A Finite Newton Algorithm for Non-degenerate Piecewise Linear Systems

Xiao-Tong Yuan **Shuicheng Yan**
Electrical and Computer Engineering Department
National University of Singapore
{eleyuanx, eleyans}@nus.edu.sg

Abstract

We investigate Newton-type optimization methods for solving piecewise linear systems (PLS) 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 solution of non-degenerate PLS. 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 to modeling, from a novel perspective of PLS, several statistical learning problems such as elitist Lasso, non-negative least squares and support vector machines. 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 (2009) introduced and investigated the *piecewise linear systems* which involve non-smooth functions of the solution itself

$$\min\{\mathbf{0}, \mathbf{x}\} + \mathbf{T}\max\{\mathbf{0}, \mathbf{x}\} = \mathbf{b}, \tag{1}$$

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 known coefficient matrix, $\mathbf{b} \in \mathbb{R}^d$ is a known vector, and

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

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

The systems (1), abbreviated by PLS($\mathbf{b}$,$\mathbf{T}$), arises from the semi-implicit methods for the numerical simulation of free-surface hydrodynamics (Stelling & Duynmeyer, 2003) and the numerical solutions to obstacle problems (Brugnano & Sestini, 2009; Brugnano & Casulli, 2008). For these problems, the coefficient matrix $\mathbf{T}$ in PLS is typically an M-matrix or inverse-positive matrix, in condition of which several finite Newton methods have been proposed (Brugnano & Sestini, 2009; Chen & Agarwal, 2010).

In this paper, we are in particular concern with Newton-type methods for solving a wide class of PLS($\mathbf{b}$,$\mathbf{T}$) where $\mathbf{T}$ is *non-degenerate*, i.e., every principal minor is non-zero. Such systems arise from several concrete machine learning problems which we shall address in Section 4.

1.1 A Motivating Example Problem: Elitist Lasso

One important motivation, for solving non-degenerate PLS, stands in the efficient optimization of the *elitist Lasso* problem (Kowalski & Torreesani, 2008). A detailed description of elitist Lasso is given in Section 4.1. Here, let us consider the problem in *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 shown by Proposition 2 stated in Section 4.1, the optimal solution $\mathbf{w}^\star$ is given by

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

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

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

Clearly, for $\lambda > 0$, the matrix $\mathbf{T} = \lambda\mathbf{Q} + \mathbf{I}$ is positive-definite, i.e., non-degenerate, but not necessarily an M-matrix or inverse-positive.

From this example we can see that an insight study on efficient numerical solutions for non-degenerate PLS($\mathbf{b}$,$\mathbf{T}$) is of interests in machine learning.

1.2 Existing Newton-type Algorithms for PLS

For obstacle problems, Brugnano & Sestini (2009) proposed a monotone and finite Newton method to solve PLS($\mathbf{b}$,$\mathbf{T}$) with $\mathbf{T}$ satisfying the following assumption

(A1) $\mathbf{T}$ is a symmetric M-matrix (i.e., it can be written as $\mathbf{T} = \alpha\mathbf{I} - \mathbf{B}$ with $\mathbf{B} \geq \mathbf{O}$ and $\|\mathbf{B}\|_2 < \alpha$)

It has been shown in (Brugnano & Sestini, 2009, Corollary 9) that the said method converges within d steps of iterations. Some variants and extensions of this method were proposed in (Brugnano & Casulli, 2008, 2009) under slightly different formulations. More recently, Chen & Agarwal (2010) proposed a similar finite Newton method under a weaker assumption

(A2) $\mathbf{T}$ is an inverse-positive matrix, i.e., $\mathbf{T}^{-1} \geq \mathbf{O}$.

Despite the remarkable success, it is still unclear about the performance of Newton-type method when applied to solve non-degenerate PLS which is obviously beyond those covered by conditions (A1) or (A2).

1.3 Our Contribution

The major contribution of this paper is the PLS-DN algorithm along with its analysis to solve the PLS($\mathbf{b}$,$\mathbf{T}$) with non-degenerate matrix $\mathbf{T}$. PLS-DN is a semi-smooth damped Newton method with global convergence guarantee. 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 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 to learning problems including elitist Lasso (eLasso), non-negative least squares (NNLS), and support vector machines (SVMs). For the problem of eLasso, we are interested in the general case with group overlaps. To the best of our knowledge, this has not yet been explicitly addressed in literature. We propose a proximal optimization method in which the proximity operator is characterized by solving a PLS with positive-definite coefficient matrix. For NNLS with over-determined design matrix, we reformulate the problem as a PLS with positive-definite coefficient matrix. Numerical results on benchmarks show that PLS-DN outperforms several representative Newton-type NNLS solvers. For SVMs, we show that the non-linear SVMs in primal form can be numerically modeled as a PLS with positive-definite coefficient matrix. The PLS-DN solver in this setting is closely related to the Newton-type algorithm proposed by Chappelle (2007). Our analysis provides finite termination guarantee for Chappelle's method.

1.4 Notation

Before continuing, we pause to establish notations formally. 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 ith 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 $\|\mathbf{T}\|_2$, we denote the spectral norm, i.e., the largest singular value of matric $\mathbf{T}$. Throughout this paper, the index set $\{1, ..., d\}$ 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 $J \neq \emptyset$. The all-zero matrix and vector are denoted as $\mathbf{O}$ and $\mathbf{0}$, respectively.

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

2 Mathematical Background

We establish in Section 2.1 a primal-dual connection between PLS and the well known *linear complementary problem* (LCP) for which several off-the-shelf solvers are available. Such a connection also leads to the results on uniqueness of non-degenerate PLS solution in Section 3.3. Some mathematical preliminaries are introduced in Section 2.2.

2.1 Links to LCP Problem: A Primal-Dual View

Actually, the efficient solution of PLS given by (1) is of interest in numerical optimization because it is closely linked to the well known *linear complementary problem* (LCP) (see, e.g., Cottle et al., 1992), which is defined as the solution to the following systems on vector $\mathbf{y} \in \mathbb{R}^d$:

$$\mathbf{y} \geq \mathbf{0}, \quad \mathbf{T}\mathbf{y} - \mathbf{b} \geq \mathbf{0}, \quad \mathbf{y}'(\mathbf{T}\mathbf{y} - \mathbf{b}) = 0. \qquad (2)$$

where matrix $\mathbf{T}$ and vector $\mathbf{b}$ are known. We refer the above form as LCP($\mathbf{b}$, $\mathbf{T}$) in short. The folllowing result shows that if we regard PLS$(\mathbf{b}, \mathbf{T})$ as a primal problem, then LCP($\mathbf{b}$, $\mathbf{T}$) can be viewed as its dual problem.

Lemma 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 LCP($\mathbf{b}$, $\mathbf{T}$) in (2), then $\mathbf{x} = \mathbf{y} - \mathbf{T}\mathbf{y} + \mathbf{b}$ is a solution of PLS($\mathbf{b}$,$\mathbf{T}$) in (1).

(b) If $\mathbf{x}$ is a solution of PLS($\mathbf{b}$,$\mathbf{T}$) in (1), then $\mathbf{y} = \max(\mathbf{0}, \mathbf{x})$ is a solution of LCP($\mathbf{b}$, $\mathbf{T}$) in (2).

The proof is given in Appendix A.1. Since PLS($\mathbf{b}$,$\mathbf{T}$) can be cast to an LCP($\mathbf{b}$, $\mathbf{T}$), one may alternatively solve PLS 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 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 restricted to be a non-degenerate matrix. It has been shown in (Fischer & Kanzow, 1996) that Harker and Pang's algorithm terminates in finite iterations under standard assumptions.

Although PLS($\mathbf{b}, \mathbf{T}$) can be solved in dual with some off-the-shelf LCP solvers, directly addressing PLS($\mathbf{b}, \mathbf{T}$) in primal using Newton method is of algorithmic interests and still remains open for non-degenerate cases. Moreover, our proposed method enriches the bank of LCP solvers.

2.2 Mathematical Preliminary

We assume that $\mathbf{T}$ 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 lemma holds:

Lemma 2. *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.*

Since $\min\{\mathbf{0}, \mathbf{x}\} = \mathbf{x} - \max\{\mathbf{0}, \mathbf{x}\}$, systems (1) is equivalent to the following equation systems:

$$\mathbf{x} + (\mathbf{T} - \mathbf{I}) \max\{\mathbf{0}, \mathbf{x}\} = \mathbf{b}. \tag{3}$$

In this paper we aim to resort to Pang's damped Newton method (Pang, 1990) for solving (3). Let us define function $F : \mathbb{R}^d \mapsto \mathbb{R}^d$ as follows:

$$F(\mathbf{x}) := \mathbf{x} + (\mathbf{T} - \mathbf{I}) \max\{\mathbf{0}, \mathbf{x}\} - \mathbf{b}. \tag{4}$$

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 (B for Bouligand) at point $\mathbf{x}^k$ on direction $\Delta\mathbf{x}$ (see, e.g. Pang, 1990; Harker & Xiao, 1990, for details):

$$BF(\mathbf{x}^k; \Delta\mathbf{x}) = \Delta\mathbf{x} + (\mathbf{T} - \mathbf{I})\mathbf{q}, \tag{5}$$

where vector $\mathbf{q} = (q_i)$ is given by

$$q_i = \begin{cases} \Delta x_i & \text{if } i \in \alpha(\mathbf{x}^k) := \{i \in \mathcal{I} | x_i^k > 0\} \\ \max\{\Delta x_i, 0\} & \text{if } i \in \beta(\mathbf{x}^k) := \{i \in \mathcal{I} | x_i^k = 0\} \\ 0 & \text{if } i \in \gamma(\mathbf{x}^k) := \{i \in \mathcal{I} | x_i^k < 0\} \end{cases}.$$

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

3 PLS-DN: A Damped Newton PLS Solver

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

$$g(\mathbf{x}) = \frac{1}{2}\|F(\mathbf{x})\|^2 \tag{6}$$

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 (7) has a solution for all $\mathbf{x}^k$. Under rather mild assumptions, classical analysis (Pang, 1990; Qi, 1993) shows that Algorithm 1 converges globally to the accumulation point $\mathbf{x}^\star$ with $g(\mathbf{x}^\star) = 0$, i.e., $F(\mathbf{x}^\star) = \mathbf{0}$. 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 $\mathbf{T}$ and a vector $\mathbf{b}$.
Output: Vector $\mathbf{x}^k$.
Initialization: Choose $\mathbf{x}^0$, $\theta, \sigma \in (0, 1)$ and set $k := 0$.
repeat

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

$$BF(\mathbf{x}^k; \Delta\mathbf{x}) = -F(\mathbf{x}^k). \tag{7}$$

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

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

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

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 (7) is highly non-trivial due to the nonlinearity of vector $\mathbf{q}$ on set $\beta(\mathbf{x}^k)$. Following the terminology in (Harker & Pang, 1990), we call the index set $\beta(\mathbf{x}^k)$ the *degenerate set* and the indices in $\beta(\mathbf{x}^k)$ the *degenerate indices*. If $\beta(\mathbf{x}^k)$ is

empty, then $\mathbf{x}^k$ is called a *non-degenerate* vector[1]. It is interesting to note that for non-degenerate $\mathbf{x}^k$, the vector $\mathbf{q}$ is a linear form respect to $\Delta\mathbf{x}$. To see this, let us define the following diagonal matrix:

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

where $p(x_i) = 1$ if $x_i \geq 0$, and 0 otherwise. Obviously $\mathbf{P}(\mathbf{x})\mathbf{x} = \max\{\mathbf{0}, \mathbf{x}\}$. Thus $F(\mathbf{x})$ can be written as:

$$F(\mathbf{x}) = (\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}(\mathbf{x}))\,\mathbf{x} - \mathbf{b}. \tag{8}$$

Let $\mathbf{P}^k := \mathbf{P}(\mathbf{x}^k)$. By trivial check, the following result immediately holds.

Lemma 3. *If $\mathbf{x}^k$ is non-degenerate, then $\mathbf{q}$ in (5) can be expressed as the following linear form*

$$\mathbf{q} = \mathbf{P}^k \Delta\mathbf{x}. \tag{9}$$

Proposition 1. *If $\mathbf{x}^k$ is non-degenerate, and $\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}^k$ is non-singular, then the solution of generalized Newton equation (7) is give by*

$$\Delta\mathbf{x} = -\mathbf{x}^k + \left(\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}^k\right)^{-1}\mathbf{b}. \tag{10}$$

The proof is given in Appendix A.2. Proposition 1 motivates us to modify Algorithm 1 so that the generated $\{\mathbf{x}^k\}_{k\geq 0}$ remain non-degenerate, and thus the generalized Newton equation (7) has analytical solution (10). The modified damped Newton method is formally given in Algorithm 2. The key difference between the two algorithms is that: 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 in turn simplifies the solution of the generalized Newton equation in (S.1). As a result, we have the following theorem on global convergence of Algorithm 2.

Theorem 1 (Global Convergence). *Let $\{\mathbf{x}^k\}$ be any sequence generated by Algorithm 2. Assume that $F(\mathbf{x}^k) \neq \mathbf{0}$ for all k. Then*

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

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

The proof is given in Appendix A.3. On convergence rate, we establish in the following theorem the linear rate of convergence for Algorithm 2. The proof is given in Appendix A.4.

[1]The concept of non-degenerate vector defined here can be regarded as a vector counterpart of non-degenerate matrix

Theorem 2 (Linear Convergence Rate). *Let $\{\mathbf{x}^k\}$ be any sequence generated by Algorithm 2. Assume that $F(\mathbf{x}^k) \neq \mathbf{0}$ for all k. Suppose that $\mathbf{x}^\star$ is an accumulation point of $\{\mathbf{x}^k\}$ and $\mathbf{x}^\star$ is a zero of F. If matrix $\mathbf{T}$ is non-degenerate, then the entire sequence $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^\star$ 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, we currently can only prove the (at least) linear rate of convergence for Algorithm 2. In practice, however, we observe that the perturbation seldom occurs in Algorithm 2 since the vectors $\{\mathbf{x}^k\}$ 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.*

Algorithm 2: The modified PLS-DN method.

Input : A non-degenerate matrix $\mathbf{T}$ and a vector $\mathbf{b}$.
Output: Vector $\mathbf{x}^k$.
Initialization: Choose a non-degenerate $\mathbf{x}^0$, $\theta, \sigma \in (0,1)$, and set $k := 0$.
repeat

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

$$\Delta\mathbf{x}^k := -\mathbf{x}^k + \left(\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}^k\right)^{-1}\mathbf{b}. \tag{11}$$

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

$$\left\|F(\mathbf{x}^k + \theta^m\Delta\mathbf{x}^k)\right\|^2 \leq (1 - \theta^m\sigma)\left\|F(\mathbf{x}^k)\right\|^2. \tag{12}$$

(S.3) Set $\tilde{\mathbf{x}}^{k+1} := \mathbf{x}^k + t_k\Delta\mathbf{x}^k$, $\mathbf{x}^{k+1} := \tilde{\mathbf{x}}^{k+1}$.
if $\left\|F(\tilde{\mathbf{x}}^{k+1})\right\| \neq 0$ **then**
Set $\mathbf{x}_i^{k+1} := \tilde{\mathbf{x}}_i^{k+1} + \delta_{k+1}, \forall i \in \beta(\tilde{\mathbf{x}}^{k+1})$, where $0 < \delta_{k+1} \leq \frac{(1-\sqrt{1-t_k\sigma})\|F(\mathbf{x}^k)\|}{2L\sqrt{d}}$.
end
$k := k + 1$
until $\|F(\mathbf{x}^k)\| = 0$;

3.2 Finite Termination

We further show in this subsection 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 $\mathbf{x}^\star$. In the following description, we denote $B_\epsilon(\mathbf{y}) := \{\mathbf{z} \in \mathbb{R}^d \mid \|\mathbf{z} - \mathbf{y}\| \leq \epsilon\}$ an Euclidean ball.

Lemma 4. *Let $\mathbf{x}^\star$ denote a solution of the PLS($\mathbf{b}$,$\mathbf{T}$). Then there exists a positive number $\epsilon(\mathbf{x}^\star)$ such that*

$$(\mathbf{P}(\mathbf{x}) - \mathbf{P}(\mathbf{x}^\star))\,\mathbf{x}^\star = \mathbf{0} \tag{13}$$

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

The proof is given in Appendix A.5.

Theorem 3. *Let $\mathbf{x}^\star \in \mathbb{R}^d$ denote a solution of the PLS($\mathbf{b}$,$\mathbf{T}$). If $\mathbf{I}-\mathbf{P}^k+\mathbf{T}\mathbf{P}^k$ is non-singular, and $\mathbf{x}^k \in B_\epsilon(\mathbf{x}^\star)$ for some sufficiently small $\epsilon > 0$, then $\mathbf{x}^{k+1}$ generated by Algorithm 2 solves the PLS($\mathbf{b}$,$\mathbf{T}$).*

Proof. Let $\epsilon := \epsilon(\mathbf{x}^\star)$ be defined as in Lemma 4. Let $\mathbf{P}^\star := \mathbf{P}(\mathbf{x}^\star)$. By Lemma 4 we have that

$$(\mathbf{I}-\mathbf{P}^\star+\mathbf{T}\mathbf{P}^\star)\mathbf{x}^* = (\mathbf{I}-\mathbf{P}^k+\mathbf{T}\mathbf{P}^k)\mathbf{x}^* = \mathbf{b}. \quad (14)$$

In (S.3) of Algorithm 2, consider $\tilde{\mathbf{x}}^{k+1} := \mathbf{x}^k + \Delta\mathbf{x}^k$. Since $\mathbf{I}-\mathbf{P}^k+\mathbf{T}\mathbf{P}^k$ is non-singular, by (11) and (14), we get

$$\tilde{\mathbf{x}}^{k+1} = \left(\mathbf{I}-\mathbf{P}^k+\mathbf{T}\mathbf{P}^k\right)^{-1}\mathbf{b} = \mathbf{x}^\star.$$

Therefore we have

$$\left\|F(\tilde{\mathbf{x}}^{k+1})\right\|^2 = \|F(\mathbf{x}^\star)\|^2 = 0 \le (1-\sigma)\left\|F(\mathbf{x}^k)\right\|^2,$$

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

Theorem 3 tells us in theory that Algorithm 2 terminates after finite counts of iteration. On such a finite termination behavior, the following two questions naturally arise:

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

Q2: Under what conditions can we guarantee that $\mathbf{I}-\mathbf{P}^k+\mathbf{T}\mathbf{P}^k$ is non-singular as required in Theorem 3?

The following Theorem 4 and Theorem 5 give answers to these two questions respectively.

Theorem 4 (Termination Criteria). *Let $\hat{\mathbf{x}}^{k+1} := \mathbf{x}^k + \Delta\mathbf{x}^k$. If, for some $k \ge 0$, one gets*

$$\left(\mathbf{P}(\hat{\mathbf{x}}^{k+1}) - \mathbf{P}^k\right)\hat{\mathbf{x}}^{k+1} = \mathbf{0},$$

then $\mathbf{x}^\star = \hat{\mathbf{x}}^{k+1}$ is an exact solution of PLS($\mathbf{b}$,$\mathbf{T}$).

Proof. If $\left(\mathbf{P}(\hat{\mathbf{x}}^{k+1}) - \mathbf{P}^k\right)\hat{\mathbf{x}}^{k+1} = \mathbf{0}$, then combining this with (11) yields

$$\begin{aligned}\mathbf{b} &= \left(\mathbf{I}+(\mathbf{T}-\mathbf{I})\mathbf{P}^k\right)\hat{\mathbf{x}}^{k+1} \\ &= \left(\mathbf{I}+(\mathbf{T}-\mathbf{I})\mathbf{P}(\hat{\mathbf{x}}^{k+1})\right)\hat{\mathbf{x}}^{k+1}. \quad (15)\end{aligned}$$

By (15) and (8) we get $F(\hat{\mathbf{x}}^{k+1}) = \mathbf{0}$ which terminates Algorithm 2 with output $\hat{\mathbf{x}}^{k+1}$ that exactly solves (1). □

As a simple consequence, if $\mathbf{P}(\hat{\mathbf{x}}^{k+1}) = \mathbf{P}^k$ is satisfied for some k, then the Algorithm 2 terminates with exact solution $\mathbf{x}^\star = \hat{\mathbf{x}}^{k+1}$.

Theorem 5 (Non-singularity). *If matrix $\mathbf{T} \in \mathbb{R}^{d\times d}$ is non-degenerate, then $\mathbf{I}-\mathbf{P}^k+\mathbf{T}\mathbf{P}^k$ is non-singular.*

Proof. The result obviously holds for $\mathbf{P}^k = \mathbf{0}$. If $\mathbf{P}^k \neq \mathbf{0}$, then we define the index sets

$$J := \{i \in \mathcal{I} : x_i^k \ge 0\} \text{ and } \bar{J} := \{i \in \mathcal{I} : x_i^k < 0\}. \quad (16)$$

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

$$\begin{aligned}\mathbf{T}_{JJ}\mathbf{z}_J &= \mathbf{0}, \quad (17)\\ \mathbf{z}_{\bar{J}} + \mathbf{T}_{\bar{J}J}\mathbf{z}_J &= \mathbf{0}. \quad (18)\end{aligned}$$

By Lemma 2 we have that $\mathbf{T}_{JJ}$ is non-singular, and thus $\mathbf{z}_J = \mathbf{0}$. Combining this with (18) yields $\mathbf{z}_{\bar{J}} = \mathbf{0}$. Consequently, we get that $(\mathbf{I}-\mathbf{P}^k+\mathbf{T}\mathbf{P}^k)$ is non-singular. □

Theorem 5 along with its proof actually motivates us an efficient implementation of step (S.1) in Algorithm 2, which requires solving a linear systems

$$\left(\mathbf{I}-\mathbf{P}^k+\mathbf{T}\mathbf{P}^k\right)\mathbf{z} = \mathbf{b}. \quad (19)$$

A direct solution of the preceding systems leads to $O(d^3)$ complexity[2]. However, by similar argument in the proof of Theorem 5, systems (19) can be decomposed as

$$\begin{aligned}\mathbf{T}_{JJ}\mathbf{z}_J &= \mathbf{b}_J, \quad (20)\\ \mathbf{z}_{\bar{J}} + \mathbf{T}_{\bar{J}J}\mathbf{z}_J &= \mathbf{b}_{\bar{J}}, \quad (21)\end{aligned}$$

where J and $\bar{J}$ are given by (16). With such a decomposition, to obtain the solution $\mathbf{z} = (\mathbf{z}_J, \mathbf{z}_{\bar{J}})$, we only need to solve the smaller linear systems (20) with complexity $O(|J|^3)$ to obtain $\mathbf{z}_J$, and to solve the equation (21) with complexity $O(|J||\bar{J}|)$ to obtain $\mathbf{z}_{\bar{J}}$. Of course, in worst case, i.e., $|J| = d$, the complexity is still $O(d^3)$. 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 (19).

3.3 Existence and Uniqueness of the Solution

We study in this section the existence and uniqueness of PLS-DN solution. Concerning the existence of a solution, the thesis follows directly from Algorithm 2 and Theorem 1. Concerning the uniqueness, one natural question is whether the solution is unique for all non-degenerate matrix $\mathbf{T}$? The answer is *negative*. To see this, we construct a counter example as follows:

A Counter Example: Let $\mathbf{T} = \text{diag}(-1, 1, ..., 1)$ and $\mathbf{b} = (-1, 1, ..., 1)'$, it is straightforward to check that both

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

$\mathbf{x}_1^\star = (1, 1, .., 1)'$ and $\mathbf{x}_2^\star = (-1, 1, .., 1)'$ are the solutions of PLS($\mathbf{b}, \mathbf{T}$).

To further derive the conditions for uniqueness, we make use of the primal-dual connection between PLS($\mathbf{b}$,$\mathbf{T}$) 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 $\mathbf{b} \in \mathbb{R}^d$, PLS($\mathbf{b}$,$\mathbf{T}$) has a unique solution if and only if LCP($\mathbf{b}$, $\mathbf{T}$) has a unique solution.*

The proof is given in Appendix A.6. The preceding lemma motivates us to discuss the uniqueness of PLS solution from the viewpoint of its dual problem, LCP. The uniqueness of LCP solution is related to the concept of P-matrix defined by:

Definition 2. *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 $J \subseteq \mathcal{I}$.*

Obviously, a P-matrix is non-degenerate. 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. The M-matrix is also a subset of P-matrix.

The following standard result gives a *sufficient and necessary* condition to guarantee 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 now ready to present the following main result on the uniqueness of PLS solution.

Theorem 6 (**Uniqueness of Solution**)**.** *PLS($\mathbf{b}$,$\mathbf{T}$) has a unique solution for all vectors $\mathbf{b} \in \mathbb{R}^d$ if and only if matrix $\mathbf{T}$ is a P-matrix.*

In the following application studies in Section 4, the matrices $\mathbf{T}$ in PLS are all positive-definite matrices, and thus are P-matrices. Therefore, the output solution of our PLS-DN algorithm is always unique from any initial point $\mathbf{x}^0$.

4 Applications to Learning Problems

In this section, we show several applications of non-degenerate PLS($\mathbf{b}$,$\mathbf{T}$) in learning problems. We numerically model the following problems as PLS and apply the PLS-DN method for optimization: elitist Lasso with *group overlaps* (Section 4.1), non-negative least squares (Section 4.2) and primal non-linear SVMs (see Appendix B). In the following description, $\mathcal{D} = \{(\mathbf{u}_i, v_i)\}_{1\leq i\leq n}$ is a set of observed data, $\mathbf{u}_i \in \mathbb{R}^d$ is the feature vector, and v_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, and the 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 $\sigma = 0.01$.

4.1 App-I: Elitist Lasso with Group Overlaps

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

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^n L(v_i, \mathbf{w}'\mathbf{u}_i) + \frac{\lambda}{2}\sum_{g\in\mathcal{G}} \|\mathbf{w}_g\|_1^2, \qquad (22)$$

where $L(\cdot,\cdot)$ is a smooth convex loss function. As shown in (Kowalski & Torreesani, 2008; Zhou et al., 2010) that such an $\ell_{1,2}$-regularized minimization will encourage the 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 consider the general model with group overlaps which is useful for exclusive feature selection where features may belong to different groups.

Since convex objective in (22) is the sum 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

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{z}\|^2 + \frac{\lambda}{2}\sum_{g\in\mathcal{G}} \|\mathbf{w}_g\|_1^2. \qquad (23)$$

Equivalently, we may reformulate the problem (23) 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 non-degenerate PLS.

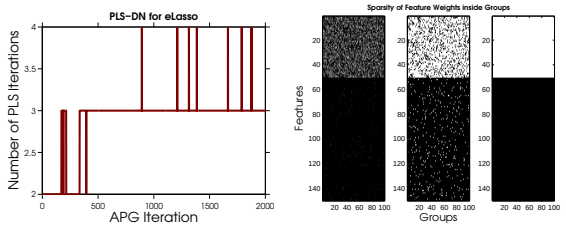
Proposition 2. *The optimizer $\mathbf{w}^\star$ of proximity operator* (23) *is given by*

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

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

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

The proof is given in Appendix A.7. For any $\lambda > 0$, the coefficient matrix $\mathbf{T} = \lambda\mathbf{Q} + \mathbf{I}$ is positive-definite, i.e., non-degenerate. We can apply the modified PLS-DN in Algorithm 2 to solve the proximity operator (23) in finite iterations. By incorporating such an operator into an accelerated



(a) (b)

Figure 1: 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 $\mathbf{w}^\star$. Middle: the sparsity pattern of $\mathbf{w}^\star$. Right: the sparsity pattern of the ground truth $\mathbf{w}$.

proximal gradient (APG) algorithm (Tseng, 2008), we can efficiently solve the eLasso problem with group overlaps.

It is worthy to note 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 its efficiency is insensitive to the extent of overlap.

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., $L(v_i, \mathbf{w}'\mathbf{u}_i) := \frac{1}{2}\|v_i - \mathbf{w}'\mathbf{u}_i\|^2$. 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 1(a) shows the number of PLS-DN iterations at each step during the APG optimization. It can be observed that PLS-DN terminates within 4 iterations. The sparsity of the recovered feature weights are shown in Figure 1(b). From these results we can see that PLS-DN is efficient to optimize eLasso with overlaps.

4.2 App-II: Non-negative Least Squares

Many applications, e.g. non-negative image restoration, contact problems for mechanical systems, control problems, involve the numerical solution of non-negative least squares (NNLS) problems

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{n}(v_i - \mathbf{w}'\mathbf{u}_i)^2, \text{ subject to } \mathbf{w} \geq \mathbf{0}. \qquad (24)$$

We assume that $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_n)$ has full row rank so that the NNLS problem (24) is a strictly convex optimization problem and there exists a unique solution $\mathbf{w}^\star$. Let $a_i \geq 0$ denote the Lagrange multipliers used to enforce the non-negativity constraint on w_i. The set of KKT conditions are given by

$$\mathbf{a} \geq \mathbf{0}, \ \ \mathbf{w} \geq \mathbf{0}, \ \ \mathbf{a}'\mathbf{w} = 0, \ \ \mathbf{a} = \mathbf{U}\mathbf{U}'\mathbf{w} - \mathbf{U}\mathbf{v}.$$

Obviously, this set of conditions form an LCP problem, which due to Lemma 1 is equivalent to the following PLS:

$$\min\{\mathbf{0}, \mathbf{x}\} + \mathbf{U}\mathbf{U}' \max\{\mathbf{0}, \mathbf{x}\} = \mathbf{U}\mathbf{v}.$$

Since $\mathbf{U}$ has full row rank, the coefficient matrix $\mathbf{U}\mathbf{U}'$ is positive-definite. Given $\mathbf{x}^\star$ the solution of the above PLS problem, by Lemma 1 we have that the optimal solution of NNLS is given by $\mathbf{w}^\star = \max\{\mathbf{0}, \mathbf{x}^\star\}$.

Simulation The numerical evaluations of PLS-DN for NNLS 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)[3]. The non-degenerate design matrices $\mathbf{U}$ in these problems are well-conditioned or moderately ill-conditioned. In this test, we uniformly set each element of ground truth $\mathbf{w}$ in $(0, 1)$. The i.i.d. noise in linear model is Gaussian with mean 0 and variance 10^{-4}. The initial point is all-zero vector. We compare our method with the following methods which are capable to solve NNLS:

- Two LCP solvers: A damped Newton solver based on (Fischer, 1995)[4] which we call LCP-Fischer in our test, and a Lemke's pivoting solver based on (Cottle et al., 1992)[5] which we call LCP-Lemke in our test.
- Two Matlab routines: the `lsqlin` which is based on reflective Newton method (Coleman & Li, 1996) and the `lsqnonneg` which is based on active set approach (Lawson & Hanson, 1974).
- The projected Quasi-Newton (PQN) solver (Schmidt et al., 2009)[6] based on LBFGS method.
- The TRESNEI solver (Morini & Porcelli, 2010)[7] based on trust-region Gaussian-Newton method.
- The SCD solver (Shalev-Shwartz & Tewari, 2009) based on stochastic coordinate descent method.

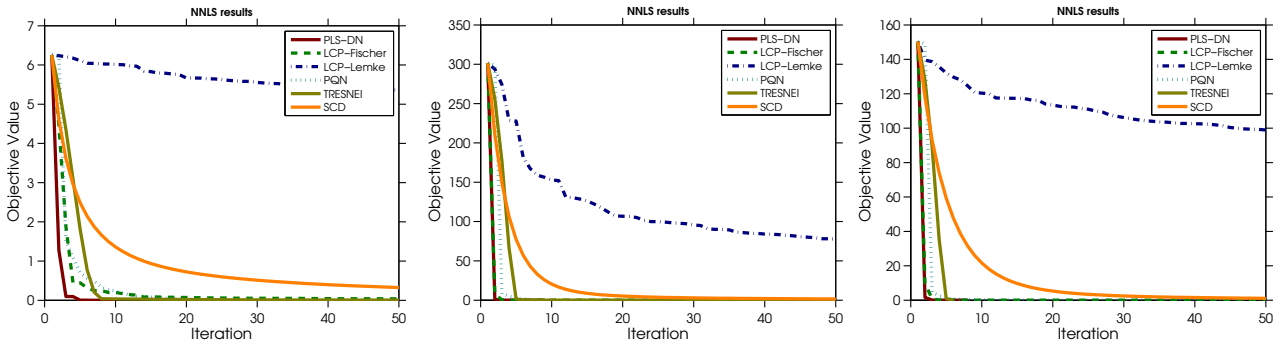
[3] These three problems are publicly available at `http://www.cise.ufl.edu/research/sparse/matrices/`

[4] `http://alice.nc.huji.ac.il/~tassa/pmwiki.php?n=Main.Code`

[5] `http://people.sc.fsu.edu/~jburkardt/m_src/lemke/lemke.html`

[6] `http://www.cs.ubc.ca/~schmidtm/Software/PQN.html`

[7] `http://tresnei.de.unifi.it/`



(a) *add20* (b) *illc1850* (c) *well1850*

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

Table 1: The quantitative results for different NNLS algorithms on Harwell Boeing collection. For all the comparing iterative methods, the initial points are set to be $\mathbf{x}^0 = \mathbf{0}$.

Methods	*add20*			*illc1850*			*well1850*		
	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}
`lsqlin`	15	6.85	2.80×10^{-6}	19	0.61	6.04×10^{-4}	20	0.71	6.61×10^{-6}
`lsqnonneg`	2342	7.03×10^{3}	2.22×10^{-7}	728	138.19	5.66×10^{-4}	723	132.43	5.64×10^{-6}
PQN	77	2.36	1.86×10^{-4}	589	8.89	7.45×10^{-4}	199	4.01	6.91×10^{-5}
TRESNEI	2555	33.20	5.22×10^{-7}	7766	119.60	5.66×10^{-4}	**5**	0.08	5.64×10^{-6}
SCD	100	60.21	0.15	100	2.64	0.67	100	2.61	0.39

Quantitative results by different methods are listed in Table 1, 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 terminates much earlier than the semi-smooth Newton LCP solver LCP-Fishcher to achieve the exact (up to the machine precision to solve linear systems) solution. Figure 2 shows the evolving curves of objective value as functions of iterations for different NNLS algorithms. It can be observed from these curves that PLS-DN and LCP-Fischer converge very quickly in 3-5 iterations, and so are PQN and TRESNEI in 5-10 iterations. Despite similar sharp convergence behaviors, it is shown in Table 1 that in all cases but one PLS-DN terminates much earlier than the other methods. To conclude, PLS-DN is an efficient and exact Newton-type solver for NNLS problem.

5 Conclusion and Future Work

This paper addressed the problem of solving a wide class of PLS with non-degenerate coefficient matrix. The proposed PLS-DN algorithm is a damped Newton method with global linear convergence behavior and finite termination guarantee. We apply PLS to numerically model several concrete statistical learning problems such as elitist Lasso, non-negative least squares and support vector machines. By comparing experiments on several benchmark tasks, we conclude that PLS-DN performs well both in time efficiency and solution accuracy.

It is noteworthy that the PLS($\mathbf{b}, \mathbf{T}$) problem (1) is a special case of the following systems (Brugnano & Casulli, 2009)

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

where $\mathbf{l} = (l_i), \mathbf{u} = (u_i) \in \mathbb{R}^d$ are known vectors and $l_i \leq u_i$. We call the preceding equation systems as PLS($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$). When $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} = \boldsymbol{\infty}$, PLS($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) reduces to (1). When $(\mathbf{T} - \mathbf{I})^{-1}$ is a symmetric M-matrix, Brugnano & Casulli (2009) proposed two finite Newton-type algorithms to solve PLS($\mathbf{b}$,$\mathbf{T}, \mathbf{l}, \mathbf{u}$) along with applications to confined-unconfined flows in porous media. Our ongoing work in this line is to develop a finite damped Newton method to solve PLS($\mathbf{b}, \mathbf{T}, \mathbf{l}, \mathbf{u}$) with non-degenerate coefficient matrix and exploit its potential applications in statistical learning problems.

Acknowledgment

This research was supported by National Research Foundation/ Interactive DigitalMedia Program, under research Grant NRF2008IDMIDM004-029, Singapore.

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, Luigi and Sestini, Alessandra. Iterative solution of piecewise linear systems for the numerical solution of obstacle problems. 2009. URL `http://arxiv.org/abs/0809.1260`.

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.

Cottle, R.W., Pang, J.-S., and Stone, R.R. *The Linear Complementarity Proble*. Academic Press, 1992.

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*. Canbridge 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.

Lawson, C.L. and Hanson, R.J. *Solving Least Squares Problems*. Prentice-Hall, 1974.

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 *Intermational Conference on Artificial Intelligence and Statistics*, 2009.

Shalev-Shwartz, S. and Tewari, A. Stochastic methods for ℓ_1 regularized loss minimization. In *International Conference on Machine Learning*, 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.

Zhou, Y., Jin, R., and Hoi, S.-C.H. Exclusive lasso for multi-task feature selection. In *International Conference on Artificial Intelligence and Statistics*, 2010.

Appendix

A Technical Proofs

A.1 Proof of Lemma 1

The goal of this appendix is to prove Lemma 1.

Proof. Part (a): Let $\mathbf{y}$ be a solution of systems (2). Let $\mathbf{x} := \mathbf{y} - \mathbf{T}\mathbf{y} + \mathbf{b}$. Since $\mathbf{y} \geq \mathbf{0}$, $\mathbf{T}\mathbf{y} - \mathbf{b} \geq \mathbf{0}$ and $\mathbf{y}'(\mathbf{T}\mathbf{y} - \mathbf{b}) = 0$, it is easy to check that $\max\{\mathbf{0}, \mathbf{x}\} = \mathbf{y}$ and $\min\{\mathbf{0}, \mathbf{x}\} = -\mathbf{T}\mathbf{y} + \mathbf{b}$, which implies $\min(\mathbf{0}, \mathbf{x}) + \mathbf{T}\max(\mathbf{0}, \mathbf{x}) = \mathbf{b}$.

Part (b): Let $\mathbf{x}$ be a solution of systems (1). Clearly, $\mathbf{y} := \max(\mathbf{0}, \mathbf{x}) \geq \mathbf{0}$. Since $\mathbf{x}$ solves (1), it follows that $\mathbf{T}\mathbf{y} - \mathbf{b} = -\min(\mathbf{0}, \mathbf{x}) \geq \mathbf{0}$ and

$$\mathbf{y}'(\mathbf{T}\mathbf{y} - \mathbf{b}) = -\max(\mathbf{0}, \mathbf{x})' \min(\mathbf{0}, \mathbf{x}) = 0.$$

Therefore $\mathbf{y}$ solves (2). □

A.2 Proof of Proposition 1

The goal of this appendix is to prove Proposition 1.

Proof. Since $\mathbf{x}^k$ is non-degenerate, the (9) holds. Combining this with B-differential (5) yields

$$BF(\mathbf{x}^k; \Delta\mathbf{x}) = [\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}^k]\Delta\mathbf{x}. \tag{A.1}$$

Therefore the generalized Newton equation (7) reads

$$\left(\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}^k\right)\Delta\mathbf{x} = -\left(\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}^k\right)\mathbf{x}^k + \mathbf{b} \tag{A.2}$$

By assumption that $\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}^k$ is non-singular, we arrive at (10). □

A.3 Proof of Theorem 1

The goal of this appendix is to prove Theorem 1.

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

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

where the second inequality follows the Lipschitiz continuity of F and the last inequality follows (12). By choosing $0 < \delta_{k+1} \leq \frac{(1-\sqrt{1-t_k\sigma})\|F(\mathbf{x}^k)\|}{2L\sqrt{d}}$, we get that

$$\left\|F(\mathbf{x}^{k+1})\right\| \leq \frac{1 + \sqrt{1 - t_k\sigma}}{2}\left\|F(\mathbf{x}^k)\right\| < \left\|F(\mathbf{x}^k)\right\|. \tag{A.3}$$

Part (b): From (a) the sequence $\{\|F(\mathbf{x}^k)\|\}_{k\geq 1}$ is non-negative and strictly decreasing. Thus it converges, and

$$\lim_{k\to\infty}\left(\left\|F(\mathbf{x}^k)\right\| - \left\|F(\mathbf{x}^{k+1})\right\|\right) = 0. \tag{A.4}$$

By (A.3) it follows that

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

If $\liminf t_k$ is positive, then

$$\|F(\mathbf{x}^\star)\| = \lim_{k\to\infty}\left\|F(\mathbf{x}^k)\right\| = 0.$$

□

A.4 Proof of Theorem 2

The goal of this appendix is to prove Theorem 2.

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 essential 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*

$$\partial_B G(\mathbf{x}) := \left\{\lim_{\mathbf{x}_i\in D_G, \mathbf{x}_i\to\mathbf{x}} \nabla G(\mathbf{x}_i)\right\}$$

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 in* (4) *is strongly BD-regular at any point $\mathbf{x}$.*

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

$$\partial_B F(\mathbf{x}) = \{\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}\}, \tag{A.5}$$

where

$$\mathbf{P} \in \partial_B \max\{\mathbf{0}, \mathbf{x}\} = \{\text{diag}(p_1, ..., p_d)\} \tag{A.6}$$

with $p_i, i = 1, ..., d$ are given by:

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

By similar argument in the proof of Theorem 5 we have that $\mathbf{I} + (\mathbf{T} - \mathbf{I})\mathbf{P}$ is always non-singular given that $\mathbf{T}$ is non-degenerate. □

To prove Theorem 2, we need the following lemma which is a direct consequence of Lemma 7 and the Corollary 3.4 in (Qi, 1993) on the function F at $\mathbf{x}^\star$.

Lemma 8. *Suppose that $\mathbf{x}^\star$ 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 $\|\mathbf{x} - \mathbf{x}^\star\| \leq \rho$, if the generalized Newton equation*

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

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

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

We are now in the position to prove Theorem 2.

Proof of Theorem 2. 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 $\|\mathbf{x}^k - \mathbf{x}^\star\| \leq \rho$,

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

Therefore,

$$\|F(\bar{\mathbf{x}}^{k+1})\|^2 \leq (1-\sigma)\|F(\mathbf{x}^k)\|^2. \tag{A.7}$$

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

$$t_k = 1 \quad \text{and} \quad \tilde{\mathbf{x}}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k = \bar{\mathbf{x}}^{k+1}. \tag{A.8}$$

The choice of perturbation δ_{k+1} ensures that

$$\begin{aligned} \delta_{k+1} &\leq \frac{(1 - \sqrt{1 - t_k\sigma})\|F(\mathbf{x}^k)\|}{2L\sqrt{d}} \\ &\leq \frac{(1 - \sqrt{1-\sigma})\|\mathbf{x}^k - \mathbf{x}^\star\|}{2\sqrt{d}}, \end{aligned} \tag{A.9}$$

where the second inequality follows $t_k \leq 1$, $F(\mathbf{x}^\star) = 0$ and the Lipschitz-continuity. Therefore,

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^\star\| &\leq \|\mathbf{x}^{k+1} - \tilde{\mathbf{x}}^{k+1}\| + \|\tilde{\mathbf{x}}^{k+1} - \mathbf{x}^\star\| \\ &\leq \sqrt{d}\delta_{k+1} + \sqrt{1-\sigma}\|\mathbf{x}^k - \mathbf{x}^\star\| \\ &\leq \frac{1 + \sqrt{1-\sigma}}{2}\|\mathbf{x}^k - \mathbf{x}^\star\| \leq \rho. \end{aligned} \tag{A.10}$$

Since $\mathbf{x}^\star$ is a limiting point of $\{\mathbf{x}^k\}$, there is a $k(\rho)$ such that $\|\mathbf{x}^{k(\rho)} - \mathbf{x}^\star\| \leq \rho$. By introduction of above arguments, (A.8) and (A.10) hold for any $k \geq k(\rho)$. Therefore, the entire sequence $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^\star$ and t_k eventually becomes 1. From (A.10) we can see that the convergence rate is linear for any $\sigma \in (0, 1)$.

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

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

which inequality indicates that the objective value sequence $\{\|F(\mathbf{x}^k)\|\}$ converges linearly towards zero. □

A.5 Proof of Lemma 4

The goal of this appendix is to prove Lemma 4.

Proof. If there is at least one index $i \in \mathcal{I}$ with $x_i^\star \neq 0$, then set

$$\epsilon(\mathbf{x}^\star) := \frac{1}{2}\min\{|x_i^\star| : i \in \mathcal{I}, x_i^\star \neq 0\}. \tag{A.11}$$

Otherwise, let $\epsilon(\mathbf{x}^\star)$ be any positive number. Now, let $x \in B_{\epsilon(\mathbf{x}^\star)}$ and

$$\Delta(x_i) := (p(x_i) - p(x_i^\star))\, x_i^\star. \tag{A.12}$$

We distinguish the following two cases

(i) If $x_i^\star = 0$, obviously $\Delta(x_i) = 0$.

(ii) If $x_i^\star \neq 0$, we obtain that $|x_i - x_i^\star| \leq \epsilon(\mathbf{x}^\star) < |x_i^\star|$ which implies that $x_i \neq 0$, and x_i, $x_i^\star$ are of the same sign. Therefore $p(x_i) = p(x_i^\star)$, $\Delta(x_i) = 0$.

Consequently, we have $\Delta(x_i) = 0$ for all $i \in \mathcal{I}$ and all $\mathbf{x} \in B_{\epsilon(\mathbf{x}^\star)}$. □

A.6 Proof of Lemma 5

The goal of this appendix is to prove Lemma 5.

Proof. "⇒": Let $\mathbf{y}^\star$ be the unique solution of LCP($\mathbf{b}$, $\mathbf{T}$). Suppose that $\mathbf{x}^\star$ and $\tilde{\mathbf{x}}^\star$ both solve PLS($\mathbf{b}$,$\mathbf{T}$). Then by the part (b) of Lemma 1 and (1) we get

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

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

"⇐": Let $\mathbf{x}^\star$ be the unique solution of PLS($\mathbf{b}$,$\mathbf{T}$). Suppose that $\mathbf{y}^\star$ and $\tilde{\mathbf{y}}^\star$ both solve LCP($\mathbf{b}$, $\mathbf{T}$). Then by the part (a) of Lemma 1 we get

$$\mathbf{y}^\star - \mathbf{T}\mathbf{y}^\star + \mathbf{b} = \tilde{\mathbf{y}}^\star - \mathbf{T}\tilde{\mathbf{y}}^\star + \mathbf{b} = \mathbf{x}^\star. \tag{A.13}$$

By similar argument as in the proof of part (a) of Lemma 1 we get that $\mathbf{y}^\star = \tilde{\mathbf{y}}^\star = \max(\mathbf{0}, \mathbf{x}^\star)$. □

A.7 Proof of Proposition 2

The goal of this appendix is to prove Proposition 2.

Proof. Since the objective function in (23) is convex, its optimal solution $\mathbf{w}^\star$ is fully characterized by the Karush-Kuhn-Tucher conditions (see, e.g., Boyd & Vandenberghe, 2004)

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

where $\xi_i := \partial|\cdot|_1(w_i^\star) = \text{sign}(w_i^\star)$ if $w_i^\star \neq 0$ and $\partial|\cdot|(0) = [-1, 1]$ is the subdifferential of the absolute function $|\cdot|$ evaluated at $w_i^\star$. By standard result of soft-thresholding method we have that

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

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

$$\mathbf{x} + \lambda\mathbf{Q}\max\{\mathbf{0}, \mathbf{x}\} = |\mathbf{z}|, \tag{A.14}$$

which obviously is a PLS($|\mathbf{z}|, \lambda\mathbf{Q} + \mathbf{I}$) problem. □

B App-III: The Application of PLS-DN to SVMs

As another concrete application, we show that the SVMs can also be numerically modeled as PLS. 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:

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

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 (B.1) to its non-linear form in terms of $\boldsymbol{\beta}$ as

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} L\left(b_i, \sum_{j=1}^{n} \beta_j k(\mathbf{a}_j, \mathbf{a}_i)\right) + \lambda \sum_{i,j=1}^{n} \beta_i\beta_j k(\mathbf{a}_i, \mathbf{a}_j). \tag{B.2}$$

or in a more compact form written as

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} L(b_i, \mathbf{K}'_{i\bullet}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\mathbf{K}\boldsymbol{\beta}. \tag{B.3}$$

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 ith column of $\mathbf{K}$. The problem (B.3) is widely known as Primal SVMs (Prim-SVMs) (Chappelle, 2007).

B.0.1 Solving Prim-SVMs with PLS

The following result connects Prim-SVMs to PLS.

Proposition 3. *Assume that* $\mathbf{K}$ *is invertible. Let* $\mathbf{B} := diag(\mathbf{b})$. *The optimizer* $\boldsymbol{\beta}^\star$ *of* (B.2) *is given by*

$$\boldsymbol{\beta}^\star = \lambda^{-1}\mathbf{B}\max\{\mathbf{0}, \mathbf{x}^\star\}, \tag{B.4}$$

where $\mathbf{x}^\star$ *is the solution of the following PLS*

$$\min\{\mathbf{0}, \mathbf{x}\} + \left(\lambda^{-1}\mathbf{B}\mathbf{K}\mathbf{B} + \mathbf{I}\right)\max\{\mathbf{0}, \mathbf{x}\} = \mathbb{1}. \tag{B.5}$$

Proof. Recall that $L(y, t)$ is the quadratic hinge loss, thus is differentiable. By setting the derivative of the objective in (B.3) to zero we get the following systems

$$-\sum_{i=1}^{n} \max\{\mathbf{0}, 1 - b_i\mathbf{K}'_{i\bullet}\boldsymbol{\beta}\}b_i\mathbf{K}_{i\bullet} + \lambda\mathbf{K}\boldsymbol{\beta} = \mathbf{0}. \tag{B.6}$$

Let us denote

$$\mathbf{x} := \mathbb{1} - \mathbf{B}\mathbf{K}\boldsymbol{\beta} \tag{B.7}$$

with $\mathbb{1}$ a size compatible all-one vector. Trivial manipulation on (B.6) leads to

$$\mathbf{x} + \lambda^{-1}\mathbf{B}\mathbf{K}\mathbf{B}\max\{\mathbf{0}, \mathbf{x}\} = \mathbb{1}, \tag{B.8}$$

or equivalently

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

Since $\mathbf{K}$ is invertible, by (B.7) the solution $\boldsymbol{\beta}^\star$ of (B.6) is calculated as

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

where the second equality follows (B.8). □

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 $\mathbf{x}^\star$ to (B.5). The expression (B.4) clearly indicates the sparse nature of $\boldsymbol{\beta}^\star$.

Notice that a similar Newton-type optimization method for solving the Prim-SVMs (B.3) has been proposed by Chappelle (2007), which solves the systems (B.6) via a Newton-type iterative scheme

$$\boldsymbol{\beta}^{k+1} = (\lambda \mathbf{I} + \mathbf{P}^k \mathbf{K})^{-1} \mathbf{P}^k \mathbf{B}, \tag{B.9}$$

where

$$\mathbf{P}^k := \begin{bmatrix} p(\beta_1^k) & & \\ & \ddots & \\ & & p(\beta_n^k) \end{bmatrix}, \tag{B.10}$$

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 (B.6) is by-passed 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 (B.7), 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 $b_i \mathbf{K}'_{i\bullet} \boldsymbol{\beta} < 1$, i.e., the loss on this point is non-zero.

B.0.2 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 (B.6). 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 2. For each data set, we construct the RBF heat kernel. The settings of parameter λ are given in the middle of Table 2. 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 2. 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 2: 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 Chapppelle's method for solving the gradient equation systems (B.6). 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
svmguide1	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