

Non-degenerate Piecewise Linear Systems: A Finite Newton Algorithm and Applications in Machine Learning

Xiao-Tong Yuan, Shuicheng Yan

Department of Electrical and Computer Engineering

National University of Singapore, 117583, Singapore

xtyuan1980@gmail.com, eleyans@nus.edu.sg

Keywords: Piecewise linear systems, non-smooth Newton method, linear complementary problem, elitist Lasso, box constrained least squares, support vector machines

Abstract

We investigate Newton-type optimization methods for solving piecewise linear systems (PLSs) with *non-degenerate* coefficient matrix. Such systems arise, for example, from the numerical solution of linear complementarity problem which is useful to model several learning and optimization problems. In this paper, we propose an effective damped Newton method, namely PLS-DN, to find the exact (up to machine precision) solution of non-degenerate PLSs. PLS-DN exhibits provable semi-iterative property, i.e., the algorithm converges globally to the exact solution in a finite number of iterations. The rate of convergence is shown to be at least linear before termination. We emphasize the applications of our method in modeling, from a novel perspective of PLSs, some statistical learning problems such as box constrained least squares, elitist Lasso (Kowalski & Torreesani, 2008) and support vector machines (Cortes & Vapnik, 1995). Numerical

results on synthetic and benchmark data sets are presented to demonstrate the effectiveness and efficiency of PLS-DN on these problems.

1 Introduction

Recently, Brugnano & Sestini (2009a) introduced and investigated the *piecewise linear systems* which involve non-smooth functions of the solution itself

$$\min\{\mathbf{0}, \mathbf{x}\} + \mathbf{T} \max\{\mathbf{0}, \mathbf{x}\} = \mathbf{b}, \quad (1)$$

where $\mathbf{x} = (x_i) \in \mathbb{R}^d$ is an unknown variable vector, $\mathbf{T} = (t_{ij}) \in \mathbb{R}^{d \times d}$ is a known coefficient matrix, $\mathbf{b} \in \mathbb{R}^d$ is a known vector, and

$$\min\{\mathbf{0}, \mathbf{x}\} := (\min\{0, x_i\}), \quad \max\{\mathbf{0}, \mathbf{x}\} := (\max\{0, x_i\}).$$

The systems (1), abbreviated by PLSs(\mathbf{b}, \mathbf{T}) hereafter, were originally proposed in (Brugnano & Casulli, 2008), and their applications have then been considered in (Brugnano & Sestini, 2009a,b). In practice, the PLSs(\mathbf{b}, \mathbf{T}) arise from the semi-implicit methods for the numerical simulation of free-surface hydrodynamics (Casulli, 1990; Stelling & Duynmeyer, 2003) and the numerical solutions to obstacle problems (Brugnano & Casulli, 2008; Brugnano & Sestini, 2009a,b). For these problems, the coefficient matrix \mathbf{T} in PLSs is typically a symmetric M -matrix (see Assumption (A1) in Section 1.3 for a definition) or inverse-positive matrix, in condition of which several finite Newton methods have been proposed in literatures (Brugnano & Casulli, 2008; Brugnano & Sestini, 2009a; Chen & Agarwal, 2010).

Since $\min\{\mathbf{0}, \mathbf{x}\} = \mathbf{x} - \max\{\mathbf{0}, \mathbf{x}\}$, systems (1) can be equivalently written by:

$$\mathbf{x} + (\mathbf{T} - \mathbf{I}) \max\{\mathbf{x}, \mathbf{0}\} = \mathbf{b},$$

which indeed is a special case of the following systems (Brugnano & Casulli, 2009):

$$\mathbf{x} + (\mathbf{T} - \mathbf{I}) \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}\}\} = \mathbf{b}, \quad (2)$$

where $\mathbf{l} = (l_i)$, $\mathbf{u} = (u_i)$, $\mathbf{b} = (b_i) \in \mathbb{R}^d$ are known vectors and $l_i \leq u_i$. We call the preceding equation systems as PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$). Obviously, when $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} =$

∞ , PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) reduces to the PLSs(\mathbf{b}, \mathbf{T}). When $(\mathbf{T} - \mathbf{I})^{-1}$ is a symmetric M -matrix, Brugnano & Casulli (2009) proposed two finite Newton algorithms to solve PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) along with applications in confined-unconfined flows in porous media.

In this paper, we are particularly concerned with Newton-type methods for solving a wide class of PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) where \mathbf{T} is *non-degenerate*, i.e., every principal minor is non-zero. Such systems arise from several concrete machine learning problems to be addressed in Section 4. The present work generalizes our previous work (Yuan & Yan, 2011) on Newton-type algorithms for PLSs(\mathbf{b}, \mathbf{T}) along with applications to machine learning. Before continuing, we first establish notation formally.

1.1 Notation and Definitions

Matrices are upper case mathematical bold letters, such as $\mathbf{T} \in \mathbb{R}^{n \times n}$, vectors are lower case mathematical bold letters, such as $\mathbf{x} \in \mathbb{R}^d$, and scalars are lower case italics such as $x \in \mathbb{R}$. The i th component of a vector \mathbf{x} is denoted by x_i or $[\mathbf{x}]_i$ interchangeably. By $\|\mathbf{x}\|_p$, we denote the ℓ_p -norm of a vector \mathbf{x} , in particular, $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}'\mathbf{x}}$ denotes the Euclidean norm and $\|\mathbf{x}\|_1 = \sum_{i=1}^d |x_i|$. If nothing else said, $\|\cdot\| = \|\cdot\|_2$. By $\rho(\mathbf{T})$, we denote the spectral norm, i.e., the largest singular value of matrix \mathbf{T} . Throughout this paper, the index set $\{1, \dots, d\}$ is abbreviated by \mathcal{I} . For arbitrary $\mathbf{x} \in \mathbb{R}^d$ and $J \subseteq \mathcal{I}$, the vector \mathbf{x}_J consists of the components $x_i, i \in J$. For a given matrix $\mathbf{T} = (t_{ij}) \in \mathbb{R}^{d \times d}$ and $J, J' \subseteq \mathcal{I}$, $\mathbf{T}_{JJ'}$ denotes the sub-matrix $(t_{ij})_{i \in J, j \in J'}$. In the following discussion, we always assume that $J \neq \emptyset$. We denote $\mathbf{0}$ and \mathbf{O} the size compatible all-zero vector and matrix respectively, and $\mathbf{1}$ a size compatible all-one vector.

As aforementioned, in this study we are interested in the situation where \mathbf{T} is a non-degenerate matrix defined by

Definition 1 (Non-degenerate matrix). *Let $\mathbf{T} \in \mathbb{R}^{d \times d}$. Then \mathbf{T} is said to be a non-degenerate matrix if $\det(\mathbf{T}_{JJ}) \neq 0$ for all $J \subseteq \mathcal{I}$.*

By definition we have that a non-degenerate matrix is non-singular and the following simple result immediately holds:

Lemma 1. *If $\mathbf{T} \in \mathbb{R}^{d \times d}$ is a non-degenerate matrix, then for any $J \subseteq \mathcal{I}$, \mathbf{T}_{JJ} is a non-degenerate matrix and thus is non-singular.*

The P -matrix as defined below is a special class of non-degenerate matrix, which is useful in the discussion on uniqueness of PLSs solution in Section 3.3.

Definition 2 (P -matrix). Let $\mathbf{T} \in \mathbb{R}^{d \times d}$. Then \mathbf{T} is said to be a P -matrix if $\det(\mathbf{T}_{JJ}) > 0$ for all $J \subseteq \mathcal{I}$.

It is well known that \mathbf{T} is a P -matrix if and only if, for all $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{x} \neq \mathbf{0}$, there exists an index $i \in \mathcal{I}$ such that $x_i \neq 0$ and $x_i[\mathbf{T}\mathbf{x}]_i > 0$ (see, e.g., Horn & Johnson, 1991). From this knowledge we may easily verify that a positive-definite matrix \mathbf{T} (i.e., $\mathbf{x}'\mathbf{T}\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{x} \neq \mathbf{0}$) is a P -matrix.

1.2 Motivating Examples

To motivate our study on non-degenerate PLSs, we briefly describe in this subsection two learning problems that can be formulated as such a class of PLSs.

Box Constrained Least Squares: Consider the following box constrained least squares (BCLS) problem

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^n (b_i - \mathbf{w}'\mathbf{a}_i)^2, \quad \text{subject to } \mathbf{l} \leq \mathbf{w} \leq \mathbf{u}.$$

Here we assume that $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$ has full row rank so that the preceding problem is a strictly convex optimization problem and thus there exists a unique solution \mathbf{w}^* . As stated in Proposition 2 in Section 4.1, the optimal solution of BCLS is given by $\mathbf{w}^* = \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}^*\}\}$, where \mathbf{x}^* is the solution of the following PLSs($\mathbf{A}\mathbf{A}'$, \mathbf{b} , \mathbf{l} , \mathbf{u}):

$$\mathbf{x} + (\mathbf{A}\mathbf{A}' - \mathbf{I}) \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}\}\} = \mathbf{A}\mathbf{b}.$$

Since \mathbf{A} has full row rank, the coefficient matrix $\mathbf{A}\mathbf{A}'$ is positive-definite, which implies non-degenerate, but not necessarily an M -matrix or inverse-positive.

Elitist Lasso: Another important motivation, for solving non-degenerate PLSs, stands in the efficient optimization of the *elitist Lasso* (Kowalski & Torreesani, 2008). A full description of elitist Lasso is given in Section 4.2. Let us consider here its *proximity operator* form:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{w} - \mathbf{z}\|^2 + \frac{\lambda}{2} |\mathbf{w}'\mathbf{Q}\mathbf{w}|,$$

where $|\mathbf{w}| := (|w_i|)$ is the element-wise absolute vector of \mathbf{w} , $\mathbf{z} = (z_i)$ is a known vector, and positive-semidefinite matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ is defined by several possibly overlapping groups of features. As stated in Proposition 3 in Section 4.2, the optimal solution \mathbf{w}^* is given by

$$w_i^* = \text{sign}(z_i) \max\{0, x_i^*\}, \forall i = 1, \dots, d,$$

where \mathbf{x}^* is the solution of the following PLSs($|\mathbf{z}|, \lambda\mathbf{Q} + \mathbf{I}$):

$$\min\{\mathbf{0}, \mathbf{x}\} + (\lambda\mathbf{Q} + \mathbf{I}) \max\{\mathbf{0}, \mathbf{x}\} = |\mathbf{z}|.$$

Clearly, for $\lambda > 0$, the matrix $\mathbf{T} = \lambda\mathbf{Q} + \mathbf{I}$ is positive-definite, which implies non-degenerate.

From these two examples we can see that developing efficient algorithms for solving non-degenerate PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) is of particular interests in machine learning.

1.3 Existing Finite Newton Methods for PLSs

We briefly review in this subsection several existing finite Newton-type methods for solving PLSs(\mathbf{b}, \mathbf{T}) and general PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$). For obstacle problems, Brugnano & Sestini (2009a) proposed a finite Newton method to solve PLSs(\mathbf{b}, \mathbf{T}) with \mathbf{T} satisfying either one of the following two assumptions:

- (A1) \mathbf{T} is an M -matrix (i.e., it can be written as $\mathbf{T} = \alpha\mathbf{I} - \mathbf{B}$ with $\mathbf{B} \geq \mathbf{O}$ and $\rho(\mathbf{B}) < \alpha$), or
- (A2) $\text{null}(\mathbf{T}') \equiv \text{span}(\mathbf{v})$, $\text{null}(\mathbf{T}) \equiv \text{span}(\mathbf{w})$, with $\mathbf{v}, \mathbf{w} > \mathbf{0}$, and $\mathbf{T} + \mathbf{D}$ is an M -matrix for all diagonal matrices $\mathbf{D} \succeq \mathbf{O}$ (i.e., $\mathbf{D} \geq \mathbf{O}$ and $\mathbf{D} \neq \mathbf{O}$).

It has been shown (Brugnano & Sestini, 2009a, Corollary 9) that the said method converges monotonically and terminates within d iterations. A variant of this method was originally proposed in a earlier work (Brugnano & Casulli, 2008) under slightly different formulations. More recently, Chen & Agarwal (2010) proposed a similar finite Newton PLSs solver under the weaker assumption

- (A3) \mathbf{T} is an inverse-positive matrix, i.e., $\mathbf{T}^{-1} \geq \mathbf{O}$,

which still guarantees that the method converges to an exact solution in at most $d + 1$ iterations. For confined-unconfined flows problem in porous media, Brugnano & Casulli (2009) further extended the method in (Brugnano & Sestini, 2009a) to two finite Newton algorithms for solving PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) with \mathbf{T} satisfying

(A4) $(\mathbf{T} - \mathbf{I})^{-1}$ is a symmetric M -matrix.

Both algorithms are shown to terminate in at most $d(d + 1)/2$ iterations (Brugnano & Casulli, 2009, Theorem 2).

Despite the remarkable success, it is still unclear about the performance of Newton-type method when applied to solve the non-degenerate PLSs which are obviously beyond those covered by conditions (A1)~(A4).

1.4 Our Contribution

The major contribution of this paper is the PLS-DN algorithm along with its analysis to solve the PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) with non-degenerate matrix \mathbf{T} and arbitrary vector \mathbf{b} . PLS-DN is a semi-smooth damped Newton method with global convergence guaranteed. The rate of convergence is shown to be at least linear for the entire solution sequence. One interesting finding is that, even targeting the wide class of non-degenerate coefficient matrix, PLS-DN method still exhibits provable finite termination behavior. Moreover, the existence and uniqueness of solution are guaranteed under mild conditions.

We then study the applications of PLS-DN in learning problems including box constrained least squares (BCLS), elitist Lasso (eLasso) and support vector machines (SVMs). For BCLS, we reformulate the problem as PLSs with positive-definite coefficient matrix. Numerical results on benchmarks show that PLS-DN outperforms several representative Newton-type BCLS solvers. For the problem of eLasso, we are interested in the general case with group overlaps, which to the best of our knowledge has not yet been explicitly addressed in literature. We propose a proximal optimization method in which the proximity operator is characterized by solving PLSs with positive-definite coefficient matrix. For SVMs, we show that the non-linear SVMs in primal form can be numerically modeled as PLSs with positive-definite coefficient matrix. The PLS-DN solver in this setting is closely related to the Newton-type algorithm stated in (Chap-

pelle, 2007). With the analysis stated in this work, we are able to provide finite termination guarantee for such a kind of primal SVMs solver.

The remainder of the paper is structured as follows: The mathematical background is stated in Section 2. We present the PLS-DN algorithm along with its convergence analysis in Section 3. The applications of PLS-DN to learning problems are investigated in Section 4. We conclude this work in Section 5.

2 Mathematical Background

In this section, we first propose in Section 2.1 a dual problem to $PLSs(\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u})$. In particular, we may establish a primal-dual connection between $PLSs(\mathbf{b}, \mathbf{T})$ and the well known *linear complementary problem* (LCP) (see, e.g., Cottle et al., 1992) for which several off-the-shelf solvers are available in literature. Such a connection also leads to our results on uniqueness of non-degenerate PLSs solution in Section 3.3. We then introduce in Section 2.2 some mathematical preliminaries used in our analysis.

2.1 A Dual Problem

Let us consider the following systems on $\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^d$:

$$\mathbf{l} \leq \mathbf{y} \leq \mathbf{u}, \quad \boldsymbol{\alpha}, \boldsymbol{\beta} \geq \mathbf{0}, \quad \boldsymbol{\alpha}'(\mathbf{y} - \mathbf{l}) = 0, \quad \boldsymbol{\beta}'(\mathbf{y} - \mathbf{u}) = 0, \quad \boldsymbol{\alpha} - \boldsymbol{\beta} = \mathbf{T}\mathbf{y} - \mathbf{b}. \quad (3)$$

where matrix \mathbf{T} and vectors $\mathbf{b}, \mathbf{l}, \mathbf{u}$ are known. The following theorem shows that if we regard $PLSs(\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u})$ as a primal problem, then the preceding systems can be viewed as its dual problem.

Theorem 1. *For any matrix $\mathbf{T} \in \mathbb{R}^{d \times d}$ and vector $\mathbf{b} \in \mathbb{R}^d$,*

- (a) *If $(\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is a solution of systems (3), then $\mathbf{x} = \mathbf{y} - \boldsymbol{\alpha} + \boldsymbol{\beta}$ is a solution of $PLSs(\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u})$.*
- (b) *If \mathbf{x} is a solution of $PLSs(\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u})$, then $\mathbf{y} = \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}\}\}$, $\boldsymbol{\alpha} = \max\{\mathbf{0}, \mathbf{l} - \mathbf{x}\}$ and $\boldsymbol{\beta} = \max\{\mathbf{0}, \mathbf{x} - \mathbf{u}\}$ together give a solution of systems in (3).*

The proof is given in Appendix A.1. In particular, when $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} = \infty$, the dual problem (3) reduces to the well known *linear complementary problem* (LCP) (see, e.g.,

Cottle et al., 1992), which is defined as the following systems on \mathbf{y} :

$$\mathbf{y} \geq \mathbf{0}, \quad \mathbf{T}\mathbf{y} - \mathbf{b} \geq \mathbf{0}, \quad \mathbf{y}'(\mathbf{T}\mathbf{y} - \mathbf{b}) = 0. \quad (4)$$

We refer the above form as $\text{LCP}(\mathbf{b}, \mathbf{T})$. As a direct consequence of Theorem 1, the following corollary indicates the primal-dual connection between $\text{PLSs}(\mathbf{b}, \mathbf{T})$ and $\text{LCP}(\mathbf{b}, \mathbf{T})$.

Corollary 1. *For any matrix $\mathbf{T} \in \mathbb{R}^{d \times d}$ and vector $\mathbf{b} \in \mathbb{R}^d$,*

- (a) *If \mathbf{y} is a solution of $\text{LCP}(\mathbf{b}, \mathbf{T})$ in (4), then $\mathbf{x} = \mathbf{y} - \mathbf{T}\mathbf{y} + \mathbf{b}$ is a solution of $\text{PLSs}(\mathbf{b}, \mathbf{T})$ in (1).*
- (b) *If \mathbf{x} is a solution of $\text{PLSs}(\mathbf{b}, \mathbf{T})$ in (1), then $\mathbf{y} = \max(\mathbf{0}, \mathbf{x})$ is a solution of $\text{LCP}(\mathbf{b}, \mathbf{T})$ in (4).*

Since $\text{PLSs}(\mathbf{b}, \mathbf{T})$ can be cast to an $\text{LCP}(\mathbf{b}, \mathbf{T})$, one may alternatively solve $\text{PLSs}(\mathbf{b}, \mathbf{T})$ by using existing LCP solvers such as pivoting methods (Cottle et al., 1992; Eaves, 1971) and interior-point methods (Potra & Liu, 2006; Wright, 1997). These methods are characterized by having convergence which is only asymptotic, thus the exact solution is obtained only in the limit of an infinite number of iterations. Alternatively, linear as well as non-linear complementarity problems can be solved by means of non-smooth / semi-smooth Newton methods (Pang, 1990; Harker & Pang, 1990; Qi, 1993; Fischer, 1995). Among others, a damped Newton method that applies to large-scale standard LCP has been investigated in (Harker & Pang, 1990). There, the matrix \mathbf{T} was assumed to be a non-degenerate matrix as addressed in this paper. It has been shown in (Fischer & Kanzow, 1996) that Harker and Pang's algorithm terminates in finite iterations under standard assumptions.

Although $\text{PLSs}(\mathbf{b}, \mathbf{T})$ can be solved in dual with some off-the-shelf LCP solvers, directly addressing $\text{PLSs}(\mathbf{b}, \mathbf{T})$ and the general $\text{PLSs}(\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u})$ in primal using finite Newton method is of algorithmic interests and still remains open for non-degenerate coefficient matrix. Moreover, PLS-DN method enriches the bank of LCP solvers.

2.2 Preliminary

Notice the the left-hand side of $\text{PLSs}(\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u})$ systems (2) is not everywhere differentiable but semi-smooth. Therefore we resort to Pang's damped Newton method (Pang,

1990) for solving PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$). Let us define function $F : \mathbb{R}^d \mapsto \mathbb{R}^d$

$$F(\mathbf{x}) := \mathbf{x} + (\mathbf{T} - \mathbf{I}) \max\{\mathbf{l}, \min(\mathbf{u}, \mathbf{x})\} - \mathbf{b}. \quad (5)$$

It is easy to check that F is a locally Lipschitz-continuous operator, i.e., $\|F(\mathbf{x}) - F(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ with $L = 1 + \|\mathbf{T} - \mathbf{I}\|_2$. Hence, we can calculate its B-derivative (see, e.g. Pang, 1990; Harker & Xiao, 1990, for details) at point $\mathbf{x}^{(k)}$ on direction $\Delta\mathbf{x}$ as the following directional derivative:

$$\begin{aligned} BF(\mathbf{x}^{(k)}; \Delta\mathbf{x}) &= \lim_{h \rightarrow 0} \frac{F(\mathbf{x}^{(k)} + h\Delta\mathbf{x}) - F(\mathbf{x}^{(k)})}{h} \\ &= \Delta\mathbf{x} + (\mathbf{T} - \mathbf{I}) \lim_{h \rightarrow 0} \frac{\max\{\mathbf{l}, \min(\mathbf{u}, \mathbf{x}^{(k)} + h\Delta\mathbf{x})\} - \max\{\mathbf{l}, \min(\mathbf{u}, \mathbf{x}^{(k)})\}}{h} \\ &= \Delta\mathbf{x} + (\mathbf{T} - \mathbf{I})\mathbf{s}^{(k)}(\Delta\mathbf{x}), \end{aligned} \quad (6)$$

where vector $\mathbf{s}^{(k)}(\Delta\mathbf{x}) = (s_i^{(k)}(\Delta\mathbf{x}))$ is given by

$$s_i^{(k)}(\Delta\mathbf{x}) = \begin{cases} \Delta x_i & \text{if } i \in \alpha(\mathbf{x}^{(k)}) := \{i \in \mathcal{I} \mid l_i < x_i^{(k)} < u_i\} \\ \max\{\Delta x_i, 0\} & \text{if } i \in \beta(\mathbf{x}^{(k)}) := \{i \in \mathcal{I} \mid x_i^{(k)} = l_i\} \\ \min\{\Delta x_i, 0\} & \text{if } i \in \gamma(\mathbf{x}^{(k)}) := \{i \in \mathcal{I} \mid x_i^{(k)} = u_i\} \\ 0 & \text{if } i \in \eta(\mathbf{x}^{(k)}) := \{i \in \mathcal{I} \mid x_i^{(k)} < l_i\} \cup \{i \in \mathcal{I} \mid x_i^{(k)} > u_i\} \end{cases} \quad (7)$$

Based on these preliminaries, we next describe a damped Newton method to efficiently solve non-degenerate PLSs.

3 PLS-DN: A Damped Newton PLSs Solver

Let $g : \mathbb{R}^d \mapsto \mathbb{R}$ defined by

$$g(\mathbf{x}) = \frac{1}{2} \|F(\mathbf{x})\|^2$$

be the norm function of F . We present in Algorithm 1 a damped Newton method, namely PLS-DN, to minimize $g(\mathbf{x})$. Non-smooth Newton methods of this kind were also considered by (Kummer, 1988; Harker & Pang, 1990; Qi, 1993; Ito & Kunisch, 2009). Suppose that the generalized Newton equation (8) has a solution for all $\mathbf{x}^{(k)}$. Under rather mild conditions, e.g., $\liminf t_k > 0$, classical analysis (Pang, 1990; Qi, 1993) shows that Algorithm 1 converges globally to the accumulation point \mathbf{x}^* with $g(\mathbf{x}^*) = 0$, which implies $F(\mathbf{x}^*) = \mathbf{0}$. The rate of convergence is shown to be superlinear under slightly stronger assumptions (Qi, 1993, Theorem 4.3).

Algorithm 1: The PLS-DN method.

Input : A non-degenerate matrix \mathbf{T} , vectors \mathbf{b} , \mathbf{l} and \mathbf{u} .

Output: Vector $\mathbf{x}^{(k)}$.

1 **Initialization:** Choose $\mathbf{x}^{(0)}$, $\theta, \sigma \in (0, 1)$ and set $k := 0$.

2 **repeat**

3 (S.1) Calculate $\Delta\mathbf{x}^{(k)}$ as a solution of the generalized Newton equation

$$BF(\mathbf{x}^{(k)}; \Delta\mathbf{x}) = -F(\mathbf{x}^{(k)}). \quad (8)$$

4 (S.2) Set $t_k := \theta^{m_k}$ where m_k is the smallest nonnegative integer m satisfying the Armijo-Goldstein condition

$$\|F(\mathbf{x}^{(k)} + \theta^m \Delta\mathbf{x}^{(k)})\|^2 \leq (1 - \theta^m \sigma) \|F(\mathbf{x}^{(k)})\|^2.$$

5 (S.3) Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t_k \Delta\mathbf{x}^{(k)}$, $k := k + 1$.

6 **until** $\|F(\mathbf{x}^{(k)})\| = 0$;

3.1 A Modified Algorithm

One difficulty for directly applying Algorithm 1 is that the subproblem of solving the generalized Newton equation (8) is highly non-trivial due to the nonlinearity of vector \mathbf{s} on sets $\beta(\mathbf{x}^{(k)})$ and $\gamma(\mathbf{x}^{(k)})$ (as defined in (7)). Following the terminology in (Harker & Pang, 1990), we call the union index set $\beta(\mathbf{x}^{(k)}) \cup \gamma(\mathbf{x}^{(k)})$ the *degenerate set* whose elements are called the *degenerate indices*. If $\beta(\mathbf{x}^{(k)})$ and $\gamma(\mathbf{x}^{(k)})$ are both empty, then $\mathbf{x}^{(k)}$ is called a *non-degenerate* vector. It is interesting to note that for non-degenerate $\mathbf{x}^{(k)}$, the vector \mathbf{s} is a linear form respect to $\Delta\mathbf{x}$. To see this, following (Brugnano & Casulli, 2009), let us define the following diagonal matrix:

$$\mathbf{P}(\mathbf{x}) = \begin{bmatrix} p(x_1) & & \\ & \ddots & \\ & & p(x_n) \end{bmatrix}, \quad \mathbf{Q}(\mathbf{x}) = \begin{bmatrix} q(x_1) & & \\ & \ddots & \\ & & q(x_n) \end{bmatrix},$$

where $p(x_i) = 1$ if $x_i \geq l_i$, and 0 otherwise, $q(x_i) = 1$ if $x_i > u_i$, and 0 otherwise. It is easy to check that

$$\max[\mathbf{l}, \min(\mathbf{u}, \mathbf{x})] = \mathbf{P}(\mathbf{x})(\mathbf{x} - \mathbf{l}) - \mathbf{Q}(\mathbf{x})(\mathbf{x} - \mathbf{u}) + \mathbf{l}.$$

Thus $F(\mathbf{x})$ can be written as:

$$F(\mathbf{x}) = \mathbf{x} + (\mathbf{T} - \mathbf{I})(\mathbf{P}(\mathbf{x})(\mathbf{x} - \mathbf{l}) - \mathbf{Q}(\mathbf{x})(\mathbf{x} - \mathbf{u}) + \mathbf{l}) - \mathbf{b}. \quad (9)$$

Let $\mathbf{P}^{(k)} := \mathbf{P}(\mathbf{x}^{(k)})$ and $\mathbf{Q}^{(k)} := \mathbf{Q}(\mathbf{x}^{(k)})$. The following result holds immediately.

Lemma 2. *If $\mathbf{x}^{(k)}$ is non-degenerate, then $\mathbf{s}^{(k)}(\Delta\mathbf{x})$ in (6) can be expressed as the following linear form*

$$\mathbf{s}^{(k)}(\Delta\mathbf{x}) = (\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})\Delta\mathbf{x}. \quad (10)$$

Given that $\mathbf{x}^{(k)}$ is non-degenerate, the following proposition shows that the generalized Newton equation in (S.1) of Algorithm 1 can be solved analytically.

Proposition 1. *If $\mathbf{x}^{(k)}$ is non-degenerate, and $\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})$ is non-singular, then the solution of generalized Newton equation (8) is given by*

$$\Delta\mathbf{x} = -\mathbf{x}^{(k)} + (\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)}))^{-1} \mathbf{c}^{(k)}, \quad (11)$$

where $\mathbf{c}^{(k)} := \mathbf{b} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)}\mathbf{l} - \mathbf{Q}^{(k)}\mathbf{u} - \mathbf{l})$.

The proof is given in A.2. Proposition 1 motivates us to modify Algorithm 1 so that the generated sequence $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ remains non-degenerate, and thus the generalized Newton equation (8) always has analytical solution of the form (11). The modified damped Newton method is formally given in Algorithm 2. The major difference between the two algorithms is: in step (S.3), Algorithm 2 adds a sufficiently small positive perturbation to the degenerate indices (if any) of current solution to guarantee the non-degeneracy, which significantly simplifies the calculation in step (S.1). As a result, we have the following theorem on global convergence of Algorithm 2.

Theorem 2 (Global Convergence). *Let $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ be any sequence generated by Algorithm 2. Assume that $F(\mathbf{x}^{(k)}) \neq \mathbf{0}$ for all k . Then*

$$(a) \quad \|F(\mathbf{x}^{(k+1)})\| < \|F(\mathbf{x}^{(k)})\|,$$

(b) *If $\liminf t_k > 0$, then any accumulation point \mathbf{x}^* of sequence $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ is a zero of F , i.e., the solution of PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$).*

The proof is given in Appendix A.3. On convergence rate, we establish in the following result a local linear rate of convergence for the sequence $\{\mathbf{x}^{(k)}\}_{k \geq 0}$. The proof is decayed to Appendix A.4.

Algorithm 2: The modified PLS-DN method.

Input : A non-degenerate matrix \mathbf{T} , vectors \mathbf{b} , \mathbf{l} and \mathbf{u} .

Output: Vector $\mathbf{x}^{(k)}$.

1 **Initialization:** Choose a non-degenerate $\mathbf{x}^{(0)}$, $\theta, \sigma \in (0, 1)$, and set $k := 0$.

2 **repeat**

3 (S.1) Calculate $\Delta \mathbf{x}^{(k)}$ as follows

$$\Delta \mathbf{x}^{(k)} := -\mathbf{x}^{(k)} + (\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)}))^{-1} \mathbf{c}^{(k)}, \quad (12)$$

where $\mathbf{c}^{(k)} := \mathbf{b} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)}\mathbf{l} - \mathbf{Q}^{(k)}\mathbf{u} - \mathbf{l})$.

4 (S.2) Set $t_k := \theta^{m_k}$ where m_k is the smallest nonnegative integer m satisfying the Armijo-Goldstein condition

$$\|F(\mathbf{x}^{(k)} + \theta^m \Delta \mathbf{x}^{(k)})\|^2 \leq (1 - \theta^m \sigma) \|F(\mathbf{x}^{(k)})\|^2. \quad (13)$$

5 (S.3) Set $\tilde{\mathbf{x}}^{(k+1)} := \mathbf{x}^{(k)} + t_k \Delta \mathbf{x}^{(k)}$, $\mathbf{x}^{(k+1)} := \tilde{\mathbf{x}}^{(k+1)}$.

6 **if** $\|F(\tilde{\mathbf{x}}^{(k+1)})\| \neq 0$ **then**

7 **Set** $\mathbf{x}_i^{(k+1)} := \tilde{\mathbf{x}}_i^{(k+1)} + \delta^{(k+1)}$, $\forall i \in \beta(\tilde{\mathbf{x}}^{(k+1)}) \cup \gamma(\tilde{\mathbf{x}}^{(k+1)})$, where
 $0 < \delta^{(k+1)} \leq \frac{(1 - \sqrt{1 - t_k \sigma}) \|F(\mathbf{x}^{(k)})\|}{2L\sqrt{d}}$.

8 **end**

9 $k := k + 1$

10 **until** $\|F(\mathbf{x}^{(k)})\| = 0$;

Theorem 3 (Local Linear Rate of Convergence). *Let $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ be any sequence generated by Algorithm 2. Assume that $F(\mathbf{x}^{(k)}) \neq \mathbf{0}$ for all k . Suppose that \mathbf{x}^* is an accumulation point of $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ and \mathbf{x}^* is a zero of F . If matrix \mathbf{T} is non-degenerate, then the entire sequence $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ converges to \mathbf{x}^* at least linearly.*

Remark 1. *As shown in (Qi, 1993, Theorem 3.4), the standard semi-smooth Newton method like Algorithm 1 enjoys superlinear rate in the final stage of convergence. Due to the perturbation in (S.3) to avoid degeneracy of $\mathbf{x}^{(k)}$, we currently can only prove the local linear rate of convergence for Algorithm 2. In practice, however, we observe that the perturbation seldom occurs in Algorithm 2 since the vectors $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ always automatically remains non-degenerate. Therefore, we may reasonably believe that in practice Algorithm 2 can achieve the same superlinear rate of convergence as Algorithm 1. In our implementation, we simply set $\delta^{(k+1)} = \frac{(1-\sqrt{1-t_k\sigma})\|F(\mathbf{x}^{(k)})\|}{2L\sqrt{d}}$ in (S.3) of Algorithm 2. Under a wide rang of trials on parameter $\theta \in (0, 1)$, we observe that $t_k > 0$ always holds in our numerical experiments. Therefore the condition of $\liminf t_k > 0$ as required in Theorem 2 is not uncommon in practice.*

3.2 Finite Termination

We now claim that Algorithm 2 terminates in one step provided that the current iterate $\mathbf{x}^{(k)}$ is in a sufficient small neighborhood of the accumulation point \mathbf{x}^* . In the following descriptions, we denote $B_\epsilon(\mathbf{y}) := \{\mathbf{z} \in \mathbb{R}^d \mid \|\mathbf{z} - \mathbf{y}\| \leq \epsilon\}$ an Euclidean ball.

Lemma 3. *Let \mathbf{x}^* denote a solution of the PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$). Then there exists a positive number $\epsilon(\mathbf{x}^*)$ such that*

$$(\mathbf{P}(\mathbf{x}) - \mathbf{P}(\mathbf{x}^*))(\mathbf{x}^* - \mathbf{l}) = \mathbf{0}, \quad (\mathbf{Q}(\mathbf{x}) - \mathbf{Q}(\mathbf{x}^*))(\mathbf{x}^* - \mathbf{u}) = \mathbf{0} \quad (14)$$

for all $\mathbf{x} \in B_{\epsilon(\mathbf{x}^*)}(\mathbf{x}^*)$.

The proof is given in Appendix A.5. The following theorem indicates the finite termination property of PLS-DN.

Theorem 4. *Let $\mathbf{x}^* \in \mathbb{R}^d$ denote a solution of the PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$). If $\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})$ is non-singular, and $\mathbf{x}^{(k)} \in B_\epsilon(\mathbf{x}^*)$ for some sufficiently small $\epsilon > 0$, then $\mathbf{x}^{(k+1)}$ generated by Algorithm 2 solves the PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$).*

Proof. Let $\epsilon := \epsilon(\mathbf{x}^*)$ be defined as in Lemma 3. Let $\mathbf{P}^* := \mathbf{P}(\mathbf{x}^*)$. From the fact $F(\mathbf{x}^*) = 0$ we have that

$$\mathbf{x}^* + (\mathbf{T} - \mathbf{I})(\mathbf{P}^*(\mathbf{x}^* - \mathbf{l}) - \mathbf{Q}^*(\mathbf{x}^* - \mathbf{u}) + \mathbf{l}) = \mathbf{b}.$$

By coupling (14) (with $\mathbf{x} = \mathbf{x}^{(k)}$) into the preceding equality, we get

$$\mathbf{x}^* + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)}(\mathbf{x}^* - \mathbf{l}) - \mathbf{Q}^{(k)}(\mathbf{x}^* - \mathbf{u}) + \mathbf{l}) = \mathbf{b},$$

or equivalently

$$(\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})) \mathbf{x}^* = \mathbf{b} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)}\mathbf{l} - \mathbf{Q}^{(k)}\mathbf{u} - \mathbf{l}) = \mathbf{c}^{(k)}. \quad (15)$$

In (S.3) of Algorithm 2, consider $\tilde{\mathbf{x}}^{(k+1)} := \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}$. Since $\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})$ is non-singular, by (12) and (15), we get

$$\tilde{\mathbf{x}}^{(k+1)} = (\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)}))^{-1} \mathbf{c}^{(k)} = \mathbf{x}^*.$$

Therefore we have

$$\|F(\tilde{\mathbf{x}}^{(k+1)})\|^2 = \|F(\mathbf{x}^*)\|^2 = 0 \leq (1 - \sigma) \|F(\mathbf{x}^{(k)})\|^2,$$

i.e., step (S.2) in Algorithm 2 computes $t_k = 1$ and step (S.3) provides $\mathbf{x}^{(k+1)} = \tilde{\mathbf{x}}^{(k+1)} = \mathbf{x}^*$ which terminates the iteration. \square

By Theorem 2 and Theorem 3 we have that the entire sequence $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ converges at least linearly to the accumulation point \mathbf{x}^* . Therefore there exists a $k(\epsilon(\mathbf{x}^*))$ such that for all $k \geq k(\epsilon(\mathbf{x}^*))$, $\mathbf{x}^{(k)} \in B_{\epsilon(\mathbf{x}^*)}(\mathbf{x}^*)$. Theorem 4 guarantees that once $\mathbf{x}^{(k)}$ enters the ball $B_{\epsilon(\mathbf{x}^*)}(\mathbf{x}^*)$, the Algorithm 2 is deemed to terminate after one more step of iteration. The following corollary (of Theorem 2, Theorem 3 and Theorem 4) summarizes the finite termination property of non-degenerate PLSs.

Corollary 2. *If $\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})$ is non-singular at any time instance k , then Algorithm 2 terminates within finite iterates with output $\mathbf{x}^{(k)}$ satisfying $\|F(\mathbf{x}^{(k)})\| = 0$.*

On such a finite termination behavior of Algorithm 2, the following three questions naturally arise:

(Q1): How to numerically verify the stopping criteria $\|F(\mathbf{x}^{(k)})\| = 0$ in Algorithm 2?

(Q2): Under what conditions can we guarantee that $\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})$ is non-singular as required in Theorem 4 and Corollary 2?

(Q3): How about the computational complexity at each iterate?

The following Theorem 5, Theorem 6 and the consequent discussions respectively give answers to these questions.

Theorem 5 (Termination Criteria). *Let $\hat{\mathbf{x}}^{(k+1)} := \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}$. If, for some $k \geq 0$, one gets*

$$\mathbf{P}(\hat{\mathbf{x}}^{(k+1)}) = \mathbf{P}^{(k)}, \quad \mathbf{Q}(\hat{\mathbf{x}}^{(k+1)}) = \mathbf{Q}^{(k)}, \quad (16)$$

then $\mathbf{x}^ := \hat{\mathbf{x}}^{(k+1)}$ is an exact solution of PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$).*

Proof. If (16) holds, then

$$\begin{aligned} & (\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}(\hat{\mathbf{x}}^{(k+1)}) - \mathbf{Q}(\hat{\mathbf{x}}^{(k+1)}))) \hat{\mathbf{x}}^{(k+1)} \\ &= (\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})) \hat{\mathbf{x}}^{(k+1)} \\ &= \mathbf{b} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)}\mathbf{l} - \mathbf{Q}^{(k)}\mathbf{u} - \mathbf{l}) \\ &= \mathbf{b} + (\mathbf{T} - \mathbf{I})(\mathbf{P}(\hat{\mathbf{x}}^{(k+1)})\mathbf{l} - \mathbf{Q}(\hat{\mathbf{x}}^{(k+1)})\mathbf{u} - \mathbf{l}). \end{aligned}$$

where the second equality follows the fact that $\hat{\mathbf{x}}^{(k+1)} := \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}$ and (12). By the preceding equality and (9) we get $F(\hat{\mathbf{x}}^{(k+1)}) = \mathbf{0}$, and thus Algorithm 2 terminates with output $\hat{\mathbf{x}}^{(k+1)}$ that exactly solves (2). \square

Theorem 6 (Non-singularity). *If matrix $\mathbf{T} \in \mathbb{R}^{d \times d}$ is non-degenerate, then $\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})$ is non-singular.*

Proof. Recall that $\mathbf{P}^{(k)}$ is a diagonal matrix with diagonal entries $p(x_i^{(k)}) = 1$ if $x_i^{(k)} \geq l_i$ and 0 otherwise, and $\mathbf{Q}^{(k)}$ is a diagonal matrix with diagonal entries $q(x_i^{(k)}) = 1$ if $x_i^{(k)} > u_i$ and 0 otherwise. Since $\mathbf{l} \leq \mathbf{u}$ we always have $\mathbf{O} \leq \mathbf{Q}^{(k)} \leq \mathbf{P}^{(k)} \leq \mathbf{I}$.

The result obviously holds for $\mathbf{P}^{(k)} = \mathbf{Q}^{(k)} = \mathbf{O}$ and $\mathbf{P}^{(k)} = \mathbf{Q}^{(k)} = \mathbf{I}$. In the following derivation, we assume that $\mathbf{O} \not\leq \mathbf{P}^{(k)} \not\leq \mathbf{I}$. Let us consider the index sets

$$J := \{i \in \mathcal{I} : l_i \leq x_i^{(k)} \leq u_i\} \text{ and } \bar{J} := \mathcal{I} \setminus J. \quad (17)$$

Obviously we have $J \neq \emptyset$. Let $\mathbf{z} \in \mathbb{R}^d$ such that $(\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})) \mathbf{z} = \mathbf{0}$. The definitions of $\mathbf{P}^{(k)}$, $\mathbf{Q}^{(k)}$, J and \bar{J} yield

$$\mathbf{T}_{JJ} \mathbf{z}_J = \mathbf{0}, \quad (18)$$

$$\mathbf{z}_{\bar{J}} + \mathbf{T}_{\bar{J}J} \mathbf{z}_J = \mathbf{0}. \quad (19)$$

By Lemma 1 we have that \mathbf{T}_{JJ} is non-singular, and thus (18) implies $\mathbf{z}_J = \mathbf{0}$. Combining this with (19) yields $\mathbf{z}_{\bar{J}} = \mathbf{0}$. Consequently, we get that $(\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)}))$ is non-singular. \square

As a by-product, Theorem 6 along with its proof motivates us an efficient implementation of step (S.1) in Algorithm 2, which requires solving the following linear systems

$$(\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})) \mathbf{z} = \mathbf{c}^{(k)}, \quad (20)$$

for which a direct solution leads to $O(d^3)$ complexity¹. However, by similar arguments in the proof of Theorem 6, systems (20) can be decomposed as

$$\mathbf{T}_{JJ} \mathbf{z}_J = \mathbf{c}_J^{(k)}, \quad (21)$$

$$\mathbf{z}_{\bar{J}} + \mathbf{T}_{\bar{J}J} \mathbf{z}_J = \mathbf{c}_{\bar{J}}^{(k)}, \quad (22)$$

where J and \bar{J} are given by (17). With such a decomposition, to obtain the solution $\mathbf{z} = \mathbf{z}_{J \cup \bar{J}}$, we only need to solve the smaller linear systems (21) with complexity $O(|J|^3)$ to obtain \mathbf{z}_J , and to solve the equation (22) with complexity $O(|J||\bar{J}|)$ to obtain $\mathbf{z}_{\bar{J}}$. In worst case, i.e., $|J| = d$, the complexity is still traditionally $O(d^3)$. However, when the positive components in the final solution is extremely sparse, $|J| \ll d$ holds – hopefully – during the iteration and the computational cost can be much cheaper than directly solving the linear systems (20). This also answers the question **(Q3)**.

3.3 Existence and Uniqueness of the Solution

We study in this section the existence and uniqueness of the solution of non-degenerate PLSs. Concerning the existence of a solution, the thesis follows directly from Algorithm 2 and Theorem 2. Concerning the uniqueness, one natural question is whether

¹We consider here that solving linear systems takes cubic time. This time complexity can however be improved.

the solution is unique under non-degenerate \mathbf{T} ? The answer is *negative*. To see this, we construct a simple counter example as follows:

A Counter Example: Let $\mathbf{T} = \text{diag}(-1, 1, \dots, 1)$, $\mathbf{b} = (-1, 1, \dots, 1)'$, $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} = \infty$. It is straightforward to check that \mathbf{T} is non-degenerate and both $\mathbf{x}_1^* = (1, 1, \dots, 1)'$ and $\mathbf{x}_2^* = (-1, 1, \dots, 1)'$ are the solutions of PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$).

To further derive the conditions for uniqueness, let us consider the P -matrices which are a subset of non-degenerate matrices.

Lemma 4. *If \mathbf{T} is a P -matrix and $\mathbf{I} \geq \mathbf{D} \geq \mathbf{O}$ is a diagonal matrix, then $\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{D}$ is a non-singular.*

The proof is provided in Appendix A.6. We now present the following main result on the uniqueness of non-degenerate PLSs.

Theorem 7 (Uniqueness of Solution). *If \mathbf{T} is a P -matrix, then the solution of PLSs($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) has a unique solution for all vectors $\mathbf{b} \in \mathbb{R}^d$.*

Proof. The proof follows the arguments in the proof of (Brugnano & Casulli, 2009, Theorem 3) with proper modifications. Let \mathbf{y} be another solution of the same systems (2). Consequently,

$$\mathbf{x} + (\mathbf{T} - \mathbf{I}) \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}\}\} = \mathbf{y} + (\mathbf{T} - \mathbf{I}) \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{y}\}\} \quad (23)$$

Moreover, one has

$$\max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{y}\}\} - \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}\}\} = (\bar{\mathbf{P}} - \bar{\mathbf{Q}})(\mathbf{y} - \mathbf{x}). \quad (24)$$

where $\bar{\mathbf{P}}$ and $\bar{\mathbf{Q}}$ are diagonal matrices, whose diagonal entries, $\{\bar{p}_i\}$ and $\{\bar{q}_i\}$, are respectively given by

$$\bar{p}_i = \begin{cases} 0 & \text{if } x_i, y_i < l_i \\ 1 & \text{if } x_i, y_i \geq l_i \\ \frac{x_i - l_i}{x_i - y_i} & \text{if } x_i \geq l_i > y_i \\ \frac{y_i - l_i}{y_i - x_i} & \text{if } y_i \geq l_i > x_i \end{cases} \quad \bar{q}_i = \begin{cases} 0 & \text{if } x_i, y_i < u_i \\ 1 & \text{if } x_i, y_i \geq u_i \\ \frac{x_i - u_i}{x_i - y_i} & \text{if } x_i \geq u_i > y_i \\ \frac{y_i - u_i}{y_i - x_i} & \text{if } y_i \geq u_i > x_i \end{cases} \quad (25)$$

so that

$$\mathbf{I} \geq \bar{\mathbf{P}} \geq \bar{\mathbf{Q}} \geq \mathbf{O}. \quad (26)$$

From (23) and (24), it holds that

$$(\mathbf{I} + (\mathbf{T} - \mathbf{I})(\bar{\mathbf{P}} - \bar{\mathbf{Q}}))(\mathbf{y} - \mathbf{x}) = \mathbf{0}.$$

Since \mathbf{T} is a P -matrix, from (26) and Lemma 4 we get that $\mathbf{I} + (\mathbf{T} - \mathbf{I})(\bar{\mathbf{P}} - \bar{\mathbf{Q}})$ is non-singular, and uniqueness ($\mathbf{y} = \mathbf{x}$) follows. \square

For the applications in learning problems to be studied in Section 4, the matrices \mathbf{T} are all positive-definite matrices, and thus P -matrices. Therefore, the output solution of our PLS-DN algorithm is unique from any initial point \mathbf{x}^0 .

In particular, for the special case PLSs(\mathbf{b}, \mathbf{T}), we are able to further show that the requirement of \mathbf{T} to be a P -matrix is also necessary for uniqueness. To see this, we need to make use of the primal-dual connection between PLSs(\mathbf{b}, \mathbf{T}) and LCP(\mathbf{b}, \mathbf{T}), as stated in Section 2.1.

Lemma 5. *For any matrix $\mathbf{T} \in \mathbb{R}^{d \times d}$ and vector $\mathbf{b} \in \mathbb{R}^d$, PLSs(\mathbf{b}, \mathbf{T}) has a unique solution if and only if LCP(\mathbf{b}, \mathbf{T}) has a unique solution.*

The proof is given in Appendix A.7. The preceding lemma motivates us to discuss the uniqueness of PLSs solution from the viewpoint of its dual problem, LCP. The following standard result gives a sufficient and necessary condition to guarantee a unique solution of LCP(\mathbf{b}, \mathbf{T}):

Lemma 6 (Theorem 3.3.7 in (Cottle et al., 1992)). *A matrix $\mathbf{T} \in \mathbb{R}^{d \times d}$ is a P -matrix if and only if LCP(\mathbf{b}, \mathbf{T}) has a unique solution for all vectors $\mathbf{b} \in \mathbb{R}^d$.*

In light of Lemma 5 & 6, we are in the position to present the following result on the uniqueness of PLSs(\mathbf{b}, \mathbf{T}) solution.

Theorem 8 (Uniqueness of Solution for PLSs(\mathbf{b}, \mathbf{T})). *PLSs(\mathbf{b}, \mathbf{T}) has a unique solution for all vectors $\mathbf{b} \in \mathbb{R}^d$ if and only if matrix \mathbf{T} is a P -matrix.*

4 Applications to Learning Problems

In this section, we show several applications of PLSs in learning problems. We numerically model the following problems as PLSs and apply the PLS-DN method for optimization: box constrained least squares (Section 4.1), elitist Lasso (Section 4.2), and

primal kernel SVMs (Section 4.3). In the following description, $\mathcal{D} = \{(\mathbf{a}_i, b_i)\}_{1 \leq i \leq n}$ is a set of observed data, $\mathbf{a}_i \in \mathbb{R}^d$ is the feature vector, and b_i is the response being continuous for regression and discrete for classification. Throughout the numerical evaluation in this work, our algorithm was implemented in Matlab 7.7 (R2008b), and the numerical experiments were run on a hardware environment with Intel Core2 CPU 2.83GHz and 8G RAM. The constant parameters in Algorithm 2 are set as $\theta = 0.8$ and $\sigma = 0.01$ throughout the experiments.

4.1 App-I: Box Constrained Least Squares

Many applications, e.g. non-negative image restoration, contact problems for mechanical systems, control problems, involve the numerical solutions of box constrained least squares (BCLS) problems given by

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^n (b_i - \mathbf{w}' \mathbf{a}_i)^2, \quad \text{subject to } \mathbf{l} \leq \mathbf{w} \leq \mathbf{u}. \quad (27)$$

We assume that $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$ has full row rank so that the BCLS problem (27) is a strictly convex optimization problem and there exists a unique solution \mathbf{w}^* .

4.1.1 Solving BCLS with PLSs

The following result shows that BCLS can be reformulated as non-degenerate PLSs.

Proposition 2. *Given that \mathbf{A} has full row rank in BCLS problem (27), then the optimal solution \mathbf{w}^* is given by*

$$\mathbf{w}^* = \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}^*\}\},$$

where \mathbf{x}^* is the unique solution to the following PLSs problem

$$\mathbf{x} + (\mathbf{A}\mathbf{A}' - \mathbf{I}) \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}\}\} = \mathbf{A}\mathbf{b}.$$

Proof. Let $\alpha_i \geq 0$ and $\beta_i \geq 0$ denote the Lagrange multipliers used to enforce the lower and upper bound constraint respectively on w_i . The set of KKT conditions are given by

$$\boldsymbol{\alpha}, \boldsymbol{\beta} \geq \mathbf{0}, \quad \boldsymbol{\alpha}'(\mathbf{w} - \mathbf{l}) = 0, \quad \boldsymbol{\beta}'(\mathbf{w} - \mathbf{u}) = 0, \quad \boldsymbol{\alpha} - \boldsymbol{\beta} = \mathbf{A}\mathbf{A}'\mathbf{w} - \mathbf{A}\mathbf{b}. \quad (28)$$

From Theorem 1 the preceding set of conditions is equivalent to the following PLSs:

$$\mathbf{x} + (\mathbf{A}\mathbf{A}' - \mathbf{I}) \max\{\mathbf{l}, \min\{\mathbf{u}, \mathbf{x}\}\} = \mathbf{A}\mathbf{b}.$$

Since \mathbf{A} has full row rank, the coefficient matrix $\mathbf{A}\mathbf{A}'$ is positive-definite, and thus by Theorem 7 the preceding PLSs has a uniqueness solution \mathbf{x}^* . From Theorem 1 we have that the optimal solution of BCLS is given by $\mathbf{w}^* = \max\{\mathbf{1}, \min\{\mathbf{u}, \mathbf{x}^*\}\}$. \square

So far, we have assumed that the design matrix \mathbf{A} is of full row rank, i.e., the BCLS problem is over-determined. On the other side, for the under-determined case, we may add a proper ridge regularization term $\lambda\|\mathbf{w}\|^2$ to the objective in (27) so that $\mathbf{A}\mathbf{A}' + \lambda\mathbf{I}$ is positive-definite. In this manner, we can still model BCLS as PLSs using similar argument presented in the above analysis.

4.1.2 Simulation

The numerical evaluations of PLS-DN for BCLS problem are carried out on the following three sparse design matrices from the Harwell Boeing collection (Duff et al., 1989): *add20* (2395×2395), *illc1850* (1850×712) and *well1850* (1850×712)². The non-degenerate design matrices \mathbf{A} in these problems are well-conditioned or moderately ill-conditioned. In this test, we uniformly set each element of ground truth \mathbf{w} in interval $[0, 1]$. The i.i.d. noise in linear model is Gaussian with mean 0 and variance 10^{-4} . The initial point is all-one vector. We compare our method with the following Newton-type methods that are capable of solving BCLS:

- The Matlab routine `lsqlin` which is based on reflective Newton method (Coleman & Li, 1996).
- The projected Quasi-Newton (PQN) solver (Schmidt et al., 2009)³ which is based on LBFGS method.
- The TRESNEI solver (Morini & Porcelli, 2010)⁴ which is based on trust-region Gaussian-Newton method.

We set the bound parameters as $l_i \equiv 0$ and $u_i \equiv 1$. Quantitative results on iteration number, CPU running time and objective value are listed in Table 1, from which we

²These three problems are publicly available at <http://www.cise.ufl.edu/research/sparse/matrices/>

³<http://www.cs.ubc.ca/~schmidtm/Software/PQN.html>

⁴<http://tresnei.de.unifi.it/>

can observe that to achieve an exact solution (up to the machine precision to solve linear systems), PLS-DN typically needs notably fewer iterations and less CPU running time. Figure 1 shows the evolving curves of objective value as functions of iterations for different algorithms. It can be observed from these curves that PLS-DN, PQN and TRESNEI all converge rapidly during early stage. Despite similar sharp convergence behaviors, it is shown in Table 1 that in all cases but one (*well1850*) PLS-DN stops much earlier than the other methods. To conclude, PLS-DN is an efficient Newton-type solver to find the exact (or extremely high accurate) solution of BCLS.

Table 1: The quantitative results for different BCLS algorithms on Harwell Boeing collection. In order to return comparative objective values in (27), we use the following key parameters on output accuracy: for PQN solver, “optTol”, “suffDec” and “SPGoptTol” in PQN are both set 10^{-8} ; for TRESNEI solver, we set “tol.F” and “tol.opt” to be 10^{-6} . For all the comparing iterative methods, the initial points are set to be $\mathbf{x}^{(0)} = \mathbf{1}$.

Methods	<i>add20</i>			<i>illc1850</i>			<i>well1850</i>		
	it	cpu (sec.)	obj	it	cpu (sec.)	obj	it	cpu (sec.)	obj
PLS-DN	103	20.85	5.05×10^{-7}	32	0.69	5.66×10^{-4}	10	0.18	5.64×10^{-6}
lsqlin	156	34.77	2.37×10^{-6}	38	1.66	5.66×10^{-4}	18	0.47	5.64×10^{-6}
PQN	147	2.46	2.31×10^{-5}	607	5.78	6.06×10^{-4}	274	2.51	1.79×10^{-5}
TRESNEI	1715	23.53	3.47×10^{-6}	2831	43.35	5.66×10^{-4}	6	0.09	5.64×10^{-6}

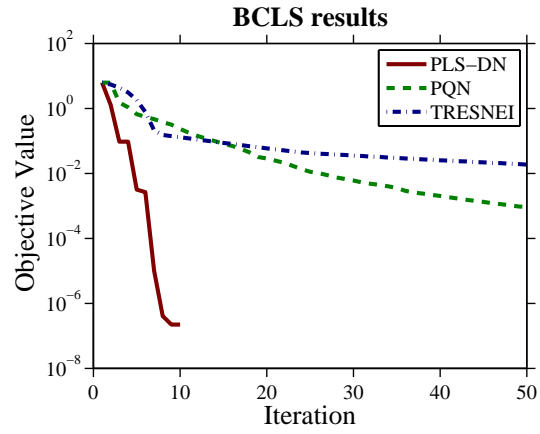
Particularly, when $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} = \infty$ the BCLS problem becomes a non-negative least squares (NNLS) problem which is widely applied in machine learning and computer vision, e.g., non-negative matrix factorization and non-negative image restoration. In this case, the KKT conditions (28) reduce to the following LCP problem:

$$\boldsymbol{\alpha} \geq \mathbf{0}, \quad \boldsymbol{\alpha}'\mathbf{w} = 0, \quad \boldsymbol{\alpha} = \mathbf{A}\mathbf{A}'\mathbf{w} - \mathbf{A}\mathbf{b},$$

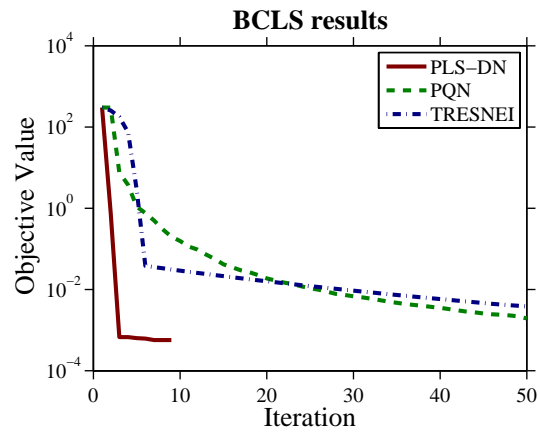
which as aforementioned can be solved with some off-the-shelf LCP solvers. To further evaluate the performance of PLS-DN for solving NNLS, we have compared PLS-DN with the following two representative LCP solvers :

- A damped Newton solver based on (Fischer, 1995)⁵ which we call LCP-Fischer in our test.

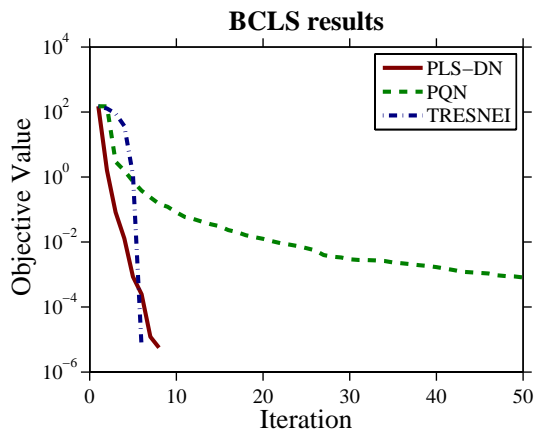
⁵<http://alice.nc.huji.ac.il/~tassa/pmwiki.php?n=Main.Code>



(a) *add20*



(b) *illc1850*



(c) *well1850*

Figure 1: Comparison of objective value versus number of iterations (only the first 50 steps are shown) for different BCLS algorithms. Note that the curves of `lsqlin` are not included here since this Matlab routine does not output intermediate results.

- A Lemke’s pivoting solver based on (Cottle et al., 1992)⁶ which we call LCP-Lemke in our test.

Quantitative results are listed in Table 2, from which we make the following observations: (i) On all these three problems, our PLS-DN method terminates within 10 iterations, and consistently achieves the best performance both in running time and solution accuracy; (ii): PLS-DN stops much earlier than the semi-smooth Newton LCP solver LCP-Fischer to achieve the exact solution. Figure 2 shows the evolving curves of objective value as functions of iterations for different NNLS solvers. It can be observed from these curves that both PLS-DN and LCP-Fischer exhibit sharp convergence behavior, but from Table 1 PLS-DN terminates much earlier than the two LCP solvers. To conclude, PLS-DN is an efficient and exact NNLS solver.

Table 2: The quantitative results for different NNLS solvers on Harwell Boeing collection. In order to return comparative objective values in (27), we use the following key parameters on output accuracy: for LCP-Lemke solver, “piv_tol” and “zer_tol” are both set 10^{-10} ; for LCP-Fischer solver, “tol” is 10^{-9} . For all the comparing iterative methods, the initial points are set to be $\mathbf{x}^{(0)} = \mathbb{1}$.

Methods	<i>add20</i>			<i>illc1850</i>			<i>well1850</i>		
	it	cpu (sec.)	obj	it	cpu (sec.)	obj	it	cpu (sec.)	obj
PLS-DN	10	0.90	2.22×10^{-7}	9	0.05	5.66×10^{-4}	8	0.05	5.64×10^{-6}
LCP-Fischer	433	285.41	2.38×10^{-5}	649	46.34	5.80×10^{-4}	308	22.37	5.64×10^{-6}
LCP-Lemke	2332	159.31	2.23×10^{-7}	749	7.30	7.10×10^{-3}	725	4.91	5.64×10^{-6}

4.2 App-II: Elitist Lasso

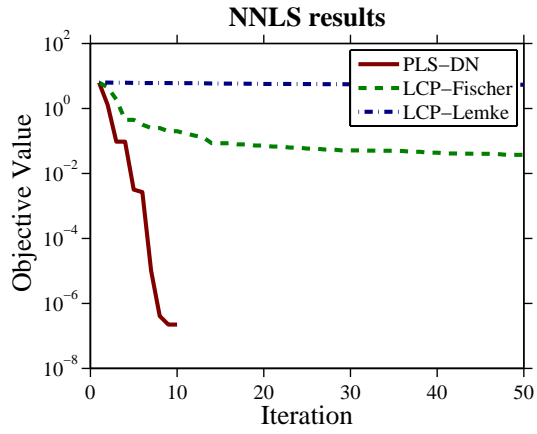
Denote \mathcal{G} a set of feature index groups with $|\mathcal{G}| = K$. Let us consider in our notation the *elitist Lasso* (eLasso) problem ⁷ (Kowalski & Torreesani, 2008) defined over \mathcal{G} :

$$\min_{\mathbf{w} \in \mathbb{R}^d} \sum_{i=1}^n L(b_i, \mathbf{w}'\mathbf{a}_i) + \frac{\lambda}{2} \sum_{g \in \mathcal{G}} \|\mathbf{w}_g\|_1^2, \quad (29)$$

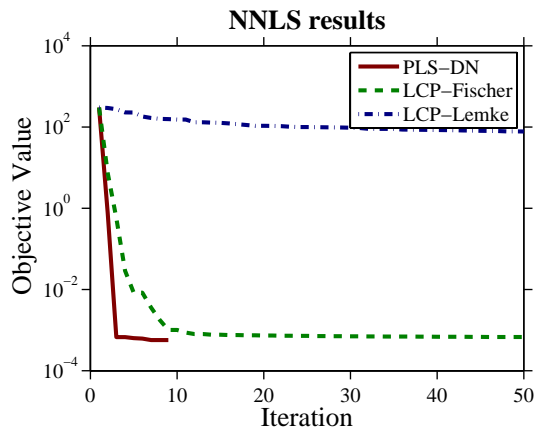
where $L(\cdot, \cdot)$ is a smooth convex loss function. In opposite to group Lasso (Yuan & Lin, 2006) which encourages the sparsity at group level, eLasso will encourage the

⁶http://people.sc.fsu.edu/~jburkardt/m_src/lemke/lemke.html

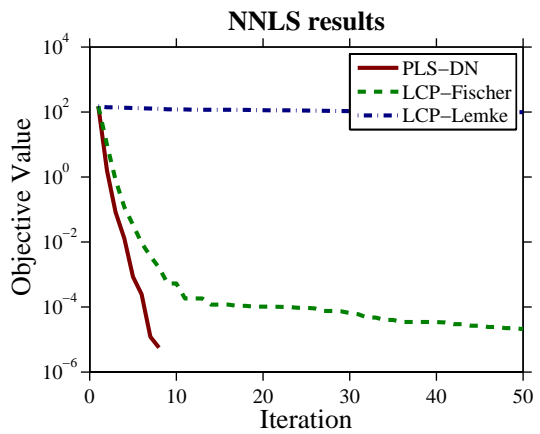
⁷The elitist Lasso is also called as *exclusive Lasso* (Zhou et al., 2010) in scenario of multitask learning.



(a) *add20*



(b) *illc1850*



(c) *well1850*

Figure 2: Comparison of objective value versus number of iterations for different BCLS algorithms. Note that the curves of `lsqlin` and `lsqnonneg` are not included here since both Matlab routines do not output intermediate results.

exclusive selection of features inside each group, and thus is particularly useful to capture the negative correlation among features. Different from the existing formulation in which any groups $g_i, g_j \in \mathcal{G}$ are required to be disjoint (Kowalski & Torreesani, 2008; Zhou et al., 2010), here we allow group overlaps which is useful for exclusive feature selection where features may belong to different groups.

4.2.1 Proximity Operator as PLSs

One issue of eLasso with group overlaps is optimization. Since convex objective in (29) is the composite of a smooth term and a non-smooth term, we resort to proximal algorithms (Tseng, 2008) for optimization. Resolving such kind of problem relies on proximity operator (Combettes & Pesquet, 2007), which in our case is given by

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{z}\|^2 + \frac{\lambda}{2} \sum_{g \in \mathcal{G}} \|\mathbf{w}_g\|_1^2. \quad (30)$$

Equivalently, we may reformulate the preceding proximity operator as

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{z}\|^2 + \frac{\lambda}{2} |\mathbf{w}' \mathbf{Q} \mathbf{w}|,$$

where $|\mathbf{w}| = (|w_i|)$, and matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ is given by

$$\mathbf{Q} = \sum_{g \in \mathcal{G}} \mathbf{Q}_g, \quad \mathbf{Q}_g(i, j) = \begin{cases} 1, & i, j \in g, \\ 0, & \text{otherwise.} \end{cases}$$

The following result indicates that the proximity operator can be reformulated as solving a problem of non-degenerate PLSs.

Proposition 3. *The optimizer \mathbf{w}^* of proximity operator (30) is given by*

$$w_i^* := \text{sign}(z_i) \max\{0, x_i^*\},$$

where $\mathbf{x}^* = (x_i^*)$ is the solution of the following PLSs

$$\min\{\mathbf{0}, \mathbf{x}\} + (\lambda \mathbf{Q} + \mathbf{I}) \max\{\mathbf{0}, \mathbf{x}\} = |\mathbf{z}|.$$

Proof. Since the objective function in (30) is convex, its optimal solution \mathbf{w}^* is fully characterized by the Karush-Kuhn-Tucher conditions (see, e.g., Boyd & Vandenberghe, 2004)

$$w_i^* - z_i + \lambda(\mathbf{Q}|\mathbf{w}^*|)_i \xi_i = 0, \forall i \in \mathcal{I},$$

where $\xi_i := \partial |\cdot| (w_i^*) = \text{sign}(w_i^*)$ if $w_i^* \neq 0$ and $\partial |\cdot| (0) \in [-1, 1]$ is a subgradient of the absolute function $|\cdot|$ evaluated at w_i^* . By standard result of soft-thresholding method (Donoho, 1995) we have that

$$|w_i^*| = \max\{0, |z_i| - \lambda(\mathbf{Q}|\mathbf{w}^*)_i\}, \forall i \in \mathcal{I}.$$

Denote $s_i := (\mathbf{Q}|\mathbf{w}^*)_i$ and $x_i := |z_i| - \lambda s_i$. By the preceding equation we have $|\mathbf{w}^*| = \max\{\mathbf{0}, \mathbf{x}\}$. Since $\mathbf{s} = \mathbf{Q}|\mathbf{w}^*$ and $\mathbf{x} = |\mathbf{z}| - \lambda\mathbf{s}$, we get

$$\mathbf{x} + \lambda\mathbf{Q}\max\{\mathbf{0}, \mathbf{x}\} = |\mathbf{z}|,$$

or equivalently

$$\min\{\mathbf{0}, \mathbf{x}\} + (\lambda\mathbf{Q} + \mathbf{I})\max\{\mathbf{0}, \mathbf{x}\} = |\mathbf{z}|.$$

which is a PLSs($|\mathbf{z}|, \lambda\mathbf{Q} + \mathbf{I}, \mathbf{0}, \infty$) problem. \square

For any $\lambda > 0$, the coefficient matrix $\mathbf{T} = \lambda\mathbf{Q} + \mathbf{I}$ is positive-definite, which implies non-degenerate. We can apply the modified PLS-DN in Algorithm 2 to solve the proximity operator (30) in finite iterations. By incorporating such an operator into an accelerated proximal gradient (APG) algorithm (Tseng, 2008), we can efficiently solve the eLasso problem with group overlaps.

It is noteworthy that one intuitive strategy to solve the eLasso with overlaps is to explicitly duplicate variables as applied in (Jacob et al., 2009). However, when overlap is severe, such a duplication strategy will significantly increase the number of variables involved in optimization, and thus degenerate the efficiency. Differently, our method is operated on the original variables and thus is insensitive to the extent of overlap.

4.2.2 Simulation

We now exhibit numerical effects of PLS-DN for solving eLasso on a synthetic data set. We consider the linear regression model, i.e., $L(b_i, \mathbf{w}'\mathbf{a}_i) := \frac{1}{2}\|b_i - \mathbf{w}'\mathbf{a}_i\|^2$. For this experiment, the input variable dimension is $d = 1000$, the sample number is $n = 100$. We set the support of \mathbf{w} to the first half of the input features. Each support feature w_i is uniformly valued in interval $[1, 2]$. The noise in linear model is i.i.d. Gaussian with mean 0 and variance 1. A total $K = 100$ number of groups of potentially exclusive features are generated as follows: we randomly select 50 support features and 100 non-support features to form each group. These generated groups are typically overlapping.

Figure 3(a) shows the number of PLS-DN iterations for each step of proximity operator optimization, which is observed to never exceed 4. The sparsity of the recovered feature weights are shown in Figure 3(b). From these results we can see that PLS-DN is efficient and effective for optimizing the eLasso with group overlaps.

4.3 App-III: Support Vector Machines in Primal

We now show that one of the most popular machine learning algorithms, the support vector machines (SVMs) (Cortes & Vapnik, 1995), can also be numerically modeled as PLSs. A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. We refer the interested readers to an excellent tutorial on SVMs by Burges (1998) and the references therein.

Let us consider binary linear SVMs with classification function $f(\mathbf{a}|\mathbf{w}, w_0) = \mathbf{w}'\mathbf{a}_i + w_0$. The parameters can be learned through solving the following regularized empirical risk suffered from quadratic hinge loss:

$$\min_{\mathbf{w}, w_0} \sum_{i=1}^n L(b_i, \mathbf{w}'\mathbf{a}_i + w_0) + \lambda \|\mathbf{w}\|^2, \quad (31)$$

where $b_i \in \{+1, -1\}$ and $L(y, t) = \max(0, 1 - yt)^2$. Herein, we consider the non-linear SVMs with a kernel function $k(\cdot, \cdot)$ and an associated Reproducing Kernel Hilbert Space (RKHS) \mathcal{H} . The well known Representer Theorem (Kimeldorf & Wahba, 1970) states that the optimal f exists in \mathcal{H} and can be written as a linear combination of kernel functions evaluated at the training samples. Therefore, we seek for a solution of the form

$$f(\mathbf{a}|\boldsymbol{\beta}) = \sum_{i=1}^n \beta_i k(\mathbf{a}_i, \mathbf{a}).$$

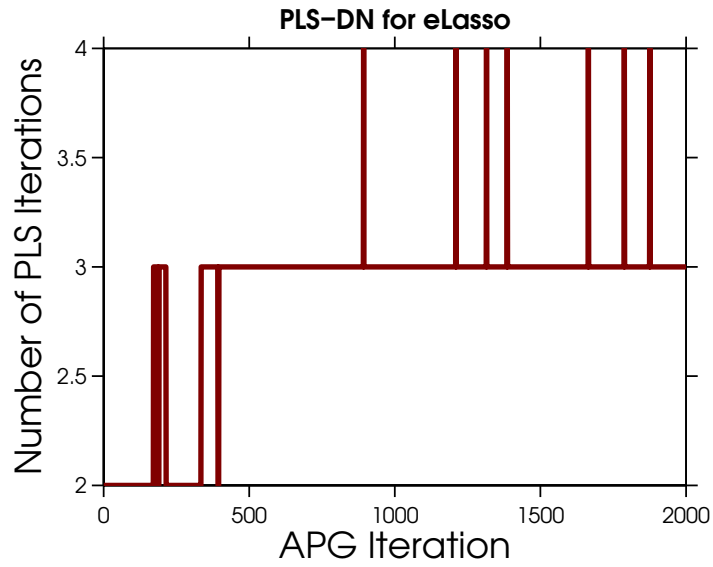
Let us convert the linear SVMs (31) to its non-linear form in terms of $\boldsymbol{\beta}$ as

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n L\left(b_i, \sum_{j=1}^n \beta_j k(\mathbf{a}_j, \mathbf{a}_i)\right) + \lambda \sum_{i,j=1}^n \beta_i \beta_j k(\mathbf{a}_i, \mathbf{a}_j), \quad (32)$$

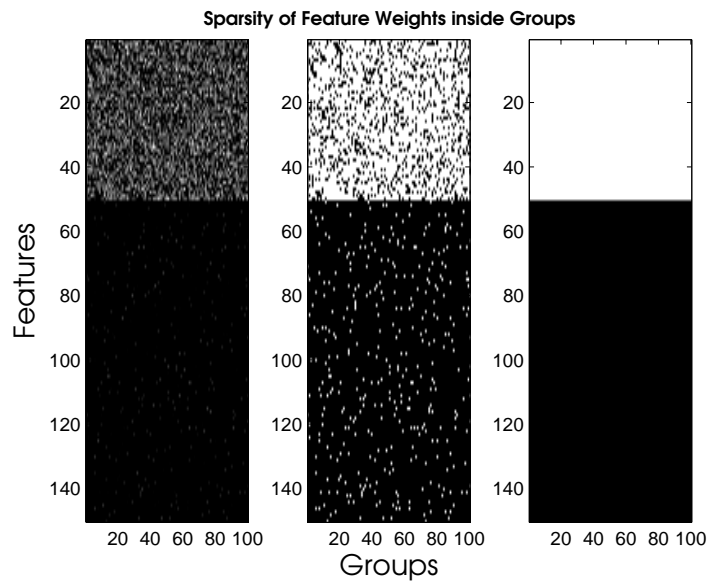
or in a more compact form written as

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n L(b_i, \mathbf{K}'_{i\bullet} \boldsymbol{\beta}) + \lambda \boldsymbol{\beta}' \mathbf{K} \boldsymbol{\beta}, \quad (33)$$

where \mathbf{K} the kernel matrix with $K_{ij} = k(\mathbf{a}_i, \mathbf{a}_j)$ and Let us denote $\mathbf{K}_{i\bullet}$ the i th column of \mathbf{K} . The problem (33) is known as Primal SVMs (Prim-SVMs) in (Chappelle, 2007).



(a)



(b)

Figure 3: Results of PLS-DN for solving eLasso with overlaps on a synthetic problem. (a): Number of PLS-DN iterations for proximity operator as a function of APG iterate counts. (b): Left: the recovered feature weights w^* inside each feature group. Middle: the sparsity pattern of vector w^* inside each feature group. Right: the sparsity pattern of the ground truth w inside each feature group.

4.3.1 Solving Prim-SVMs as PLSs

The following result connects Prim-SVMs to PLSs.

Proposition 4. *Assume that \mathbf{K} is invertible. Let $\mathbf{B} := \text{diag}(\mathbf{b})$. The optimizer β^* of (32) is given by*

$$\beta^* = \lambda^{-1} \mathbf{B} \max\{\mathbf{0}, \mathbf{x}^*\}, \quad (34)$$

where \mathbf{x}^* is the solution of the following PLSs

$$\min\{\mathbf{0}, \mathbf{x}\} + (\lambda^{-1} \mathbf{B} \mathbf{K} \mathbf{B} + \mathbf{I}) \max\{\mathbf{0}, \mathbf{x}\} = \mathbf{1}. \quad (35)$$

Proof. Recall that $L(y, t)$ is the quadratic hinge loss, thus is differentiable. By setting the derivative of the objective in (33) to zero we get the following systems

$$-\sum_{i=1}^n \max\{\mathbf{0}, 1 - b_i \mathbf{K}'_{i\bullet} \beta\} b_i \mathbf{K}_{i\bullet} + \lambda \mathbf{K} \beta = \mathbf{0}. \quad (36)$$

Let us denote

$$\mathbf{x} := \mathbf{1} - \mathbf{B} \mathbf{K} \beta. \quad (37)$$

Trivial manipulation on (36) leads to

$$\mathbf{x} + \lambda^{-1} \mathbf{B} \mathbf{K} \mathbf{B} \max\{\mathbf{0}, \mathbf{x}\} = \mathbf{1}, \quad (38)$$

or equivalently

$$\min\{\mathbf{0}, \mathbf{x}\} + (\lambda^{-1} \mathbf{B} \mathbf{K} \mathbf{B} + \mathbf{I}) \max\{\mathbf{0}, \mathbf{x}\} = \mathbf{1}.$$

Since \mathbf{K} is invertible, by (37) the solution β^* of (36) is calculated as

$$\beta^* = \mathbf{K}^{-1} \mathbf{B}^{-1} (\mathbf{1} - \mathbf{x}^*) = \lambda^{-1} \mathbf{B} \max\{\mathbf{0}, \mathbf{x}^*\},$$

where the second equality follows (38). □

Since \mathbf{K} is positive-semidefinite, $\lambda^{-1} \mathbf{B} \mathbf{K} \mathbf{B} + \mathbf{I}$ is a positive-definite matrix, i.e., non-degenerate. Therefore we can apply PLS-DN to obtain solution \mathbf{x}^* to (35). The expression (34) clearly indicates the sparse nature of β^* .

Notice that a similar Newton-type optimization method for solving the Prim-SVMs (33) has been proposed by Chappelle (2007), which solves the systems (36) via a Newton-type iterative scheme

$$\beta^{(k+1)} = (\lambda \mathbf{I} + \mathbf{P}^{(k)} \mathbf{K})^{-1} \mathbf{P}^{(k)} \mathbf{b},$$

where

$$\mathbf{P}^{(k)} := \begin{bmatrix} p(\beta_1^{(k)}) & & & \\ & \ddots & & \\ & & & p(\beta_n^{(k)}) \end{bmatrix},$$

where $p(\beta_1^{(k)}) = 1$ if $1 - b_i \mathbf{K}'_{i,\bullet} \boldsymbol{\beta}^{(k)} \geq 0$, and 0 otherwise. It has been empirically validated that Chappelle’s primal solver is quite competitive to LIBSVM (Chang & Lin, 2001), one of representative dual SVMs solvers. Although converge extremely fast in practice, the algorithmic analysis for Chappelle’s solver is incomplete in two aspects: 1) the non-smoothness of gradient equation systems (36) is neglected when calculating the Hessian; 2) the global convergence and finite termination properties are not explicitly addressed in a rigorous way. Our PLS-DN method, up to an affine transform (37), can be regarded as a globalization of Chappelle’s method with finite termination guarantee. Similar to the definition in (Chappelle, 2007), we say a point \mathbf{a}_i is a support vector if $1 - b_i \mathbf{K}'_{i,\bullet} \boldsymbol{\beta} > 0$, i.e., the loss on this point is non-zero.

4.3.2 Simulation

We have conducted a group of numerical experiments to compare PLS-DN with Chappelle’s method in terms of efficiency and accuracy for solving the gradient equation systems (36). We use seven binary classification tasks publicly available at <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>. The statistics of data sets are described in the left part of Table 3. For each data set, we construct the radial basis function (RBF) heat kernel, i.e., $k(\mathbf{a}_i, \mathbf{a}_j) := \exp(-\|\mathbf{a}_i - \mathbf{a}_j\|^2)/t$ where t is the temperature parameter. The settings of parameter λ are given in the middle of Table 3. To further accelerate the computation for data set larger than 1000, we apply a similar recursive down sampling strategy as applied in (Chappelle, 2007). The quantitative results are listed in the right part of Table 3. From these results we can observe that PLS-DN performs equally efficient and accurate as Chappelle’s method. This is as expected since both PLS-DN and Chappelle’s method are essentially finite Newton methods for training Prim-SVMs.

Table 3: The left part lists statistics of data sets. The middle part lists setting of parameters λ . The right part lists the quantitative results by PLS-DN and Chappelle’s method for solving the gradient equation systems (36). Here “sv” abbreviates for the number of *support vectors*.

Datasets	Sizes	Dim.	λ	PLS-DN				Chappelle’s method			
				it	cpu (sec.)	obj	sv	it	cpu (sec.)	obj	sv
a5a	6,414	123	10^{-5}	15	11.97	2.08×10^{-12}	2265	17	15.03	3.08×10^{-9}	2265
a6a	11,220	123	10^{-5}	15	48.97	1.39×10^{-7}	4041	16	61.29	4.16×10^{-9}	4041
w3a	4,912	300	10^{-5}	14	2.39	1.75×10^{-9}	786	14	2.01	2.50×10^{-8}	786
w5a	9,888	300	10^{-5}	16	16.51	2.95×10^{-6}	1511	16	13.97	8.58×10^{-6}	1511
svmguidel	3,089	4	10^{-3}	9	0.81	4.45×10^{-16}	691	10	0.77	4.37×10^{-12}	691
splice	1,000	60	10^{-3}	6	0.16	2.48×10^{-17}	503	7	0.26	2.03×10^{-18}	503
mushrooms	8,124	112	10^{-3}	12	2.29	4.36×10^{-19}	443	13	2.56	3.05×10^{-20}	443

5 Conclusion

This paper addressed the non-degenerate PLSs which are a powerful tool for modeling many problems deriving from the practice of machine learning. To solve the non-degenerate PLSs, we have proposed the PLS-DN algorithm which is a damped Newton method with global convergence behavior and finite termination guarantee. The rate of local convergence before algorithm termination is established to be at least linear. The existence and uniqueness of solution of non-degenerate PLSs are guaranteed when \mathbf{T} is a P -matrix. We apply non-degenerate PLSs to numerically model several concrete statistical learning problems such as box-constrained least squares, elitist Lasso, and support vector machines. Extensive comparing experiments on several benchmark tasks show that PLS-DN is an efficient and accurate solver for non-degenerate PLSs in learning problems.

Appendix

A Technical Proofs

A.1 Proof of Theorem 1

The goal of this appendix section is to prove Theorem 1.

Proof. Part (a): Let \mathbf{y} , $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ be a solution of systems (3). Let

$$\mathbf{x} := \mathbf{y} - \boldsymbol{\alpha} + \boldsymbol{\beta}.$$

We now check that the following relation holds

$$\max\{\mathbf{1}, \min\{\mathbf{u}, \mathbf{x}\}\} = \mathbf{y}. \quad (\text{A.1})$$

To see this, let us first assume $l_i < u_i$ and distinguish the following three cases

- (i) $y_i = l_i$. From $\beta_i(y_i - u_i) = 0$ we get $\beta_i = 0$. Therefore $x_i = y_i - \alpha_i + \beta_i \leq y_i = l_i$, which implies that $\max\{l_i, \min\{u_i, x_i\}\} = l_i = y_i$.
- (ii) $y_i = u_i$. By similar argument in (i) we get that $\max\{l_i, \min\{u_i, x_i\}\} = u_i = y_i$.
- (iii) $l_i < y_i < u_i$. From $\alpha_i(y_i - l_i) = 0$ and $\beta_i(y_i - u_i) = 0$, we get $\alpha_i = \beta_i = 0$. Therefore, $x_i = y_i$ and $\max\{l_i, \min\{u_i, x_i\}\} = y_i$.

If $l_i = u_i$, then obviously $y_i = l_i$ and thus $\max\{l_i, \min\{u_i, x_i\}\} = l_i = y_i$.

Since $\boldsymbol{\alpha} - \boldsymbol{\beta} = \mathbf{T}\mathbf{y} - \mathbf{b}$, by (A.1) and (A.1) it holds that

$$\mathbf{x} + (\mathbf{T} - \mathbf{I}) \max\{\mathbf{1}, \min\{\mathbf{u}, \mathbf{x}\}\} = \mathbf{b}.$$

which proves the part (a).

Part (b): Let \mathbf{x} be a solution of systems (2). Let

$$\mathbf{y} := \max\{\mathbf{1}, \min\{\mathbf{u}, \mathbf{x}\}\}, \quad \boldsymbol{\alpha} := \max\{\mathbf{0}, \mathbf{1} - \mathbf{x}\}, \quad \boldsymbol{\beta} := \max\{\mathbf{0}, \mathbf{x} - \mathbf{u}\} \quad (\text{A.2})$$

By definition it holds that

$$\mathbf{1} \leq \mathbf{y} \leq \mathbf{u}, \quad \boldsymbol{\alpha}, \boldsymbol{\beta} \geq \mathbf{0}. \quad (\text{A.3})$$

Moreover, it is easy to check that

$$\boldsymbol{\alpha}'(\mathbf{y} - \mathbf{1}) = 0, \quad \boldsymbol{\beta}'(\mathbf{y} - \mathbf{u}) = 0, \quad \boldsymbol{\alpha} - \boldsymbol{\beta} = \mathbf{y} - \mathbf{x}. \quad (\text{A.4})$$

Since \mathbf{x} solves (2), it follows that

$$\mathbf{T}\mathbf{y} - \mathbf{b} = \mathbf{y} - \mathbf{x}. \quad (\text{A.5})$$

By combining (A.3)~(A.5) we can see that $(\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ defined in (A.2) solves systems (3). \square

A.2 Proof of Proposition 1

The goal of this appendix section is to prove Proposition 1.

Proof. Since $\mathbf{x}^{(k)}$ is non-degenerate, the (10) holds. Combining this with B-differential (6) yields

$$BF(\mathbf{x}^{(k)}; \Delta\mathbf{x}) = [\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})]\Delta\mathbf{x}.$$

By (9) and the preceding equation we may write the generalized Newton equation (8) as

$$(\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})) \Delta\mathbf{x} = -(\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})) \mathbf{x}^{(k)} + \mathbf{c}^{(k)},$$

where $\mathbf{c}^{(k)} := \mathbf{b} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)}\mathbf{1} - \mathbf{Q}^{(k)}\mathbf{u} - \mathbf{1})$. By assumption that $\mathbf{I} + (\mathbf{T} - \mathbf{I})(\mathbf{P}^{(k)} - \mathbf{Q}^{(k)})$ is non-singular, we arrive at (11). \square

A.3 Proof of Theorem 2

The goal of this appendix section is to prove Theorem 2

Proof. Part (a): From (S.3) in Algorithm 2, with triangle inequality we get that

$$\begin{aligned} \|F(\mathbf{x}^{(k+1)})\| &\leq \|F(\mathbf{x}^{(k+1)}) - F(\tilde{\mathbf{x}}^{(k+1)})\| + \|F(\tilde{\mathbf{x}}^{(k+1)})\| \\ &\leq L\sqrt{d}\delta^{(k+1)} + \|F(\tilde{\mathbf{x}}^{(k+1)})\| \\ &\leq L\sqrt{d}\delta^{(k+1)} + \sqrt{1 - t_k\sigma} \|F(\mathbf{x}^{(k)})\| \end{aligned}$$

where the second inequality follows the Lipschitz continuity of F and the last inequality follows (13). By choosing $0 < \delta^{(k+1)} \leq \frac{(1-\sqrt{1-t_k\sigma})\|F(\mathbf{x}^{(k)})\|}{2L\sqrt{d}}$, we get that

$$\|F(\mathbf{x}^{(k+1)})\| \leq \frac{1 + \sqrt{1-t_k\sigma}}{2} \|F(\mathbf{x}^{(k)})\| < \|F(\mathbf{x}^{(k)})\|. \quad (\text{A.6})$$

Part (b): From (a) the sequence $\{\|F(\mathbf{x}^{(k)})\|\}_{k \geq 0}$ is non-negative and strictly decreasing. Thus it converges, and

$$\lim_{k \rightarrow \infty} (\|F(\mathbf{x}^{(k)})\| - \|F(\mathbf{x}^{(k+1)})\|) = 0. \quad (\text{A.7})$$

By (A.6)

$$\|F(\mathbf{x}^{(k)})\| - \|F(\mathbf{x}^{(k+1)})\| \geq \frac{1 - \sqrt{1-t_k\sigma}}{2} \|F(\mathbf{x}^{(k)})\|$$

which together with (A.7) implies that

$$\lim_{k \rightarrow \infty} \frac{1 - \sqrt{1-t_k\sigma}}{2} \|F(\mathbf{x}^{(k)})\| = 0.$$

If $\liminf t_k$ is positive, then

$$\|F(\mathbf{x}^*)\| = \lim_{k \rightarrow \infty} \|F(\mathbf{x}^{(k)})\| = 0.$$

□

A.4 Proof of Theorem 3

The goal of this appendix section is to prove Theorem 3. We first introduce the concept of strongly BD -regular (BD for B -derivative) for a function $G : \mathbb{R}^d \mapsto \mathbb{R}^d$, which is key to derive the convergence rate of semi-smooth Newton methods.

Definition 3 (Strongly BD -regular). Let D_G be the set where G is differentiable. Denote

$$\partial_B G(\mathbf{x}) := \left\{ \lim_{\mathbf{x}^{(k)} \in D_G, \mathbf{x}^{(k)} \rightarrow \mathbf{x}} \nabla G(\mathbf{x}^{(k)}) \right\}$$

the B -subdifferential of G at \mathbf{x} . We say that G is strongly BD -regular at \mathbf{x} if all $\mathbf{P} \in \partial_B G(\mathbf{x})$ are non-singular.

Lemma 7. If matrix \mathbf{T} is non-degenerate, then function F defined in (5) is strongly BD -regular at any point \mathbf{x} .

Proof. Trivial algebraic manipulation shows that at any \mathbf{x}

$$\partial_B F(\mathbf{x}) = \{\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{R}\},$$

where

$$\mathbf{R} \in \partial_B \max\{\mathbf{1}, \min\{\mathbf{u}, \mathbf{x}\}\} = \{\text{diag}(r_1, \dots, r_d)\}$$

with $r_i, i = 1, \dots, d$ are given by:

$$r_i = \begin{cases} 1 & \text{if } l_i < x_i < u_i \\ 0 \text{ or } 1 & \text{if } x_i = l_i \text{ or } x_i = u_i \\ 0 & \text{if } x_i < l_i \text{ or } x_i > u_i \end{cases}.$$

The result obviously holds for $\mathbf{R} = \mathbf{0}$. Now suppose that $\mathbf{R} \neq \mathbf{0}$, then we define the index sets

$$J := \{i \in \mathcal{I} : r_i = 1\} \text{ and } \bar{J} := \mathcal{I} \setminus J. \quad (\text{A.8})$$

Obviously $J \neq \emptyset$. Let $\mathbf{z} \in \mathbb{R}^d$ such that $(\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{R})\mathbf{z} = \mathbf{0}$. The definitions of \mathbf{R} , J and \bar{J} yield

$$\begin{aligned} \mathbf{T}_{JJ}\mathbf{z}_J &= \mathbf{0}, \\ \mathbf{z}_{\bar{J}} + \mathbf{T}_{\bar{J}J}\mathbf{z}_J &= \mathbf{0}. \end{aligned}$$

Following the same arguments in the proof of Theorem 6 (see (18) and (19)) we obtain that $\mathbf{z}_J = \mathbf{0}$ and $\mathbf{z}_{\bar{J}} = \mathbf{0}$. Consequently, $\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{R}$ is non-singular. \square

To prove Theorem 3, we need the following lemma which is a direct consequence of the preceding Lemma 7 and the Corollary 3.4 in (Qi, 1993) on the function F at \mathbf{x}^* .

Lemma 8. *Suppose that \mathbf{x}^* is a zero of F and \mathbf{T} is non-degenerate. For any $\epsilon > 0$, there is a $\rho > 0$ such that for all \mathbf{x} with $\|\mathbf{x} - \mathbf{x}^*\| \leq \rho$, if the generalized Newton equation*

$$BF(\mathbf{x}; \Delta\mathbf{x}) = -F(\mathbf{x})$$

is solvable for $\Delta\mathbf{x}$, then

$$\begin{aligned} \|\mathbf{x} + \Delta\mathbf{x} - \mathbf{x}^*\| &\leq \epsilon \|\mathbf{x} - \mathbf{x}^*\|, \\ \|F(\mathbf{x} + \Delta\mathbf{x})\| &\leq \epsilon \|F(\mathbf{x})\|. \end{aligned}$$

We are now in the position to prove Theorem 3.

Proof of Theorem 3. Let $\bar{\mathbf{x}}^{(k+1)} := \mathbf{x}^{(k)} + \Delta \mathbf{x}^{(k)}$. By Lemma 8, there exists a $\rho > 0$ such that for all $\mathbf{x}^{(k)}$ with $\|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \rho$,

$$\begin{aligned}\|\bar{\mathbf{x}}^{(k+1)} - \mathbf{x}^*\| &\leq \sqrt{1 - \sigma} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|, \\ \|F(\bar{\mathbf{x}}^{(k+1)})\| &\leq \sqrt{1 - \sigma} \|F(\mathbf{x}^{(k)})\|.\end{aligned}$$

Therefore,

$$\|F(\bar{\mathbf{x}}^{(k+1)})\|^2 \leq (1 - \sigma) \|F(\mathbf{x}^{(k)})\|^2.$$

By (S.2) of Algorithm 2 we have that

$$t_k = 1 \quad \text{and} \quad \tilde{\mathbf{x}}^{(k+1)} = \mathbf{x}^{(k)} + \Delta \mathbf{x}^{(k)} = \bar{\mathbf{x}}^{(k+1)}. \quad (\text{A.9})$$

The choice of perturbation $\delta^{(k+1)}$ ensures that

$$\begin{aligned}\delta^{(k+1)} &\leq \frac{(1 - \sqrt{1 - t_k \sigma}) \|F(\mathbf{x}^{(k)})\|}{2L\sqrt{d}} \\ &\leq \frac{(1 - \sqrt{1 - \sigma}) \|\mathbf{x}^{(k)} - \mathbf{x}^*\|}{2\sqrt{d}},\end{aligned} \quad (\text{A.10})$$

where the second inequality follows by considering $t_k \leq 1$, $F(\mathbf{x}^*) = 0$ and the Lipschitz-continuity. By triangle inequality and the perturbation operation in (S.3) in Algorithm 2,

$$\begin{aligned}\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| &\leq \|\mathbf{x}^{(k+1)} - \tilde{\mathbf{x}}^{(k+1)}\| + \|\tilde{\mathbf{x}}^{(k+1)} - \mathbf{x}^*\| \\ &\leq \sqrt{d}\delta^{(k+1)} + \sqrt{1 - \sigma} \|\mathbf{x}^{(k)} - \mathbf{x}^*\| \\ &\leq \frac{1 + \sqrt{1 - \sigma}}{2} \|\mathbf{x}^{(k)} - \mathbf{x}^*\| \leq \rho,\end{aligned} \quad (\text{A.11})$$

where the last but one inequality follows (A.10). Since \mathbf{x}^* is a limiting point of $\{\mathbf{x}^{(k)}\}_{k \geq 0}$, there is a $k(\rho)$ such that $\|\mathbf{x}^{(k(\rho))} - \mathbf{x}^*\| \leq \rho$. By introduction of above arguments, (A.9) and (A.11) hold for any $k \geq k(\rho)$. Therefore, the entire sequence $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ converges to \mathbf{x}^* and t_k eventually becomes 1. From (A.11) we can see that the convergence rate is linear for any $\sigma \in (0, 1)$.

Moreover, when $k \geq k(\rho)$, we have that

$$\begin{aligned}
\|F(\mathbf{x}^{(k+1)})\| &\leq \|F(\mathbf{x}^{(k+1)}) - F(\tilde{\mathbf{x}}^{(k+1)})\| + \|F(\tilde{\mathbf{x}}^{(k+1)})\| \\
&\leq L\sqrt{d}\delta_{(k+1)} + \|F(\tilde{\mathbf{x}}^{(k+1)})\| \\
&\leq \frac{1 - \sqrt{1 - \sigma}}{2} \|F(\mathbf{x}^{(k)})\| + \sqrt{1 - \sigma} \|F(\mathbf{x}^{(k)})\| \\
&\leq \frac{1 + \sqrt{1 - \sigma}}{2} \|F(\mathbf{x}^{(k)})\|,
\end{aligned}$$

which indicates that the objective value sequence $\{\|F(\mathbf{x}^{(k)})\|\}_{k \geq 0}$ converges at least linearly towards zero. \square

A.5 Proof of Lemma 3

The goal of this appendix section is to prove Lemma 3.

Proof. If there is at least one index $i \in \mathcal{I}$ with $x_i^* \neq l_i$, then set

$$\epsilon_l(\mathbf{x}^*) := \frac{1}{2} \min\{|x_i^* - l_i| : i \in \mathcal{I}, x_i^* \neq l_i\}.$$

Otherwise, let $\epsilon_p(\mathbf{x}^*)$ be any positive number. If there is at least one index $i \in \mathcal{I}$ with $x_i^* \neq u_i$, then set

$$\epsilon_u(\mathbf{x}^*) := \frac{1}{2} \min\{|x_i^* - u_i| : i \in \mathcal{I}, x_i^* \neq u_i\}.$$

Otherwise, let $\epsilon_u(\mathbf{x}^*)$ be any positive number. Set $\epsilon(\mathbf{x}^*) := \min(\epsilon_l(\mathbf{x}^*), \epsilon_u(\mathbf{x}^*))$.

Now, let $\mathbf{x} \in B_{\epsilon(\mathbf{x}^*)}$ and

$$\Delta_l(x_i) := (p(x_i) - p(x_i^*)) (x_i^* - l_i), \quad \Delta_u(x_i) := (q(x_i) - q(x_i^*)) (x_i^* - u_i).$$

If $l_i < u_i$, we distinguish the following three cases

- (i) If $x_i^* = l_i$, obviously $\Delta_l(x_i) = 0$. Meanwhile, $|x_i - x_i^*| \leq \epsilon(\mathbf{x}^*) \leq \epsilon_u(\mathbf{x}^*) < |x_i^* - u_i|$ which implies that $x_i \neq u_i$, and $x_i - u_i, x_i^* - u_i$ are of the same sign (or $|x_i - x_i^*| = |x_i - u_i| + |x_i^* - u_i| \geq |x_i^* - u_i|$). Therefore $q(x_i) = q(x_i^*)$, so that $\Delta_u(x_i) = 0$.
- (ii) If $x_i^* = u_i$, obviously $\Delta_u(x_i) = 0$. Meanwhile, $|x_i - x_i^*| \leq \epsilon(\mathbf{x}^*) \leq \epsilon_l(\mathbf{x}^*) < |x_i^* - l_i|$ which implies that $x_i \neq l_i$, and $x_i - l_i, x_i^* - l_i$ are of the same sign. Therefore $p(x_i) = p(x_i^*)$, so that $\Delta_l(x_i) = 0$.

- (iii) If $x_i^* \neq l_i$ and $x_i^* \neq u_i$, by similar argument in (i) and (ii) we obtain that $x_i - l_i$, $x_i^* - l_i$ are of the same sign and thus $p(x_i) = p(x_i^*)$, $\Delta_l(x_i) = 0$, and $x_i - u_i$, $x_i^* - u_i$ are of the same sign and thus $q(x_i) = q(x_i^*)$, so that $\Delta_u(x_i) = 0$.

If $l_i = u_i$, we distinguish the following two cases

- (i) If $x_i^* = l_i$, obviously $\Delta_l(x_i) = \Delta_u(x_i) = 0$.
- (ii) If $x_i^* \neq l_i$, we get $|x_i - x_i^*| \leq \epsilon(\mathbf{x}^*) \leq \epsilon_l(\mathbf{x}^*) < |x_i^* - u_i|$ which implies that $x_i \neq l_i$, and $x_i - l_i$, $x_i^* - l_i$ are of the same sign. Therefore $p(x_i) = q(x_i) = p(x_i^*) = q(x_i^*)$, so that $\Delta_l(x_i) = \Delta_u(x_i) = 0$.

Consequently, we have $\Delta_l(x_i) = \Delta_u(x_i) = 0$ for all $i \in \mathcal{I}$ and all $\mathbf{x} \in B_{\epsilon}(\mathbf{x}^*)$. \square

A.6 Proof of Lemma 4

The goal of this appendix section is to prove Lemma 4.

Proof. If $\mathbf{D} = \mathbf{O}$ then the result obviously holds. We now assume that $\mathbf{D} \neq \mathbf{O}$. Consequently we can always find an index set $J \neq \emptyset$ and $\bar{J} = \mathcal{I} \setminus J$ such that

$$\mathbf{D}_{JJ} \leq \mathbf{I}_{JJ}, \mathbf{D}_{JJ} \text{ is positive diagonal, and } \mathbf{D}_{\bar{J}\bar{J}} = \mathbf{O}. \quad (\text{A.12})$$

Let \mathbf{z} satisfy $(\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{D})\mathbf{z} = \mathbf{0}$. It holds that

$$(\mathbf{I}_{JJ} + (\mathbf{T}_{JJ} - \mathbf{I}_{JJ})\mathbf{D}_{JJ})\mathbf{z}_J = \mathbf{0}, \quad (\text{A.13})$$

$$\mathbf{z}_{\bar{J}} + (\mathbf{T}_{\bar{J}\bar{J}} - \mathbf{I}_{\bar{J}\bar{J}})\mathbf{D}_{\bar{J}\bar{J}}\mathbf{z}_J = \mathbf{0}. \quad (\text{A.14})$$

We claim that $\mathbf{z}_J = \mathbf{0}$ and $\mathbf{z}_{\bar{J}} = \mathbf{0}$. Indeed, by (A.12) we get that $\det(\mathbf{I}_{JJ} - \mathbf{D}_{JJ} + \mathbf{T}_{JJ}\mathbf{D}_{JJ}) \geq \det \mathbf{T}_{JJ}\mathbf{D}_{JJ} > 0$ (see, e.g. Horn & Johnson, 1991, Problem 18 in Chapter 2.5), which leads to $\mathbf{z}_J = \mathbf{0}$ in (A.13) and in turn from (A.14) $\mathbf{z}_{\bar{J}} = \mathbf{0}$. Therefore we conclude $(\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{D})$ is non-singular. \square

A.7 Proof of Lemma 5

The goal of this appendix section is to prove Lemma 5.

Proof. “ \Rightarrow ”: Let \mathbf{y}^* be the unique solution of $\text{LCP}(\mathbf{b}, \mathbf{T})$. Suppose that \mathbf{x}^* and $\tilde{\mathbf{x}}^*$ both solve $\text{PLS}(\mathbf{b}, \mathbf{T})$. Then by the part (b) of Corollary 1 and (1) we get

$$\begin{aligned}\max(\mathbf{x}^*, \mathbf{0}) &= \max(\tilde{\mathbf{x}}^*, \mathbf{0}) = \mathbf{y}^*, \\ \min(\mathbf{x}^*, \mathbf{0}) &= \min(\tilde{\mathbf{x}}^*, \mathbf{0}) = -\mathbf{T}\mathbf{y}^* + \mathbf{b},\end{aligned}$$

which indicates that $\mathbf{x}^* = \tilde{\mathbf{x}}^*$.

“ \Leftarrow ”: Let \mathbf{x}^* be the unique solution of $\text{PLS}(\mathbf{b}, \mathbf{T})$. Suppose that \mathbf{y}^* and $\tilde{\mathbf{y}}^*$ both solve $\text{LCP}(\mathbf{b}, \mathbf{T})$. Then by the part (a) of Corollary 1 we get

$$\mathbf{y}^* - \mathbf{T}\mathbf{y}^* + \mathbf{b} = \tilde{\mathbf{y}}^* - \mathbf{T}\tilde{\mathbf{y}}^* + \mathbf{b} = \mathbf{x}^*.$$

By similar argument as in the proof of part (a) of Corollary 1 we have $\mathbf{y}^* = \tilde{\mathbf{y}}^* = \max(\mathbf{x}^*, \mathbf{0})$.

□

References

- Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.
- Brugnano, L. and Casulli, V. Iterative solution of piecewise linear systems. *SIAM Journal on Scientific Computing*, 30:463–472, 2008.
- Brugnano, L. and Casulli, V. Iterative solution of piecewise linear systems and applications to flows in porous media. *SIAM Journal on Scientific Computing*, 31: 1858–1873, 2009.
- Brugnano, L. and Sestini, A. Iterative solution of piecewise linear systems for the numerical solution of obstacle problems. 2009a. URL <http://arxiv.org/abs/0809.1260>.
- Brugnano, L. and Sestini, A. A new approach based on piecewise linear systems for the numerical solution of obstacle problems. In *Proceedings of AIP Conference*, volume 1168, pp. 746–749, 2009b.
- Burges, C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

- Casulli, V. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *J. Comput. Phys.*, 86:56–74, 1990.
- Chang, C.-C. and Lin, C.-J. Libsvm: a library for support vector machines. 2001.
- Chappelle, O. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.
- Chen, Jinhai and Agarwal, Ravi P. On Newton-type approach for piecewise linear systems. *Linear Algebra and its Applications*, 433:1463–1471, 2010.
- Coleman, T.F. and Li, Y. A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variable. *SIAM Journal on Optimization*, 6(4): 1040–1058, 1996.
- Combettes, P.L. and Pesquet, J.-C. A douglascrachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, 4(1):564–574, 2007.
- Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20, 1995.
- Cottle, R.W., Pang, J.-S., and Stone, R.R. *The Linear Complementarity Problem*. Academic Press, 1992.
- Donoho, D. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41, 1995.
- Duff, I., Grimes, R., and Lewis, J. Sparse matrix test problems. *ACM Transactions on Mathematical Software*, 15:1–14, 1989.
- Eaves, B.C. The linear complementarity problem. *Management Science*, 17:612–634, 1971.
- Fischer, Andreas. A Newton-type method for positive-semidefinite linear complementarity problems. *Journal of Optimization Theory and Applications*, 86(3):585–608, 1995.

- Fischer, Andreas and Kanzow, Christian. On finite termination of an iterative method for linear complementarity problems. *Mathematical Programming: Series A and B*, 74:279–292, 1996.
- Harker, P. T. and Pang, J.-S. A damped Newton method for the linear complementarity problem. In Allgower, E. L. and Georg, K. (eds.), *Computational Solution of Nonlinear Systems of Equations (Lectures on Applied Mathematics 26, AMS)*. 1990.
- Harker, P.T. and Xiao, B. Newton’s method for the nonlinear complementarity problem: a b-differentiable equation approach. *Mathematical Programming*, 48:339–357, 1990.
- Horn, R.A. and Johnson, C.R. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- Ito, Kazufumi and Kunisch, Karl. On a semi-smooth Newton method and its globalization. *Mathematical Programming*, 118:347–370, 2009.
- Jacob, Laurent, Obozinski, Guillaume, and Vert, Jean-Philippe. Group lasso with overlap and graph lasso. In *ICML*, 2009.
- Kimeldorf, George S. and Wahba, Grace. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41:495–502, 1970.
- Kowalski, M. and Torreesani, B. Sparsity and persistence: mixed norms provide simple signals models with dependent coefficient. *Signal, Image and Video Processing*, doi:10.1007/s11760-008-0076-1, 2008.
- Kummer, B. Newton’s method for non-differentiable functions. In et al., J. Guddat (ed.), *Mathematical Research, Advances in Mathematical Optimization*. Akademie-Verlag, Berlin, Germany, 1988.
- Morini, Benedetta and Porcelli, Margherita. Tresnei, a matlab trust-region solver for systems of nonlinear equalities and inequalities. *Computational Optimization and Applications*, DOI: 10.1007/s10589-010-9327-5., 2010.

- Pang, J.-S. Newton's method for b-differentiable equations. *Mathematics of Operations Research*, 15:311–341, 1990.
- Potra, F. A. and Liu, X. Corrector-predictor methods for sufficient linear complementarity problems in a wide neighborhood of the central path. *SIAM Journal on Optimization*, 17:871–890, 2006.
- Qi, L. Convergence analysis of some algorithms for solving nonsmooth equations. *Mathematics of Operations Research*, 18:227–244, 1993.
- Schmidt, Mark, van den Berg, Ewout, Friedlander, Michael P., and Murph, Kevin. Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In *International Conference on Artificial Intelligence and Statistics*, 2009.
- Stelling, G.S. and Duynmeyer, S.P.A. A staggered conservative scheme for every froude number in rapidly varied shallow water flows. *Int. J. Numer. Methods Fluids*, 43: 1329–1354, 2003.
- Tseng, P. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal of Optimization*, 2008.
- Wright, S. J. *Primal-Dual Interior Point Methods*. SIAM, 1997.
- Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.
- Yuan, X. and Yan, S. A finite newton algorithm for non-degenerate piecewise linear systems. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, 2011.
- Zhou, Yang, Jin, Rong, and Hoi, Steven C.H. Exclusive lasso for multi-task feature selection. In *International Conference on Artificial Intelligence and Statistics*, 2010.