Efficient Euclidean Projections in Linear Time

Jun Liu J.LIU@ASU.EDU
Jieping Ye JIEPING.YE@ASU.EDU
Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287, USA

Abstract

We consider the problem of computing the Euclidean projection of a vector of length n onto a closed convex set including the ℓ_1 ball and the specialized polyhedra employed in (Shalev-Shwartz & Singer, 2006). These problems have played building block roles in solving several ℓ_1-norm based sparse learning problems. Existing methods have a worst-case time complexity of $O(n \log n)$. In this paper, we propose to cast both Euclidean projections as root finding problems associated with specific auxiliary functions, which can be solved in linear time via bisection. We further make use of the special structure of the auxiliary functions, and propose an improved bisection algorithm. Empirical studies demonstrate that the proposed algorithms are much more efficient than the competing ones for computing the projections.

1. Introduction

The Euclidean projection of a vector $\mathbf{v} \in \mathbb{R}^n$ onto a set $G \subseteq \mathbb{R}^n$ is defined as:

$$\pi_G(\mathbf{v}) = \arg\min_{\mathbf{x} \in G} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2, \tag{1}$$

where $\|.\|$ is the Euclidean (ℓ_2) norm. Since the objective function in (1) is strictly convex, its solution is unique for a closed and convex set G. When the set G is simple, e.g., the hyperplane, the halfspace, and the rectangle, the problem in (1) has an analytical solution (Boyd & Vandenberghe, 2004). However, for a general closed and convex set G, the problem in (1) does not admit an analytical solution. For example, when G is a general polyhedra, (1) leads to a Quadratic Programming problem.

In this paper, we address the problem of computing the Euclidean projection onto the following two closed and convex sets: (see Fig. 1 for an illustration of G_2):

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009. Copyright 2009 by the author(s)/owner(s).

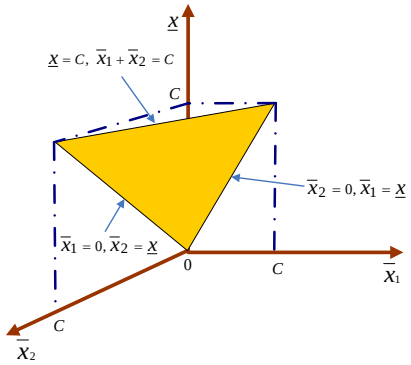

Figure 1. Illustration of the set G_2 in the three dimension case ($p = 2, n = 3$). G_2 is the region that is bounded by the following three lines: 1) $\overline{x}_1 = 0, \overline{x}_2 = \underline{x}$; 2) $\overline{x}_2 = 0, \overline{x}_1 = \underline{x}$; and 3) $\underline{x} = C, \overline{x}_1 + \overline{x}_2 = C$.

$$G_1 = \{\mathbf{x} \in \mathbb{R}^n | \ \|\mathbf{x}\|_1 \leq z\}, \tag{2}$$

$$G_2 = \{\mathbf{x} = (\overline{\mathbf{x}}, \underline{\mathbf{x}}) \in \mathbb{R}^n, \overline{\mathbf{x}} \in \mathbb{R}^p, \underline{\mathbf{x}} \in \mathbb{R}^{n-p} | \ \overline{\mathbf{x}} \geq 0, \underline{\mathbf{x}} \geq 0, \|\overline{\mathbf{x}}\|_1 = \|\underline{\mathbf{x}}\|_1 \leq C\}, \tag{3}$$

where $\|.\|_1$ denotes the ℓ_1 norm, and $z > 0$ and $C > 0$ denote the radiuses of the $\|.\|_1$ balls. These two Euclidean projections have played building block roles in solving several ℓ_1-norm based sparse learning problems (Tibshirani, 1996; Koh et al., 2007; Ng, 2004; Duchi et al., 2008; Shalev-Shwartz & Singer, 2006; Shalev-Shwartz, 2007).

The Euclidean projection onto the ℓ_1 ball (G_1) can be applied to solve the ℓ_1 ball constrained learning problem:

$$\min_{\mathbf{x}: \|\mathbf{x}\|_1 \leq z} \text{loss}(\mathbf{x}), \tag{4}$$

where loss(.) is a given convex loss function. For example, setting loss(.) to the least squares loss leads to the well-known Lasso problem (Tibshirani, 1996); and setting loss(.) to the empirical logistic loss leads to the ℓ_1 ball constrained logistic regression problem (Koh et al., 2007).

The use of the ℓ_1 ball constraint (or equivalently the ℓ_1 norm regularization) results in sparse solutions and empirical success in various applications (Candès & Wakin,

2008; Donoho, 2006; Ng, 2004; Koh et al., 2007; Shalev-Shwartz & Srebro, 2008; Tibshirani, 1996). To solve (4) in the large-scale scenario, one may rely on the first-order methods—those using at each iteration function values and (sub)gradients only. Well-known first-order methods include subgradient descent, gradient descent, and Nesterov's optimal method (Nesterov, 2003; Nemirovski, 1994). When applied to solve (4), one key building block is the Euclidean projection onto the ℓ_1 ball. Duchi et al. (2008) proposed two algorithms for solving this projection. The first algorithm is motivated by the work of (Shalev-Shwartz & Singer, 2006; Shalev-Shwartz, 2007), and it works by sorting the elements of the vector, and then obtaining the projection by thresholding. The resulting algorithm has a time complexity of $O(n \log n)$. The second algorithm is based a modification of the randomized median finding algorithm (Cormen et al., 2001), and it has an expected (not the worst-case) time complexity of $O(n)$.

The Euclidean projection onto the specialized polyhedra G_2 was studied in (Shalev-Shwartz & Singer, 2006) in the context of learning to rank labels from a feedback graph. Shalev-Shwartz and Singer (2006) reformulated their proposed model as a Quadratic Programming problem subject to a set of affine constraints, in which the projection onto G_2 is a key building block. To solve this projection, Shalev-Shwartz and Singer (2006) proposed to first sort the elements of the vectors $\overline{\mathbf{v}}$ and $\underline{\mathbf{v}}$, then solve a piecewise quadratic minimization problem, and finally obtain the solution by thresholding. The resulting algorithm has a time complexity of $O(n \log n)$.

In this paper, we propose to cast both Euclidean projections as root finding problems associated with specific auxiliary functions. Based on such reformulations, we propose to solve both problems using bisection, which has a (worst-case) linear time complexity. We further make use of the special structure of the auxiliary functions, and propose an improved bisection algorithm. Empirical studies demonstrate the efficiency of the proposed algorithms in comparison with existing algorithms.

Notation: Vectors are denoted by lower case bold face letters, e.g., $\mathbf{x} \in \mathbb{R}^n$ is an n-dimensional vector. The i-th element of $\mathbf{x}$ is denoted by x_i. $\|.\|$ denotes the Euclidean (ℓ_2) norm, and $\|.\|_1$ denotes the ℓ_1 norm.

Organization: We cast both Euclidean projections as root finding problems in Section 2, propose efficient projection algorithms in Section 3, report empirical results in Section 4, and conclude this paper in Section 5.

2. Reformulation as Root Finding Problems

In this section, we reformulate both Euclidean projections as root finding problems using the Lagrangian technique.

2.1. Projection onto the ℓ_1 Ball

The problem of Euclidean projections onto the ℓ_1 ball G_1 can be formally defined as:

$$\pi_{G_1}(\mathbf{v}) = \arg \min_{\mathbf{x}:\|\mathbf{x}\|_1 \leq z} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2. \tag{5}$$

Introducing the Lagrangian variable λ for the constraint $\|\mathbf{x}\|_1 \leq z$, we can write the Lagrangian of (5) as

$$L(\mathbf{x}, \lambda) = \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2 + \lambda(\|\mathbf{x}\|_1 - z).$$

Let $\mathbf{x}^\star$ be the primal optimal point, and $\lambda^\star$ be the dual optimal point. The primal and dual optimal points $\mathbf{x}^\star$ and $\lambda^\star$ should satisfy $\|\mathbf{x}^\star\|_1 \leq z$ and $\lambda^\star \geq 0$. Moreover, the ℓ_1 ball constraint in (5) satisfies Slater's condition (Boyd & Vandenberghe, 2004, Section 5.2.3) since $\|\mathbf{0}\|_1 < z$. Therefore, strong duality holds, the primal and dual optimal values are equal, and we have the complementary slackness condition:

$$\lambda^\star(\|\mathbf{x}^\star\|_1 - z) = 0. \tag{6}$$

We show how to compute the primal optimal point $\mathbf{x}^\star$ when the dual optimal point $\lambda^\star$ is known. $\mathbf{x}^\star$ is the optimal solution to the following problem:

$$\mathbf{x}^\star = \arg \min_{\mathbf{x}} L(\mathbf{x}, \lambda^\star). \tag{7}$$

The problem in (7) has a unique solution, since $L(.,.)$ is strictly convex in the first argument. Since the variables in (7) are decoupled, we have

$$x_i^\star = \arg \min_{x_i} \frac{1}{2}(x_i - v_i)^2 + \lambda^\star(|x_i| - z),$$

which leads to

$$x_i^\star = \text{sgn}(v_i) \max(|v_i| - \lambda^\star, 0), \tag{8}$$

where $\text{sgn}(t)$ is the signum function: if $t > 0$, $\text{sgn}(t) = 1$; if $t < 0$, $\text{sgn}(t) = -1$; and if $t = 0$, $\text{sgn}(t) = 0$.

Our methodology for solving the problem (5) is to first solve the dual optimal point $\lambda^\star$, with which we can obtain the primal optimal point $\mathbf{x}^\star$ based on (8). We consider the following two cases: $\|\mathbf{v}\|_1 \leq z$ and $\|\mathbf{v}\|_1 > z$. We show in the following lemma that for the first case, we have $\lambda^\star = 0$:

Lemma 1 *If $\|\mathbf{v}\|_1 \leq z$, then the dual optimal point $\lambda^\star$ is zero and the primal optimal point $\mathbf{x}^\star$ is given by $\mathbf{x}^\star = \mathbf{v}$.*

Proof: We first prove $\lambda^\star = 0$ by contradiction. Assume that $\lambda^\star > 0$. It follows from (8) that

$$\|\mathbf{x}^\star\|_1 < z,$$

thus $\lambda^\star(\|\mathbf{x}^\star\|_1 - z) \neq 0$, which contradicts with the complementary slackness condition in (6). Therefore, $\lambda^\star = 0$. It follows from (8) that $\mathbf{x}^\star = \mathbf{v}$. □

Next, we focus on the case when $\|\mathbf{v}\|_1 > z$. We show that $\lambda^\star$ can be obtained by computing the root of an auxiliary function, as summarized in the following theorem:

Theorem 1 *If $\|\mathbf{v}\|_1 > z$, then the dual optimal point $\lambda^\star$ is positive, and $\lambda^\star$ is given by the unique root of*

$$f(\lambda) = \sum_{i=1}^{n} \max(|v_i| - \lambda, 0) - z. \tag{9}$$

Proof: We first prove that the auxiliary function $f(.)$ has a unique root, and then prove that $\lambda^\star > 0$ is the root of $f(.)$.

Denote the maximal absolute element in $\mathbf{v}$ by $v_{\max}$, that is, $v_{\max} = \max_i |v_i|$. It is clear that for any i, $\max(|v_i| - \lambda, 0)$ is continuous and monotonically decreasing in $(-\infty, +\infty)$ with respect to λ, and strictly decreasing in $(-\infty, |v_i|]$. Thus, $f(.)$ is continuous and monotonically decreasing in $(-\infty, +\infty)$, and strictly decreasing in $(-\infty, v_{\max}]$. From (9), we have $f(0) > 0$ (since $\|\mathbf{v}\|_1 > z$), $f(v_{\max} - z) \geq 0$, and $f(v_{\max}) = -z < 0$. According to the Intermediate Value Theorem, $f(.)$ has a unique root lying in the interval $[\max(0, v_{\max} - z), v_{\max})$.

Next, consider the dual optimal point $\lambda^\star$. First, we show that $\lambda^\star$ must be positive. Otherwise, if $\lambda^\star = 0$, we have $\mathbf{x}^\star = \mathbf{v}$ from (8), and $\|\mathbf{x}^\star\|_1 = \|\mathbf{v}\|_1 > z$, which contradicts with $\|\mathbf{x}^\star\|_1 \leq z$. It follows from the complementary slackness condition in (6) that $\|\mathbf{x}^\star\|_1 = z$. Following (8), we have $f(\lambda^\star) = 0$. □

2.2. Projection onto the Specialized Polyhedra

The Euclidean projection onto the specialized polyhedra G_2 can be formally defined as:

$$\pi_{G_2}(\mathbf{v}) = \arg\min_{\mathbf{x}=(\overline{\mathbf{x}},\underline{\mathbf{x}})} \left\{ \frac{1}{2}\|\overline{\mathbf{x}} - \overline{\mathbf{v}}\|^2 + \frac{1}{2}\|\underline{\mathbf{x}} - \underline{\mathbf{v}}\|^2 \right\}, \quad \text{s.t.} \quad \overline{\mathbf{x}} \geq 0, \underline{\mathbf{x}} \geq 0, \overline{\mathbf{x}}^T\overline{\mathbf{e}} = \underline{\mathbf{x}}^T\underline{\mathbf{e}} \leq C \tag{10}$$

where $\mathbf{v} = (\overline{\mathbf{v}}, \underline{\mathbf{v}}), \overline{\mathbf{x}}, \overline{\mathbf{v}}, \overline{\mathbf{e}} \in \mathbb{R}^p$, $\underline{\mathbf{x}}, \underline{\mathbf{v}}, \underline{\mathbf{e}} \in \mathbb{R}^{n-p}$, and the elements of $\overline{\mathbf{e}}$ and $\underline{\mathbf{e}}$ are all 1's.

Introducing the Lagrangian variables λ, $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ and η for the constraints $\overline{\mathbf{x}}^T\overline{\mathbf{e}} = \underline{\mathbf{x}}^T\underline{\mathbf{e}}, \overline{\mathbf{x}} \geq 0, \underline{\mathbf{x}} \geq 0$ and $\overline{\mathbf{x}}^T\overline{\mathbf{e}} \leq C$, respectively, we can write the Lagrangian of (10) as

$$L(\overline{\mathbf{x}}, \underline{\mathbf{x}}, \lambda, \boldsymbol{\mu}, \boldsymbol{\nu}) = \frac{1}{2}\|\overline{\mathbf{x}} - \overline{\mathbf{v}}\|^2 + \frac{1}{2}\|\underline{\mathbf{x}} - \underline{\mathbf{v}}\|^2 + \lambda(\overline{\mathbf{x}}^T\overline{\mathbf{e}} - \underline{\mathbf{x}}^T\underline{\mathbf{e}}) + \eta(\overline{\mathbf{x}}^T\overline{\mathbf{e}} - C) - \boldsymbol{\mu}^T\overline{\mathbf{x}} - \boldsymbol{\nu}^T\underline{\mathbf{x}}.$$

Let $\mathbf{x}^\star = (\overline{\mathbf{x}}^\star, \underline{\mathbf{x}}^\star)$ be the primal optimal point, and $\lambda^\star$, $\boldsymbol{\mu}^\star$, $\boldsymbol{\nu}^\star$ and $\eta^\star$ be the dual optimal points. It is easy to verify that the objective function in (10) is convex and differentiable, and the constraint functions are differentiable, thus the duality gap is zero, and any points that satisfy the KKT conditions are primal and dual optimal (Boyd & Vandenberghe, 2004, Chapter 5.5.3).

The KKT conditions for (10) are given by

$$\overline{x}_i^\star \geq 0, \overline{x}_i^\star = \overline{v}_i - \lambda^\star - \eta^\star + \mu_i^\star, \tag{11}$$

$$\underline{x}_j^\star \geq 0, \underline{x}_j^\star = \underline{v}_j + \lambda^\star + \nu_j^\star, \tag{12}$$

$$\mu_i^\star \geq 0, \mu_i^\star\overline{x}_i^\star = 0, \nu_j^\star \geq 0, \nu_j^\star\underline{x}_j^\star = 0, \tag{13}$$

$$\eta^\star \geq 0, \eta^\star(\sum_{i=1}^{p} \overline{x}_i^\star - C) = 0, \sum_{i=1}^{p} \overline{x}_i^\star \leq C, \tag{14}$$

$$\sum_{i=1}^{p} \overline{x}_i^\star = \sum_{j=1}^{n-p} \underline{x}_i^\star, \tag{15}$$

for all $i = 1, 2, \ldots, p$, and $j = 1, 2, \ldots, n-p$.

We show in the following lemma that the KKT conditions (11-15) can be simplified:

Lemma 2 *The conditions in (11-13) are equivalent to:*

$$\overline{x}_i^\star = \max(\overline{v}_i - \lambda^\star - \eta^\star, 0), \tag{16}$$

$$\underline{x}_j^\star = \max(\underline{v}_j + \lambda^\star, 0). \tag{17}$$

Proof: First, we show that if (11-13) hold, then (16-17) hold. If $\overline{v}_i - \lambda^\star - \eta^\star > 0$, we have $\overline{x}_i^\star > 0$ from (11), $\mu_i^\star = 0$ from (13), and thus $\overline{x}_i^\star = \overline{v}_i - \lambda^\star - \eta^\star$ by using (11); if $\overline{v}_i - \lambda^\star - \eta^\star \leq 0$, we have $\overline{x}_i^\star = 0$, because if $\overline{x}_i^\star > 0$, we have $\mu_i^\star > 0$ from (11), and $\overline{x}_i^\star = 0$ from (13), leading to a contradiction. Therefore, (16) holds. Following similar arguments, we can obtain (17).

Next, we assume (16-17) hold. By constructing $\mu_i^\star = \overline{x}_i^\star - (\overline{v}_i - \lambda^\star - \eta^\star)$ and $\nu_j^\star = \underline{x}_j^\star - (\underline{v}_j + \lambda^\star)$, we can verify that (11-13) hold. □

Based on Lemma 2, the KKT conditions (11-15) can be simplified as (14-17). In the following discussions, we focus on computing the primal and dual optimal points by the simplified KKT conditions. We define the following three auxiliary functions:

$$\overline{g}(\lambda) = \sum_{i=1}^{p} \max(\overline{v}_i - \lambda, 0) - C, \tag{18}$$

$$\underline{g}(\lambda) = \sum_{j=1}^{n-p} \max(\underline{v}_j + \lambda, 0) - C, \tag{19}$$

$$g(\lambda) = \overline{g}(\lambda) - \underline{g}(\lambda). \tag{20}$$

Using similar arguments as in the proof of Theorem 1, we obtain the following properties of these functions:

Lemma 3 *Denote $\overline{v}_{\max} = \max_i \overline{v}_i$ and $\underline{v}_{\max} = \max_j \underline{v}_j$.*

i) $\overline{g}(.)$ is continuous and monotonically decreasing in $(-\infty, +\infty)$, and strictly decreasing in $(-\infty, \overline{v}_{\max}]$;

ii) The root of $\overline{g}(.)$ is unique and lies in $[\overline{v}_{\max} - C, \overline{v}_{\max})$;

iii) $\underline{g}(.)$ is continuous and monotonically increasing in $(-\infty, +\infty)$, and strictly increasing in $[-\underline{v}_{\max}, +\infty)$;

iv) The root of $\underline{g}(.)$ is unique and lies in $(-\underline{v}_{\max}, -\underline{v}_{\max} + C]$;

v) $g(.)$ is continuous and monotonically decreasing in $(-\infty, +\infty)$, and strictly increasing in both $(-\infty, \overline{v}_{\max}]$ and $[-\underline{v}_{\max}, +\infty)$.

vi) $g(.)$ has at leat one root. Moreover, if $\overline{v}_{\max} > -\underline{v}_{\max}$, then the root of $g(.)$ is unique and lies in $(-\underline{v}_{\max}, \overline{v}_{\max})$.

We summarize the main results of this section in the following theorem:

Theorem 2 *Denote the unique roots of $\overline{g}(.)$ and $\underline{g}(.)$ by $\overline{\lambda}$ and $\underline{\lambda}$, respectively.*

i) If $\overline{\lambda} \geq \underline{\lambda}$, then by setting $\lambda^\star = \underline{\lambda}$, $\eta^\star = \overline{\lambda} - \underline{\lambda}$, and the primal points according to (16) and (17), the simplified KKT conditions hold;

ii) If $\overline{\lambda} < \underline{\lambda}$, then by setting $\eta^\star = 0$, $\lambda^\star$ as a root of $g(.)$, and the primal points according to (16) and (17), the simplified KKT conditions hold. Moreover, if $\overline{v}_{\max} > -\underline{v}_{\max}$, then $\lambda^\star$ is the unique root of $g(.)$; and if $\overline{v}_{\max} \leq -\underline{v}_{\max}$, then $\lambda^\star$ can be any element in $[\overline{v}_{\max}, -\underline{v}_{\max}]$, and meanwhile the primal optimal point $\mathbf{x}^\star$ is a zero vector.

Proof: In both cases, (16) and (17) hold, so we only need to verify the conditions in (14) and (15).

We first prove *i*). Since $\overline{\lambda}$ and $\underline{\lambda}$ are the roots of $\overline{g}(.)$ and $\underline{g}(.)$, respectively, we have $\overline{g}(\overline{\lambda}) = 0$ and $\underline{g}(\underline{\lambda}) = 0$. Since we set $\lambda^\star = \underline{\lambda}$ and $\eta^\star = \overline{\lambda} - \underline{\lambda}$, we have $\lambda^\star + \eta^\star = \overline{\lambda}$, and $\overline{g}(\lambda^\star + \eta^\star) = 0$. It follows from (16) and (18) that $\sum_{i=1}^{p} \overline{x}_i^\star = C$. Similarly, we can verify that $\sum_{j=1}^{n-p} \underline{x}_i^\star = C$. Therefore, (15) holds. Since $\eta^\star = \overline{\lambda} - \underline{\lambda} \geq 0$ due to $\overline{\lambda} \geq \underline{\lambda}$, and $\sum_{i=1}^{p} \overline{x}_i^\star = C$, we verify (14).

Next, we prove *ii*). We first show that $\lambda^\star \in (\overline{\lambda}, \underline{\lambda})$. According to the second property in Lemma 3, we have $\overline{\lambda} < \overline{v}_{\max}$; similarly, we have $\underline{\lambda} > -\underline{v}_{\max}$, according to the fourth property in Lemma 3. From the first property in Lemma 3, $\overline{g}(.)$ is strictly decreasing in $(-\infty, \overline{v}_{\max}]$ and monotonically decreasing in $(-\infty, +\infty)$, thus

$$\overline{g}(\underline{\lambda}) < \overline{g}(\overline{\lambda}) = 0. \tag{21}$$

Similarly, from the third property in Lemma 3, we have

$$\underline{g}(\overline{\lambda}) < \underline{g}(\underline{\lambda}) = 0. \tag{22}$$

It follows from (20), (21), and (22) that

$$g(\overline{\lambda}) = \overline{g}(\overline{\lambda}) - \underline{g}(\overline{\lambda}) > 0, \quad g(\underline{\lambda}) = \overline{g}(\underline{\lambda}) - \underline{g}(\underline{\lambda}) < 0.$$

Since $g(.)$ is continuous and monotonically decreasing (see the fifth property in Lemma 3), we have $\lambda^\star \in (\overline{\lambda}, \underline{\lambda})$. Following the similar arguments for obtaining (21), we have $\overline{g}(\lambda^\star) < 0$. Since we set $\eta^\star = 0$, we have $\sum_{i=1}^{p} \overline{x}_i^\star = \sum_{i=1}^{p} \max(\overline{v}_i - \lambda^\star, 0) = \overline{g}(\lambda^\star) + C < C$ by using (16) and (18). Therefore, (14) holds. It follows from (16-20) together with $g(\lambda^\star) = 0$ and $\eta^\star = 0$ that (15) holds.

From the sixth property in Lemma 3, the root of $g(.)$ is unique, if $\overline{v}_{\max} > -\underline{v}_{\max}$. If $\overline{v}_{\max} \leq -\underline{v}_{\max}$, then following (18) and (19), we have $\overline{g}(\lambda) = \underline{g}(\lambda) = -C$ and $g(\lambda) = 0$, $\forall \lambda \in [\overline{v}_{\max}, -\underline{v}_{\max}]$. Meanwhile, from (16) and (17), we have $\overline{x}_i^\star = \underline{x}_j^\star = 0, \forall i, j$, so that the primal optimal point $\mathbf{x}^\star$ is a zero vector. □

Following Theorem 2, we propose the following procedure for solving (10). First, we compute $\overline{\lambda}$ and $\underline{\lambda}$, the unique roots of $\overline{g}(.)$ and $\underline{g}(.)$. If $\overline{\lambda} \geq \underline{\lambda}$, we set $\lambda^\star = \underline{\lambda}$ and $\eta^\star = \overline{\lambda} - \underline{\lambda}$; if $\overline{\lambda} < \underline{\lambda}$ and $\overline{v}_{\max} > -\underline{v}_{\max}$, we set $\eta^\star = 0$ and compute $\lambda^\star$ as the unique root of $g(.)$; and if $\overline{\lambda} < \underline{\lambda}$ and $\overline{v}_{\max} \leq -\underline{v}_{\max}$, we set $\eta^\star = 0$ and choose any element in $[\overline{v}_{\max}, -\underline{v}_{\max}]$ as $\lambda^\star$. Finally, with the computed dual optimal points, we obtain the primal optimal points from (16) and (17).

3. Efficient Euclidean Projections

We reformulated the Euclidean projection as root finding problems in the last section. In this section, we present efficient algorithms for computing the roots. Specifically, we present the bisection algorithm in Section 3.1, and an improved bisection algorithm in Section 3.2.

3.1. Euclidean Projections by Bisection

We first propose to make use of bisection for computing the dual optimal points. Bisection works by producing a sequence of intervals of uncertainty with strictly decreasing lengths. It is known that, for any continuous function with a unique root in the interval $[a, b]$, the number of bisection iterations is upper-bounded by $\lceil \log_2(\frac{b-a}{\delta}) \rceil$, where $b - a$ is the length of the initial interval of uncertainty and δ denotes the pre-specified precision parameter.

When applying bisection for solving the roots of $f(.)$, $\overline{g}(.)$, $\underline{g}(.)$ and $g(.)$, it costs $O(n)$, $O(p)$, $O(n-p)$ and $O(n)$ floating operations (flops) for evaluating the function values once, respectively. From Lemma 3 and Theorems 1 and 2, the lengths of the initialized interval are upper-bounded by z, C, C and $|\overline{v}_{\max} + \underline{v}_{\max}|$, respectively, so the bisection iterations are upper-bounded by $\lceil \log_2(z/\delta) \rceil$, $\lceil \log_2(C/\delta) \rceil$, $\lceil \log_2(C/\delta) \rceil$ and $\lceil \log_2(|\overline{v}_{\max} + \underline{v}_{\max}|/\delta) \rceil$, respectively. Once the dual optimal point(s) have been computed, we can recover the primal optimal point $\mathbf{x}^\star$ from (8), (16) and (17) in $O(n)$ flops. Therefore, the time complexity for solving these two Euclidean projections by bisection is $O(n)$.

3.2. Euclidean Projections by Improved Bisection

Although bisection can solve the Euclidean projections in linear time, it has the limitation that its efficiency is independent of the function, and it cannot be improved even when the function is "well-behaved". The underlying reason is that, bisection only utilizes the signs of the function at the two boundary points, but not their values.

To improve the efficiency, one natural alternative is the interpolation method (Brent, 1971) that has a better local convergence rate than bisection. Well-known linear interpolation methods include Newton's method and Secant which have locally Q-quadratic and Q-superlinear convergence rates, respectively. However, both Newton's method and Secant can diverge. To overcome this limitation, the safeguarded methods (Brent, 1971) have been proposed. The interpolation methods such as Newton's method, Secant and their safeguarded versions are developed for solving the general purpose root finding problems. In this subsection, we aim at developing an efficient improved bisection algorithm for finding the root by explicitly using the "structure" of the auxiliary functions.

Due to similarities of these auxiliary functions, we take $f(.)$ as an example in the following discussions. We note that, the two key factors that influence the efficiency of the root finding algorithm are: (1) the cost for evaluating the function value, and (2) the number of iterations. In what follows, we detail how to reduce the cost for evaluating $f(\lambda)$ in Section 3.2.1 and reduce the number of iterations in Section 3.2.2.

For convenience of illustration, we denote $\mathbf{u} = |\mathbf{v}|$, that is, $u_i = |v_i|$, with which the auxiliary function $f(.)$ in (9) can be written as $f(\lambda) = \sum_{i=1}^{n} \max(u_i - \lambda, 0) - z$. We first reveal the convexity property of $f(\lambda)$. Since $\max(u_i - \lambda, 0)$ is convex for all i, the auxiliary function $f(\lambda)$ is convex, as summarized in the following lemma:

Lemma 4 *The auxiliary function $f(\lambda)$ in (9) is convex.*

3.2.1. Efficient Evaluation of $f(\lambda)$

In this subsection, we aim to reduce the computational cost for evaluating $f(.)$. Denote

$$R_\lambda = \{i | i \in [n], u_i > \lambda\},$$

we can write $f(\lambda)$ as

$$\begin{aligned} f(\lambda) &= \sum_{i=1}^{n} \max(u_i - \lambda, 0) - z \\ &= \sum_{i \in R_\lambda} (u_i - \lambda) - z = \sum_{i \in R_\lambda} u_i - \lambda |R_\lambda| - z, \end{aligned} \tag{23}$$

where $|R_\lambda|$ denotes the number of elements in R_λ.

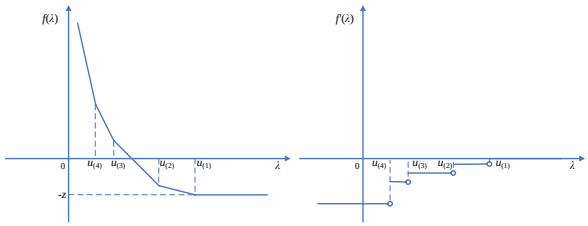

Figure 2. Illustration of the auxiliary function $f(\lambda)$ (left) and its subgradient $f'(\lambda)$ (right).

It is easy to verify that $f(\lambda)$ is a piece-wise linear function, as illustrated in the left figure of Fig. 2[1]. It is clear that $f(\lambda)$ is not differentiable at u_i, for $i = 1, 2, \ldots, n$. However, as revealed in Lemma 4, $f(\lambda)$ is convex, and we can define the subgradient (Nemirovski, 1994) of $f(\lambda)$ as

$$f'(\lambda) = -|R_\lambda|. \tag{24}$$

Thus, $f'(\lambda)$ is monotonically increasing and non-positive, and $f'(\lambda) = 0$ if and only if $\lambda \geq u_{(1)} = v_{\max}$. The right figure of Fig. 2 illustrates the subgradient $f'(\lambda)$, which is also a piece-wise linear function. From (23) and (24), we can rewrite $f(\lambda)$ as:

$$f(\lambda) = f'(\lambda)\lambda + b(\lambda), \tag{25}$$

where

$$b(\lambda) = \sum_{i \in R_\lambda} u_i - z$$

is the bias of the piece-wise linear function at λ. (25) implies that the efficient evaluation of $f(\lambda)$ lies in the efficient calculation of $f'(\lambda)$ and $b(\lambda)$.

Let the current interval of uncertainty be $[\lambda_1, \lambda_2]$, and $f'(\lambda_2)$ and $b(\lambda_2)$ have been computed. We show how to evaluate the value of $f(\lambda)$ for any $\lambda \in [\lambda_1, \lambda_2]$. Denote $U_\lambda = \{i | \lambda < u_i \leq \lambda_2\}$, we can compute $f'(\lambda)$ and $b(\lambda)$ as

$$\begin{aligned} f'(\lambda) &= -|U_\lambda| + f'(\lambda_2), \\ b(\lambda) &= \sum_{i \in U_\lambda} u_i + b(\lambda_2), \end{aligned}$$

which shows that we focus on those elements in the interval $(\lambda_1, \lambda_2]$ only for computing the subgradient $f'(\lambda)$ and the bias $b(\lambda)$ for any $\lambda \in [\lambda_1, \lambda_2]$. Note that the number of elements in the interval $(\lambda_1, \lambda_2]$ decreases when the iterative procedure proceeds (for example, the length of the interval is decreased by a factor of 2 in each iteration of bisection), thus reducing the computational cost for evaluating $f(\lambda)$.

[1]For illustration convenience, we denote $u_{(i)}$ as the i-th order statistic of $\mathbf{u}$, i.e., $u_{(1)} \geq u_{(2)} \geq \ldots \geq u_{(n)}$. However, in the proposed algorithm, we do not need to sort the elements in $\mathbf{u}$.

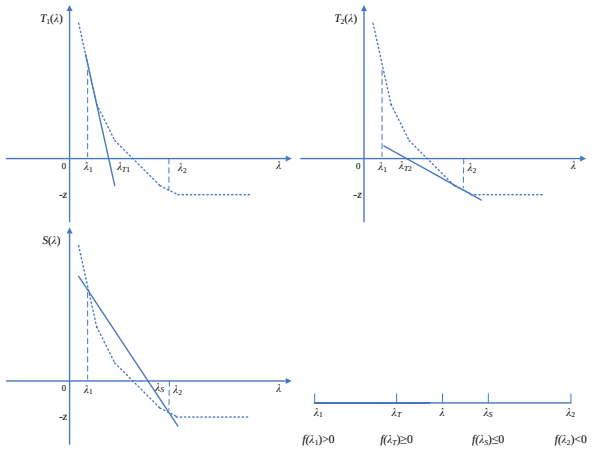

Figure 3. Illustration of the three constructed models and the relationship among the roots of the models. The dashed piecewise linear line denotes $f(.)$, and the solid line is the constructed model.

3.2.2. Reducing the Number of Iterations

To reduce the number of iterations, we propose to employ several models including Newton's method and Secant to obtain some approximate solutions for tightening the interval of uncertainty. We then apply bisection to this tightened interval to obtain a new interval of uncertainty. Our method is an improved bisection algorithm, which effectively integrates Newton's method and Secant in bisection. Our method can decrease the interval of uncertainty by a factor strictly larger than 2 in each iteration.

Let the current interval of uncertainty be $[\lambda_1, \lambda_2]$ ($f(\lambda_1) > 0$ and $f(\lambda_2) < 0$), and the following values have been obtained: the subgradients $f'(\lambda_1)$ and $f'(\lambda_2)$, the function values $f(\lambda_1)$ and $f(\lambda_2)$, and the biases $b(\lambda_1)$ and $b(\lambda_2)$. We construct three models for approximating $f(.)$ (see Fig. 3).

The first model corresponds to the line that passes through $(\lambda_1, f(\lambda_1))$ with derivative $f'(\lambda_1)$:

$$T_1(\lambda) = f(\lambda_1) + f'(\lambda_1)(\lambda - \lambda_1). \tag{26}$$

The second model corresponds to the line that passes through $(\lambda_2, f(\lambda_2))$ with derivative $f'(\lambda_2)$:

$$T_2(\lambda) = f(\lambda_2) + f'(\lambda_2)(\lambda - \lambda_2). \tag{27}$$

When $\lambda_1 \neq u_i$ for any i, $T_1(.)$ is the tangent line of $f(.)$ at λ_1. Similarly, when $\lambda_2 \neq u_i$ for any i, $T_2(.)$ is the tangent line of $f(.)$ at λ_2.

Based on the definition of $f'(\lambda)$ in (24) and $\lambda_1 < v_{\max}$, we have $f'(\lambda_1) < 0$. Therefore, the unique root of $T_1(.)$ is

$$\lambda_{T1} = \lambda_1 - f(\lambda_1)/f'(\lambda_1), \tag{28}$$

which satisfies $\lambda_{T1} > \lambda_1$ since $f(\lambda_1) > 0$ and $f'(\lambda_1) < 0$. Similarly, when $f'(\lambda_2)$ is nonzero, the unique root of $T_2(.)$ can be computed as:

$$\lambda_{T2} = \lambda_2 - f(\lambda_2)/f'(\lambda_2). \tag{29}$$

Since $f(.)$ is convex and $f'(.)$ is the subgradient of $f(.)$, the lines $T_1(.)$ and $T_2(.)$ always underestimate $f(.)$, i.e.,

$$f(\lambda_{T1}) \geq T_1(\lambda_{T1}) = 0,$$
$$f(\lambda_{T2}) \geq T_2(\lambda_{T2}) = 0.$$

Denote

$$\lambda_T = \max(\lambda_{T1}, \lambda_{T2}). \tag{30}$$

It follows that $f(\lambda_T) \geq 0$, and $\lambda_T > \lambda_1$. Thus, λ_T forms a tighter lower-bound of the interval of uncertainty than λ_1.

The third model is based on the line passing through the two points $(\lambda_1, f(\lambda_1))$ and $(\lambda_2, f(\lambda_2))$:

$$S(\lambda) = f(\lambda_2) + \frac{f(\lambda_2) - f(\lambda_1)}{\lambda_2 - \lambda_1}(\lambda - \lambda_2). \tag{31}$$

Since $f(\lambda_1) \neq f(\lambda_2)$ (note that, $f(\lambda_1) > 0, f(\lambda_2) < 0$), the unique root of $S(.)$ can be written as

$$\lambda_S = \lambda_2 - f(\lambda_2)\frac{\lambda_2 - \lambda_1}{f(\lambda_2) - f(\lambda_1)}, \tag{32}$$

where $\lambda_S < \lambda_2$, since $f(\lambda_2)\frac{\lambda_2-\lambda_1}{f(\lambda_2)-f(\lambda_1)} > 0$. From the convexity of $f(.)$, we have

$$\begin{aligned} f(\lambda_S) =& f\left(\frac{f(\lambda_2)}{f(\lambda_2) - f(\lambda_1)}\lambda_1 + \frac{-f(\lambda_1)}{f(\lambda_2) - f(\lambda_1)}\lambda_2\right) \\ &\leq 0. \end{aligned}$$

Thus, λ_S forms a tighter upper-bound of the interval of uncertainty than λ_2.

Since $f(.)$ is monotonically decreasing, we obtain the following relationship among λ_1, λ_2, λ_T, and λ_S (see Fig. 3):

$$\lambda_1 < \lambda_T \leq \lambda_S < \lambda_2, \tag{33}$$

where $\lambda_T = \lambda_S$ if and only if $f(\lambda_T) = f(\lambda_S) = 0$.

With the computed tighter interval of uncertainty $[\lambda_T, \lambda_S]$, we can choose λ used in bisection as the middle point of λ_T and λ_S:

$$\lambda = \frac{1}{2}(\lambda_T + \lambda_S). \tag{34}$$

The updated interval of uncertainty is $[\lambda, \lambda_S]$ if $f(\lambda) > 0$, and $[\lambda_T, \lambda]$ if $f(\lambda) < 0$. The length of interval of uncertainty is decreased by a factor strictly larger than 2, since

$$\lambda - \lambda_T = \lambda_S - \lambda = \frac{1}{2}(\lambda_S - \lambda_T) < \frac{1}{2}(\lambda_1 - \lambda_2). \tag{35}$$

Table 1. Illustration of the improved bisection algorithm: each row corresponds to an iteration; $[\lambda_1, \lambda_2]$ denotes the current interval, and $|U|$ denotes its size; λ_T is computed from the two models $T_1(.)$ and $T_2(.)$; λ_S is computed from the model $S(.)$; λ is the middle point of λ_T and λ_S; and the found root is in bold.

$\lvert U\rvert$	λ_1	λ_T	λ	λ_S	λ_2
10^5	0	0.79907	2.49512	4.19116	4.19641
1242	2.49512	2.72716	3.24086	3.75455	4.19116
502	2.72716	2.85927	2.93296	3.00665	3.24086
88	2.85927	2.89934	2.90139	2.90343	2.93296
1	2.89934	**2.90105**	2.90105	2.90105	2.90139

3.2.3. Discussions

The improved bisection algorithm enjoys the following two properties: (1) consistently decreasing computation cost for evaluating $f(.)$ with increasing iterations; and (2) fewer iterations than bisection, benefited by the good local convergence rate of Newton's method and Secant.

The improved bisection can also allow an initial guess of the root (denoted by λ_0), which can help reduce the number of iterations, if it is close to our target. Let the initialized interval be $[\lambda_1, \lambda_2]$, we can easily incorporate λ_0 into the improved bisection algorithm, by setting $\lambda_1 = \lambda_0$ if $f(\lambda_0) > 0$ and $\lambda_2 = \lambda_0$ otherwise. We note that, when applying the Euclidean projections for solving problems such as (4), the adjacent Euclidean projections usually have close dual optimal points. Therefore, we can use the root found in the previous projection as the "warm" start.

In deriving the improved bisection algorithm, we only make use of the piecewise linear and convex "structures" of the auxiliary function, and thus the improved bisection is applicable to $\overline{g}(.)$ and $\underline{g}(.)$, which enjoy these two "structures". By some careful deductions, this improved bisection algorithm can also be extended to solve the root of $g(.)$. The key observation is that $g(.)$ is the difference of the two convex and piecewise linear functions $\overline{g}(.)$ and $\underline{g}(.)$, so that we can efficiently evaluate $g(.)$ similar to $f(.)$. Moreover, following similar ideas in Section 3.2.2, we can construct models to obtain λ_T and λ_S to tighten the interval of uncertainty as follows. Let $[\lambda_1, \lambda_2]$ be the current interval of uncertainty. We obtain λ_T as the intersection of the tangent line of $\overline{g}(.)$ at λ_1 and the secant model of $\underline{g}(.)$ passing through $(\lambda_1, \underline{g}(\lambda_1))$ and $(\lambda_2, \underline{g}(\lambda_2))$, and λ_S as the intersection of the tangent line of $\underline{g}(.)$ at λ_2 and the secant model of $\overline{g}(.)$ passing through $(\lambda_1, \overline{g}(\lambda_1))$ and $(\lambda_2, \overline{g}(\lambda_2))$.

4. Experiments

To study the performance of the proposed projection algorithms, we randomly generated the input vector $\mathbf{v}$ according to two distributions: (1) normal distribution with mean 0 and standard deviation 1, and (2) uniform distribution in the interval $[-1, 1]$. We implement the proposed projection algorithms in C, and carry out the experiments on an Intel (R) Core (TM)2 Duo 3.00GHZ processor.

An Illustrative Example We first present an example to illustrate the improved bisection algorithm. In this experiment, we compute the Euclidean projection onto the ℓ_1 ball on a problem of size $n = 10^5$. We generate $\mathbf{v}$ from the normal distribution, and set $z = 100$. The result is presented in Table 1. We can observe from this table that the proposed improved bisection converges quite fast, and the computational cost (proportional to $|U|$) for evaluating $f(.)$ decreases rapidly.

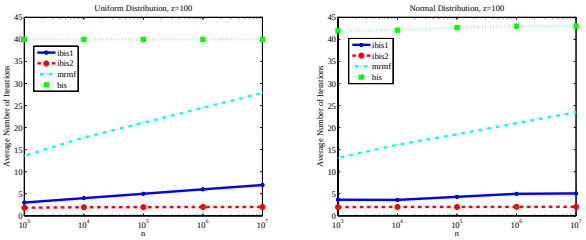

Figure 4. Comparison of the average number of iterations over 1000 runs. $\mathbf{v}$ is generated from the uniform distribution in the left figure, and from the normal distribution in the right figure.

Number of Iterations We compare the improved bisection (ibis) with bisection (bis), and the modified randomized median finding (mrmf) (Duchi et al., 2008), in terms of the number of iterations for solving the projection onto the ℓ_1 ball. For ibis, we try two different settings: (1) ibis1, which does not require an initial guess of the root, (2) ibis2, which employs the "warm" start, that is, the root found by the previous problem is used as a 'warm" start (we solved 1000 different problems for a fixed size n). We set $z = 100$, and report the results in Figure 4, where the average number of iterations over 1000 runs is shown. We can observe from these figures that: 1) the number of iterations by bisection is around 40; 2) the number of iterations by mrmf is significantly smaller than that of bisection, which validates the good practical behavior of the randomized median finding algorithm; 3) the number of iterations for ibis1 is within 7, which is less than that required by mrmf; and 4) by employing the "warm" start technique, the number of iterations for ibis2 is further reduced to about 2.

Computation Efficiency We report the total computational time (in seconds) for solving 1000 independent projection onto the ℓ_1 ball by different methods in Tables 2 and 3, from which we can observe that, all methods scale (roughly) linearly with n, and ibis1 is more efficient than bisection and mrmf. With a "warm" start technique, ibis2 is much more efficient than ibis1.

We also compare the improved bisection algorithm with the soft projections onto polyhedra (sopopo) proposed in

Table 2. The total computational time (in seconds) for solving 1000 independent projections onto the ℓ_1 ball: normal distribution with $z = 10$ (top half) and $z = 100$ (bottom half).

n	10^3	10^4	10^5	10^6	10^7
bis	0.0543	0.4323	4.691	78.35	788.9
mrmf	0.0130	0.2720	2.776	37.68	380.3
ibis1	0.0074	0.1276	1.509	19.62	196.2
ibis2	0.0024	0.0877	1.126	17.06	167.9
bis	0.1521	0.6178	4.926	78.52	790.3
mrmf	0.0319	0.2901	2.766	37.84	383.2
ibis1	0.0305	0.1843	1.541	19.61	196.5
ibis2	0.0195	0.0946	1.133	17.06	167.8

Table 3. The total computational time (in seconds) for solving 1000 independent projections onto the ℓ_1 ball: uniform distribution with $z = 10$ (top half) and $z = 100$ (bottom half).

n	10^3	10^4	10^5	10^6	10^7
bis	0.1247	0.7511	6.554	82.61	833.6
mrmf	0.0332	0.2941	2.992	37.99	389.0
ibis1	0.0286	0.1698	2.091	24.74	247.7
ibis2	0.0210	0.0946	1.332	17.46	173.7
bis	0.2030	1.1644	8.165	86.31	844.4
mrmf	0.0332	0.3373	3.187	39.43	394.7
ibis1	0.0266	0.1859	2.159	24.90	248.5
ibis2	0.0198	0.1135	1.416	17.48	175.4

(Shalev-Shwartz & Singer, 2006) for solving the projection onto the specialized polyhedra G_2. The results are presented in Table 4. We can observe from the table that ibis1 and ibis2 are more efficient than sopopo. The experimental results verify the efficiency of the proposed algorithms.

5. Conclusion

In this paper, we study the problem of Euclidean projections onto the ℓ_1 ball G_1 and the specialized polyhedra G_2. Our main results show that both Euclidean projections can be formulated as root finding problems. Based on such reformulation, we can solve the Euclidean projections in (the worst-case) linear time via bisection. We further explore the piecewise linear and convex "structures" of the auxiliary functions, and propose the improved bisection algorithm. Empirical studies show that our proposed algorithms are much more efficient that the competing ones.

We are currently investigating the ℓ_1 ball constrained sparse learning problems by the first-order methods, which include the proposed Euclidean projections as a key building block. We plan to extend the proposed algorithms to efficiently solve the projection $\pi_G(\mathbf{v} + \boldsymbol{\varepsilon})$ when the result of $\pi_G(\mathbf{v})$ is known and $\boldsymbol{\varepsilon}$ has sparse structure, which can be useful in the scenario of online learning (Duchi et al., 2008; Shalev-Shwartz, 2007). We also plan to explore efficient entropic projections (Shalev-Shwartz, 2007), which uses the entropy instead of the Euclidean norm in (1).

Table 4. The total computational time (in seconds) for solving 1000 independent projections onto the specialized polyhedra G_2 (we set $p = n/2$ and $C = 10$): normal distribution (top half) and uniform distribution (bottom half).

n	10^3	10^4	10^5	10^6	10^7
sopopo	0.0934	0.8401	9.574	142.5	1593
ibis1	0.0274	0.1482	1.774	22.77	226.9
ibis2	0.0200	0.0920	1.249	18.50	185.2
sopopo	0.1147	0.9077	10.07	143.5	1605
ibis1	0.0288	0.1725	2.084	24.73	258.2
ibis2	0.0216	0.1002	1.364	18.57	191.4

Acknowledgments

This work was supported by NSF IIS-0612069, IIS-0812551, CCF-0811790, and NGA HM1582-08-1-0016.

References

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Brent, R. (1971). An algorithm with guaranteed convergence for finding a zero of a function. *Computer Journal*, *14*, 422–425.

Candès, E., & Wakin, M. (2008). An introduction to compressive sampling. *IEEE Signal Processing Magazine*, *25*, 21–30.

Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2001). *Introduction to algorithms*. MIT Press.

Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*, 1289–1306.

Duchi, J., Shalev-Shwartz, S., Singer, Y., & Tushar, C. (2008). Efficient projection onto the l_1-ball for learning in high dimensions. *International Conference on Machine Learning* (pp. 272–279).

Koh, K., Kim, S., & Boyd, S. (2007). An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine Learning Research*, *8*, 1519–1555.

Nemirovski, A. (1994). *Efficient methods in convex programming*. Lecture Notes.

Nesterov, Y. (2003). *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers.

Ng, A. (2004). Feature selection, ℓ_1 vs. ℓ_2 regularization, and rotational invariance. *International Conference on Machine Learning* (pp. 78–85).

Shalev-Shwartz, S. (2007). *Online learning: Theory, algorithms, and applications*. Doctoral dissertation, Hebrew University.

Shalev-Shwartz, S., & Singer, Y. (2006). Efficient learning of label ranking by soft projections onto polyhedra. *Journal of Machine Learning Research*, *7*, 1567–1599.

Shalev-Shwartz, S., & Srebro, N. (2008). *Iterative loss minimization with ℓ_1-norm constraint and guarantees on sparsity* (Technical Report). TTI.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, *58*, 267–288.