i}(b) = P^T \Pi_{\Delta, i}(Pb). \)