Constrained convex minimization via model-based excessive gap

Quoc Tran-Dinh and **Volkan Cevher**
Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL), CH1015-Lausanne, Switzerland
{quoc.trandinh, volkan.cevher}@epfl.ch

Abstract

We introduce a model-based excessive gap technique to analyze first-order primal-dual methods for constrained convex minimization. As a result, we construct new primal-dual methods with optimal convergence rates on the objective residual and the primal feasibility gap of their iterates separately. Through a dual smoothing and prox-function selection strategy, our framework subsumes the augmented Lagrangian, and alternating methods as special cases, where our rates apply.

1 Introduction

In [1], Nesterov introduced a primal-dual technique, called the *excessive gap*, for constructing and analyzing first-order methods for nonsmooth and unconstrained convex optimization problems. This paper builds upon the same idea for constructing and analyzing algorithms for the following a class of *constrained* convex problems, which captures a surprisingly broad set of applications [2, 3, 4, 5]:

$$f^\star := \min_{\mathbf{x}} \left\{ f(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathcal{X} \right\}, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper, closed and convex function; $\mathcal{X} \subseteq \mathbb{R}^n$ is a nonempty, closed and convex set; and $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ are given.

In the sequel, we show how Nesterov's excessive gap relates to the smoothed gap function for a variational inequality that characterizes the optimality condition of (1). In the light of this connection, we enforce a simple linear model on the excessive gap, and use it to develop efficient first-order methods to numerically approximate an optimal solution $\mathbf{x}^\star$ of (1). Then, we rigorously characterize how the following structural assumptions on (1) affect their computational efficiency:

Structure 1: Decomposability. We say that problem (1) is *p-decomposable* if its objective function f and its feasible set $\mathcal{X}$ can be represented as follows:

$$f(\mathbf{x}) := \sum\nolimits_{i=1}^{p} f_i(\mathbf{x}_i), \ \text{ and } \ \mathcal{X} := \prod\nolimits_{i=1}^{p} \mathcal{X}_i, \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_i}$, $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$, $f_i : \mathbb{R}^{n_i} \to \mathbb{R} \cup \{+\infty\}$ is proper, closed and convex for $i = 1, \ldots, p$, and $\sum_{i=1}^p n_i = n$. Decomposability naturally arises in machine learning applications such as group sparsity linear recovery, consensus optimization, and the dual formulation of empirical risk minimization problems [5]. As an important example, the composite convex minimization problem $\min_{\mathbf{x}_1}\{f_1(\mathbf{x}_1) + f_2(\mathbf{K}\mathbf{x}_1)\}$ can be cast into (1) with a 2-decomposable structure using intermediate variables $\mathbf{x}_2 = \mathbf{K}\mathbf{x}_1$ to represent $\mathbf{A}\mathbf{x} = \mathbf{b}$. Decomposable structure immediately supports parallel and distributed implementations in synchronous hardware architectures.

Structure 2: Proximal tractability. By proximal tractability, we mean that the computation of the following operation with a given proper, closed and convex function g is "efficient" (e.g., by a closed form solution or by polynomial time algorithms) [6]:

$$\mathrm{prox}_g(\mathbf{z}) := \mathrm{arg}\min_{\mathbf{w}}\{g(\mathbf{w}) + (1/2)\|\mathbf{w} - \mathbf{z}\|^2\}. \tag{3}$$

When the constraint $\mathbf{z} \in \mathcal{Z}$ is available, we consider the proximal operator of $g(\cdot) + \delta_{\mathcal{Z}}(\cdot)$ instead of g, where $\delta_{\mathcal{Z}}$ is the indicator function of $\mathcal{Z}$. Many smooth and non-smooth convex functions have tractable proximal operators such as norms, and the projection onto a simple set [3, 7, 4, 5].

Scalable algorithms for (1) **and their limitations.** We can obtain scalable numerical solutions of (1) when we augment the objective f with simple penalty functions on the constraints. Despite the fundamental difficulties in choosing the penalty parameter, this approach enhances our computational capabilities as well as numerical robustness since we can apply modern proximal gradient, alternating direction, and primal-dual methods. Unfortunately, existing approaches invariably feature one or both of the following two limitations:

Limitation 1: Non-ideal convergence characterizations. Ideally, the convergence rate characterization of a first-order algorithm for solving (1) must simultaneously establish for its iterates $\mathbf{x}^k \in \mathcal{X}$ both on the objective residual $f(\mathbf{x}^k) - f^\star$ and on the primal feasibility gap $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|$ of its linear constraints. The constraint feasibility is critical so that the primal convergence rate has any significance. Rates on a joint of the objective residual and feasibility gap is not necessarily meaningful since (1) is a constrained problem and $f(\mathbf{x}^k) - f^\star$ can easily be negative at all times as compared to the unconstrained setting, where we trivially have $f(\mathbf{x}^k) - f^\star \geq 0$.

Hitherto, the convergence results of state-of-the-art methods are far from ideal; see Table 1 in [28]. Most algorithms have guarantees in the ergodic sense [8, 9, 10, 11, 12, 13, 14] with non-optimal rates, which diminishes the practical performance; they rely on special function properties to improve convergence rates on the function and feasibility [12, 15], which reduces the scope of their applicability; they provide rates on dual functions [16], or a weighted primal residual and feasibility score [13], which does not necessarily imply convergence on the primal residual or the feasibility; or they obtain convergence rate on the gap function value sequence composed both the primal and dual variables via variational inequality and gap function characterizations [8, 10, 11], where the rate is scaled by a diameter parameter of the dual feasible set which is not necessary bounded.

Limitation 2: Computational inflexibility. Recent theoretical developments customize algorithms to special function classes for scalability, such as convex functions with global Lipschitz gradient and strong convexity. Unfortunately, these algorithms often require knowledge of function class parameters (e.g., the Lipschitz constant and the strong convexity parameter); they do not address the full scope of (1) (e.g., with self-concordant [barrier] functions or fully non-smooth decompositions); and they often have complicated algorithmic implementations with backtracking steps, which can create computational bottlenecks. These issues are compounded by their penalty parameter selection, which can significantly decrease numerical efficiency [17]. Moreover, they lack a natural ability to handle p-decomposability in a parallel fashion at optimal rates.

Our specific contributions. To this end, this paper addresses the question: "Is it possible to efficiently solve (1) using only the proximal tractability assumption with rigorous global convergence rates on the objective residual and the primal feasibility gap?" The answer is indeed positive provided that there exists a solution in a bounded feasible set $\mathcal{X}$. Surprisingly, we can still leverage favorable function classes for fast convergence, such as strongly convex functions, and exploit p-decomposability at optimal rates.

Our characterization is radically different from existing results, such as in [18, 8, 19, 9, 10, 11, 12, 13]. Specifically, we unify primal-dual methods [20, 21], smoothing (both for Bregman distances and for augmented Lagrangian functions) [22, 21], and the excessive gap function technique [1] in one. As a result, we develop an efficient algorithmic framework for solving (1), which covers augmented Lagrangian method [23, 24], [preconditioned] alternating direction method-of-multipliers ([P]ADMM) [8] and fast dual descent methods [18] as special cases.

Based on the new technique, we establish rigorous convergence rates for a few well-known primal-dual methods, which is optimal (in the sense of first order black-box models [25]) given our particular assumptions. We also discuss adaptive strategies for trading-off between the objective residual $|f(\mathbf{x}^k) - f^\star|$ and the feasibility gap $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|$, which enhance practical performance. Finally, we describe how strong convexity of f can be exploited, and numerically illustrate theoretical results.

2 Preliminaries

2.1. A semi-Bregman distance. Given a nonempty, closed and convex set $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$, a nonnegative, continuous and μ_b-strongly convex function b is called a μ_b-proximity function (or prox-function) of $\mathcal{Z}$ if $\mathcal{Z} \subseteq \operatorname{dom}(b)$. Then $\mathbf{z}^c := \operatorname{argmin}_{\mathbf{z}\in\mathcal{Z}} b(\mathbf{z})$ exists and is unique, called the center point of b. Given a smooth μ_b-prox-function b of $\mathcal{Z}$ (with $\mu_b = 1$), we define $d_b(\mathbf{z}, \hat{\mathbf{z}}) := b(\hat{\mathbf{z}}) - b(\mathbf{z}) - \nabla b(\mathbf{z})^T(\hat{\mathbf{z}} - \mathbf{z})$, $\forall \mathbf{z}, \hat{\mathbf{z}} \in \operatorname{dom}(b)$, as the Bregman distance between $\mathbf{z}$ and $\hat{\mathbf{z}}$ given b. As an example, with $b(\mathbf{z}) := (1/2)\|\mathbf{z}\|_2^2$, we have $d_b(\mathbf{z}, \hat{\mathbf{z}}) = (1/2)\|\mathbf{z} - \hat{\mathbf{z}}\|_2^2$, which is the Euclidean distance.

In order to unify both the Bregman distance and augmented Lagrangian smoothing methods, we introduce a new semi-Bregman distance $d_b(\mathbf{Sx}, \mathbf{Sx}^c)$ between $\mathbf{x}$ and $\mathbf{x}^c$, given matrix $\mathbf{S}$. Since $\mathbf{S}$ is not necessary square, we use the prefix "semi" for this measure. We also denote by:

$$D_{\mathcal{X}}^{\mathbf{S}} := \sup\{d_b(\mathbf{Sx}, \mathbf{Sx}^c) : \mathbf{x}, \mathbf{x}^c \in \mathcal{X}\}, \tag{4}$$

the semi-diameter of $\mathcal{X}$. If $\mathcal{X}$ is bounded, then $0 \leq D_{\mathcal{X}}^{\mathbf{S}} < +\infty$.

2.2. The dual problem of (1)**.** Let $\mathcal{L}(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}) + \mathbf{y}^T(\mathbf{Ax} - \mathbf{b})$ be the Lagrange function of (1), where $\mathbf{y} \in \mathbb{R}^m$ is the Lagrange multipliers. The dual problem of (1) is defined as:

$$g^\star := \max_{\mathbf{y} \in \mathbb{R}^m} g(\mathbf{y}), \tag{5}$$

where g is the dual function, which is defined as:

$$g(\mathbf{y}) := \min_{\mathbf{x} \in \mathcal{X}} \{f(\mathbf{x}) + \mathbf{y}^T(\mathbf{Ax} - \mathbf{b})\}. \tag{6}$$

For $\mathbf{y} \in \mathbb{R}^m$, let us denote by $\mathbf{x}^\star(\mathbf{y})$ the solution of (6). Corresponding to $\mathbf{x}^\star(\mathbf{y})$, we also define the domain of g as $\mathrm{dom}\,(g) := \{\mathbf{y} \in \mathbb{R}^m : \mathbf{x}^\star(\mathbf{y}) \text{ exists}\}$. If f is continuous on $\mathcal{X}$ and if $\mathcal{X}$ is bounded, then $\mathbf{x}^\star(\mathbf{y})$ exists for all $\mathbf{y} \in \mathbb{R}^m$. Unfortunately, g is nonsmooth, and numerical solutions of (5) are difficult [25]. In general, we have $g(\mathbf{y}) \leq f(\mathbf{x})$ which is the weak-duality condition in convex optimization. To guarantee strong duality, i.e., $f^\star = g^\star$ for (1) and (5), we need an assumption:

Assumption A. 1. *The solution set $\mathcal{X}^\star$ of* (1) *is nonempty. The function f is proper, closed and convex. In addition, either $\mathcal{X}$ is a polytope or the Slater condition holds, i.e.:* $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b}\} \cap \mathrm{relint}(\mathcal{X}) \neq \emptyset$*, where* $\mathrm{relint}(\mathcal{X})$ *is the relative interior of $\mathcal{X}$.*

Under Assumption A.1, the solution set $\mathcal{Y}^\star$ of (5) is also nonempty and bounded. Moreover, the strong duality holds, i.e., $f^\star = g^\star$. Any point $(\mathbf{x}^\star, \mathbf{y}^\star) \in \mathcal{X}^\star \times \mathcal{Y}^\star$ is a primal-dual solution to (1) and (5), and is also a saddle point of $\mathcal{L}$, i.e., $\mathcal{L}(\mathbf{x}^\star, \mathbf{y}) \leq \mathcal{L}(\mathbf{x}^\star, \mathbf{y}^\star) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}^\star), \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathbb{R}^m$.

2.3. Mixed-variational inequality formulation and the smoothed gap function. We use $\mathbf{w} := [\mathbf{x}, \mathbf{y}] \in \mathbb{R}^n \times \mathbb{R}^m$ to denote the primal-dual variable, $F(\mathbf{w}) := \begin{bmatrix} \mathbf{A}^T\mathbf{y} \\ \mathbf{b} - \mathbf{Ax} \end{bmatrix}$ to denote a partial Karush-Kuhn-Tucker (KKT) mapping, and $\mathcal{W} := \mathcal{X} \times \mathbb{R}^m$. Then, we can write the optimality condition of (1) as:

$$f(\mathbf{x}) - f(\mathbf{x}^\star) + F(\mathbf{w}^\star)^T(\mathbf{w} - \mathbf{w}^\star) \geq 0, \ \ \forall \mathbf{w} \in \mathcal{W}, \tag{7}$$

which is known as the *mixed-variational inequality* (MVIP) [26]. If we define:

$$G(\mathbf{w}^\star) := \max_{\mathbf{w} \in \mathcal{W}} \left\{f(\mathbf{x}^\star) - f(\mathbf{x}) + F(\mathbf{w}^\star)^T(\mathbf{w}^\star - \mathbf{w})\right\}, \tag{8}$$

then G is known as the Auslender gap function of (7) [27]. By the definition of F, we can see that:

$$G(\mathbf{w}^\star) := \max_{[\mathbf{x}, \mathbf{y}] \in \mathcal{W}} \left\{f(\mathbf{x}^\star) - f(\mathbf{x}) - (\mathbf{Ax} - \mathbf{b})^T\mathbf{y}^\star\right\} = f(\mathbf{x}^\star) - g(\mathbf{y}^\star) \geq 0.$$

It is clear that $G(\mathbf{w}^\star) = 0$ if and only if $\mathbf{w}^\star := [\mathbf{x}^\star, \mathbf{y}^\star] \in \mathcal{W}^\star := \mathcal{X}^\star \times \mathcal{Y}^\star$—i.e., the strong duality.

Since G is generally nonsmooth, we strictly smooth it by adding an augmented term:

$$d_{\gamma\beta}(\mathbf{w}) \equiv d_{\gamma\beta}(\mathbf{x}, \mathbf{y}) := \gamma d_b(\mathbf{Sx}, \mathbf{Sx}^c) + (\beta/2)\|\mathbf{y}\|^2, \tag{9}$$

where d_b is a Bregman distance, $\mathbf{S}$ is a given matrix, and $\gamma, \beta > 0$ are two *smoothness parameters*. The smoothed gap function for G is defined as:

$$G_{\gamma\beta}(\bar{\mathbf{w}}) := \max_{\mathbf{w} \in \mathcal{W}} \left\{f(\bar{\mathbf{x}}) - f(\mathbf{x}) + F(\bar{\mathbf{w}})^T(\bar{\mathbf{w}} - \mathbf{w}) - d_{\gamma\beta}(\mathbf{w})\right\}, \tag{10}$$

where F is defined in (7). By the definition of G and $G_{\gamma\beta}$, we can easily show that:

$$G_{\gamma\beta}(\bar{\mathbf{w}}) \leq G(\bar{\mathbf{w}}) \leq G_{\gamma\beta}(\bar{\mathbf{w}}) + \max\{d_{\gamma\beta}(\mathbf{w}) : \mathbf{w} \in \mathcal{W}\}, \tag{11}$$

which is key to develop the algorithm in the next section.

Problem (10) is convex, and its solution $\mathbf{w}_{\gamma\beta}^\star(\bar{\mathbf{w}})$ can be computed as:

$$\mathbf{w}_{\gamma\beta}^\star(\bar{\mathbf{w}}) := [\mathbf{x}_\gamma^\star(\bar{\mathbf{y}}), \mathbf{y}_\beta^\star(\bar{\mathbf{x}})] \Leftrightarrow \begin{cases} \mathbf{x}_\gamma^\star(\bar{\mathbf{y}}) := \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \left\{f(\mathbf{x}) + \mathbf{y}^T(\mathbf{Ax} - \mathbf{b}) + \gamma d_b(\mathbf{Sx}, \mathbf{Sx}^c)\right\} \\ \mathbf{y}_\beta^\star(\bar{\mathbf{x}}) := \beta^{-1}(\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}). \end{cases} \tag{12}$$

In this case, the following concave function:

$$g_\gamma(\mathbf{y}) := \min_{\mathbf{x}\in\mathcal{X}} \left\{ f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \gamma d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}^c) \right\}, \tag{13}$$

can be considered as a smooth approximation of the dual function g defined by (6).

2.4. Bregman distance smoother vs. augmented Lagrangian smoother. Depending on the choice of $\mathbf{S}$ and $\mathbf{x}^c$, we deal with two smoothers as follows:

1. If we choose $\mathbf{S} = \mathbb{I}$, the identity matrix, and $\mathbf{x}^c$ is then center point of b, then we obtain a Bregman distance smoother.
2. If we choose $\mathbf{S} = \mathbf{A}$, and $\mathbf{x}^c \in \mathcal{X}$ such that $\mathbf{A}\mathbf{x}^c = \mathbf{b}$, then we have the augmented Lagrangian smoother.

Clearly, with both smoothing techniques, the function g_γ is smooth and concave. Its gradient is Lipschitz continuous with the Lipschitz constant $L_\gamma^g := \gamma^{-1}\|\mathbf{A}\|^2$ and $L_\gamma^g := \gamma^{-1}$, respectively.

3 Construction and analysis of a class of first-order primal-dual algorithms

3.1. Model-based excessive gap technique for (1)**.** Since $G(\mathbf{w}^\star) = 0$ iff $\mathbf{w}^\star = [\mathbf{x}^\star, \mathbf{y}^\star]$ is a primal-dual optimal solution of (1)-(5). The goal is to construct a sequence $\{\bar{\mathbf{w}}^k\}$ such that $G(\bar{\mathbf{w}}^k) \to 0$, which implies that $\{\bar{\mathbf{w}}^k\}$ converges to $\mathbf{w}^\star$. As suggested by (11), if we can construct two sequences $\{\bar{\mathbf{w}}^k\}$ and $\{(\gamma_k, \beta_k)\}$ such that $G_{\gamma_k\beta_k}(\bar{\mathbf{w}}^k) \to 0^+$ as $\gamma_k\beta_k \downarrow 0^+$, then $G(\bar{\mathbf{w}}^k) \to 0$.

Inspired by Nesterov's excessive gap idea in [1], we construct the following model-based excessive gap condition for (1) in order to achieve our goal.

Definition 1 (Model-based Excessive Gap)**.** *Given $\bar{\mathbf{w}}^k \in \mathcal{W}$ and $(\gamma_k, \beta_k) > 0$, a new point $\bar{\mathbf{w}}^{k+1} \in \mathcal{W}$ and $(\gamma_{k+1}, \beta_{k+1}) > 0$ with $\gamma_{k+1}\beta_{k+1} < \gamma_k\beta_k$ is said to reduce the primal-dual gap if:*

$$G_{k+1}(\bar{\mathbf{w}}^{k+1}) \le (1 - \tau_k)G_k(\bar{\mathbf{w}}^k) - \psi_k, \tag{14}$$

where $G_k := G_{\gamma_k\beta_k}$, $\tau_k \in [0, 1)$ and $\psi_k \ge 0$.

From Definition 1, if $\left\{\bar{\mathbf{w}}^k\right\}$ and $\{(\gamma_k, \beta_k)\}$ satisfy (14), then we have $G_k(\bar{\mathbf{w}}^k) \le \omega_k G_0(\bar{\mathbf{w}}^0) - \Psi_k$ by induction, where $\omega_k := \prod_{j=0}^{k-1}(1-\tau_j)$ and $\Psi_k := \psi_k + \sum_{j=1}^k \prod_{i=j}^k (1-\tau_i)\psi_{j-1}$. If $G_0(\bar{\mathbf{w}}^0) \le 0$, then we can bound the objective residual $|f(\bar{\mathbf{x}}^k) - f^\star|$ and the primal feasibility $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|$ of (1):

Lemma 1 ([28])**.** *Let $G_{\gamma\beta}$ be defined by* (10)*. Let $\left\{\bar{\mathbf{w}}^k\right\} \subset \mathcal{W}$ and $\{(\gamma_k, \beta_k)\} \subset \mathbb{R}^2_{++}$ be the sequences that satisfy* (14)*. Then, it holds that:*

$$-D_{\mathcal{Y}}^\star \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \le f(\bar{\mathbf{x}}^k) - f^\star \le \gamma_k D_{\mathcal{X}}^{\mathbf{S}} \quad \text{and} \quad \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \le 2\beta_k D_{\mathcal{Y}}^\star + \sqrt{2\gamma_k\beta_k D_{\mathcal{X}}^{\mathbf{S}}}, \tag{15}$$

where $D_{\mathcal{Y}}^\star := \min\{\|\mathbf{y}^\star\| : \mathbf{y}^\star \in \mathcal{Y}^\star\}$, which is the norm of a minimum norm dual solutions.

Hence, we can derive algorithms based (γ_k, β_k) with a predictable convergence rate via (15). In the sequel, we manipulate τ_k and ψ_k to do just that in order to preserve (14) á la Nesterov [1]. Finally, we say that $\bar{\mathbf{x}}^k \in \mathcal{X}$ is an ε-solution of (1) if $|f(\bar{\mathbf{x}}^k) - f^\star| \le \varepsilon$ and $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \le \varepsilon$.

3.2. Initial points. We first show how to compute an initial point $\mathbf{w}^0$ such that $G_0(\bar{\mathbf{w}}^0) \le 0$.

Lemma 2 ([28])**.** *Given $\mathbf{x}^c \in \mathcal{X}$, $\bar{\mathbf{w}}^0 := [\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0] \in \mathcal{W}$ is computed by:*

$$\begin{cases} \bar{\mathbf{x}}^0 = \mathbf{x}_{\gamma_0}^\star(0^m) := \arg\min\limits_{\mathbf{x}\in\mathcal{X}} \{f(\mathbf{x}) + (\gamma_0/2)d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\mathbf{x}^c)\}, \\ \bar{\mathbf{y}}^0 = \mathbf{y}_{\beta_0}^\star(\bar{\mathbf{x}}^0) := \beta_0^{-1}(\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}). \end{cases} \tag{16}$$

satisfies $G_{\gamma_0\beta_0}(\bar{\mathbf{w}}^0) \le 0$ provided that $\beta_0\gamma_0 \ge \bar{L}^g$, where $\bar{L}^g$ is the Lipschitz constant of ∇g_γ with g_γ given by (13)*.*

3.3. An algorithmic template. Algorithm 1 combines the above ingredients for solving (1). We observe that the key computational step of Algorithm 1 is Step 3, where we update $[\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1}]$. In the algorithm, we provide two update schemes (1P2D) and (2P1D) based on the updates of the primal or dual variables. The primal step $\mathbf{x}_{\gamma_k}^\star(\bar{\mathbf{y}}^k)$ is calculated via (12). At line 3 of (2P1D), the operator $\mathrm{prox}_{\beta f}^{\mathbf{S}}$ is computed as:

$$\mathrm{prox}_{\beta f}^{\mathbf{S}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) := \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \left\{ f(\mathbf{x}) + \hat{\mathbf{y}}^T\mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) + \beta^{-1}d_b(\mathbf{S}\mathbf{x}, \mathbf{S}\hat{\mathbf{x}}) \right\}, \tag{17}$$

Algorithm 1: (*A primal-dual algorithmic template using model-based excessive gap*)

Inputs: Fix $\gamma_0 > 0$. Choose $c_0 \in (-1, 1]$.

Initialization:

1: Compute $a_0 := 0.5(1+c_0+\sqrt{4(1-c_0)+(1+c_0)^2}$, $\tau_0 := a_0^{-1}$, and $\beta_0 := \gamma_0^{-1}\bar{L}^g$ (c.f. the text).

2: Compute $[\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0]$ by (16) as in Lemma 2.

For $k = 0$ **to** $k_{\max}$, **perform the following steps:**

3: If **stopping_criterion** holds, then terminate. Otherwise, use one of the following schemes:

$$(\text{2P1D}): \left\{\begin{array}{ll} \hat{\mathbf{x}}^k & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_k}(\bar{\mathbf{y}}^k) \\ \hat{\mathbf{y}}^k & := \beta_{k+1}^{-1}(\mathbf{A}\hat{\mathbf{x}}^k - \mathbf{b}) \\ \bar{\mathbf{x}}^{k+1} & := \operatorname{prox}^{\mathbf{S}}_{\beta_{k+1}f}(\hat{\mathbf{x}}^k, \hat{\mathbf{y}}^k) \\ \bar{\mathbf{y}}^{k+1} & := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k \hat{\mathbf{y}}^k. \end{array}\right. \qquad (\text{1P2D}): \left\{\begin{array}{ll} \bar{\mathbf{y}}^\star_k & := \beta_k^{-1}(\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}), \\ \hat{\mathbf{y}}^k & := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k \bar{\mathbf{y}}^\star_k, \\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_k}(\hat{\mathbf{y}}^k), \\ \bar{\mathbf{y}}^{k+1} & := \hat{\mathbf{y}}^k + \gamma_k\big(\mathbf{A}\mathbf{x}^\star_{\gamma_k}(\hat{\mathbf{y}}^k) - \mathbf{b}\big). \end{array}\right.$$

4: Update $\beta_{k+1} := (1-\tau_k)\beta_k$ and $\gamma_{k+1} := (1-c_k\tau_k)\gamma_k$. Update c_{k+1} from c_k (optional).

5: Update $a_{k+1} := 0.5\big(1 + c_{k+1} + \sqrt{4a_k^2 + (1-c_{k+1})^2}\big)$ and set $\tau_{k+1} := a_{k+1}^{-1}$.

End For

where we overload the notation of the proximal operator prox defined by (3). At Step 2 of Algorithm 1, if we choose $\mathbf{S} := \mathbb{I}$, i.e., $d_b(\mathbf{Sx}, \mathbf{Sx}^c) := d_b(\mathbf{x}, \mathbf{x}^c)$ for $\mathbf{x}^c$ being the center point of b, then we set $\bar{L}^g := \|\mathbf{A}\|^2$. If $\mathbf{S} := \mathbf{A}$, i.e., $d_b(\mathbf{Sx}, \mathbf{Sx}^c) := (1/2)\|\mathbf{Ax} - \mathbf{b}\|^2$, then we set $\bar{L}^g := 1$.

Theorem 1 characterizes three variants of Algorithm 1, whose proof can be found in [28].

Theorem 1. *Let $\big\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)\big\}$ be the sequence generated by Algorithm 1 after k iterations. Then:*

a) *If $\mathbf{S} = \mathbf{A}$, i.e., using the augmented Lagrangian smoother, $\gamma_0 := \sqrt{\bar{L}^g} = 1$, and $c_k := 0$, then the* (1P2D) *update satisfies:*

$$(\text{1P2D}): \ -\frac{1}{2}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|^2 - D^\star_{\mathcal{Y}}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \le f(\bar{\mathbf{x}}^k) - f^\star \le 0 \ \ \textit{and} \ \ \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \le \frac{8D^\star_{\mathcal{Y}}}{(k+1)^2}.$$

Consequently, the worst-case complexity of Algorithm 1 to achieve an ε-solution $\bar{\mathbf{x}}^k$ is $\mathcal{O}(\varepsilon^{-1/2})$.

b) *If $\mathbf{S} = \mathbb{I}$, i.e., using the Bregman distance smoother, $\gamma_0 := \sqrt{\bar{L}^g} = \|\mathbf{A}\|$, and $c_k := 1$, then, for the* (2P1D) *scheme, we have:*

$$(\text{2P1D}): \ -D^\star_{\mathcal{Y}}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \le f(\bar{\mathbf{x}}^k) - f^\star \le \frac{\|\mathbf{A}\|}{k+1} D^{\mathbb{I}}_{\mathcal{X}} \ \ \textit{and} \ \ \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \le \frac{\|\mathbf{A}\|(2D^\star_{\mathcal{Y}} + \sqrt{2D^{\mathbb{I}}_{\mathcal{X}}})}{k+1}.$$

c) *Similarly, if $\gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|}{K+1}$ and $c_k := 0$ for all $k = 0, 1, \ldots, K$, then, for the* (1P2D) *scheme, we have:*

$$(\text{1P2D}): \ -D^\star_{\mathcal{Y}}\|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| \le f(\bar{\mathbf{x}}^K) - f^\star \le \frac{2\sqrt{2}\|\mathbf{A}\|}{(K+1)} D^{\mathbb{I}}_{\mathcal{X}} \ \ \textit{and} \ \ \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| \le \frac{2\sqrt{2}\|\mathbf{A}\|(D^\star_{\mathcal{Y}} + \sqrt{D^{\mathbb{I}}_{\mathcal{X}}})}{(K+1)}.$$

Hence, the worst-case complexity to achieve an ε-solution $\bar{\mathbf{x}}^k$ of (1) *in either* b) *or* c) *is $\mathcal{O}\left(\varepsilon^{-1}\right)$.*

The (1P2D) scheme has close relationship to some well-known primal dual methods we describe below. Unfortunately, (1P2D) has the drawback of fixing the total number of iterations *a priori*, which (2P1D) can avoid at the expense of one more proximal operator calculation at each iteration.

3.4. Impact of strong convexity. We can improve the above schemes when $f \in \mathcal{F}_\mu$, i.e., f is strongly convex with parameter $\mu_f > 0$. The dual function g given in (6) is smooth and Lipschitz gradient with $L^g_f := \mu_f^{-1}\|\mathbf{A}\|^2$. Let us illustrate this when $\mathbf{S} = \mathbb{I}$ and using the (1P2D) scheme as:

$$(\text{1P2D}_\mu): \left\{\begin{array}{ll} \hat{\mathbf{y}}^k & := (1-\tau_k)\bar{\mathbf{y}}^k + \tau_k\beta_k^{-1}(\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}), \\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star(\hat{\mathbf{y}}^k), \\ \bar{\mathbf{y}}^{k+1} & := \hat{\mathbf{y}}^k + \frac{1}{L^g_f}\big(\mathbf{A}\mathbf{x}^\star(\hat{\mathbf{y}}^k) - \mathbf{b}\big). \end{array}\right.$$

Here, $\mathbf{x}^\star(\hat{\mathbf{y}}^k)$ is the solution of (6) at $\hat{\mathbf{y}}^k$. We can still choose the starting point as in (16) with $\beta_0 := L^g_f$. The parameters β_k and τ_k at Steps 4 and 5 of Algorithm 1 are updated as $\beta_{k+1} := (1-\tau_k)\beta_k$,

and $\tau_{k+1} := \frac{\tau_k}{2}(\sqrt{\tau_k^2+4}-\tau_k)$, where $\beta_0 := L_f^g$ and $\tau_0 := (\sqrt{5}-1)/2$. The following corollary illustrates the convergence of Algorithm 1 using (1P2D_μ); see [28] for the detail proof.

Corollary 1. *Let $f \in \mathcal{F}_\mu$ and $\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k)\}_{k\geq 0}$ be generated by Algorithm 1 using (1P2D_μ). Then:*

$$-D_{\mathcal{Y}}^\star\|\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}\| \leq f(\bar{\mathbf{x}}^k)-f^\star \leq 0 \quad \textit{and} \quad \|\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}\| \leq \frac{4\|\mathbf{A}\|^2 D_{\mathcal{Y}}^\star}{\mu_f(k+2)^2}.$$

Moreover, we also have $\|\bar{\mathbf{x}}^k-\mathbf{x}^\star\| \leq \frac{4\|\mathbf{A}\|}{(k+2)\mu_f}D_{\mathcal{Y}}^\star$.

It is important to note that, when $f \in \mathcal{F}_\mu$, we only have one smoothness parameter β and, hence, we do not need to fix the number of iterations a priori (compared with [18]).

4 Algorithmic enhancements through existing methods

Our framework can be applied to develop other variants of popular primal-dual methods for (1) including alternating minimization algorithms and alternating direction methods of multipliers. We illustrate in this section three variants of Algorithm 1. We also borrow adaptation heuristics from other algorithms to enhance our practical performance.

4.1. Proximal-based decomposition method. We can choose $\mathbf{x}_c^k := \mathbf{x}_{\gamma_{k-1}}^\star(\hat{\mathbf{y}}^{k-1})$. This makes the (1P2D) scheme of Algorithm 1 similar to the proximal-based decomposition algorithm in [30], which employs the proximal term $d_b(\cdot, \hat{\mathbf{x}}_{k-1}^\star)$ with the Bregman distance d_b.

4.2. ADMM. Let f and $\mathcal{X}$ be 2-decomposable, i.e., $f(\mathbf{x}) := f_1(\mathbf{x}_1)+f_2(\mathbf{x}_2)$ and $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2$. We can apply the (1P2D) scheme of Algorithm 1 to this case with f_1 being $f_{\gamma,1}(\cdot) := f_1(\cdot) + \frac{\gamma}{2}\|\mathbf{A}_1(\cdot-\mathbf{x}_1^c)\|_2^2$ for fixed $\mathbf{x}_1^c \in \mathcal{X}_1$. For this variant, we substitute the primal step of computing $\mathbf{x}_\gamma^\star(\hat{\mathbf{y}}^k) = [\mathbf{x}_{\gamma,1}^\star(\hat{\mathbf{y}}^k), \mathbf{x}_{\gamma,2}^\star(\hat{\mathbf{y}}^k)]$ in (1P2D) by the following alternating step:

$$\begin{cases} \mathbf{x}_{\gamma,1}^\star(\hat{\mathbf{y}}^k) := \arg\min_{\mathbf{x}_1\in\mathcal{X}_1}\big\{f_1(\mathbf{x}_1)+(\hat{\mathbf{y}}^k)^T\mathbf{A}_1\mathbf{x}_1+\frac{\rho_k}{2}\|\mathbf{A}_1\mathbf{x}_1+\mathbf{A}_2\hat{\mathbf{x}}_2^k-\mathbf{b}\|^2+\frac{\gamma}{2}\|\mathbf{A}_1(\mathbf{x}_1-\mathbf{x}_1^c)\|_2^2\big\} \\ \mathbf{x}_{\gamma,2}^\star(\hat{\mathbf{y}}^k) := \arg\min_{\mathbf{x}_2\in\mathcal{X}_2}\big\{f_2(\mathbf{x}_2)+(\hat{\mathbf{y}}^k)^T\mathbf{A}_2\mathbf{x}_2+\frac{\eta_k}{2}\|\mathbf{A}_1\mathbf{x}_{\gamma,1}^\star(\hat{\mathbf{y}}^k)+\mathbf{A}_2\mathbf{x}_2-\mathbf{b}\|^2\big\}. \end{cases} \tag{18}$$

Here, ρ_k and η_k are two penalty parameters, and $\hat{\mathbf{x}}_2^k$ is the previous iteration of $\mathbf{x}_{\gamma,2}^\star(\hat{\mathbf{y}}^k)$. The update of parameters, as well as the complete algorithm and its convergence can be found in [29].

4.3. Primal-dual hybrid gradient (PDHG). When $\mathbf{A}_1$ and $\mathbf{A}_2$ are not orthogonal, one can linearize the quadratic terms in both steps of (18) to obtain a new preconditioned ADMM (PADMM) algorithm that employes the proximal operator of f_1 and f_2 instead of two general convex subproblems. In this case, the (1P2D) scheme with (18) leads to a new variant of PADMM in [8] or PDHG in [9]. Details of the complete algorithm can be found in [29].

4.4. Enhancements of our schemes. For the PADMM and ADMM methods, a great deal of adaptation techniques has been proposed to enhance their convergence. We can view some of these techniques in the light of model-based excessive gap condition. For instance, Algorithm 1 decreases the smoothed gap function $G_{\gamma_k\beta_k}$ as illustrated in Definition 1. The actual decrease is then given by $f(\bar{\mathbf{x}}^k)-f^\star \leq \gamma_k(D_{\mathcal{X}}^{\mathbf{S}}-\Psi_k/\gamma_k)$. In practice, $D_k := D_{\mathcal{X}}^{\mathbf{S}}-\Psi_k/\gamma_k$ can be dramatically smaller than $D_{\mathcal{X}}^{\mathbf{S}}$ in the early iterations. This implies that increasing γ_k can improve practical performance. Such a strategy indeed forms the basis of many adaptation techniques in PADMM and in ADMM.

Specifically, if γ_k increases, then τ_k also increases and β_k decreases. Since β_k measures the primal feasibility gap $\mathcal{F}_k := \|\mathbf{A}\bar{\mathbf{x}}^k-\mathbf{b}\|$ due to Lemma 1, we should only increase γ_k if the feasibility gap $\mathcal{F}_k$ is relatively high. Indeed, when $\mathbf{x}^c = \mathbf{x}_k^c$ is updated adaptively, we can compute the dual feasibility gap as $\mathcal{H}_k := \gamma_k\|\mathbf{A}_1^T\mathbf{A}_2((\hat{\mathbf{x}}_2^\star)_{k+1}-(\hat{\mathbf{x}}_2^\star)_k)\|$. Then, if $\mathcal{F}_k \geq s\mathcal{H}_k$ for some $s>0$, we increase $\gamma_{k+1} := c\gamma_k$ for some $c>1$ (we use $c_k = c := 1.05$ in practice). We can also decrease the parameter γ_k in (1P2D) by $\gamma_{k+1} := (1-c_k\tau_k)\gamma_k$, where $c_k := d_b(\mathbf{S}\mathbf{x}_{\gamma_k}^\star(\hat{\mathbf{y}}^k), \mathbf{S}\mathbf{x}_c)/D_{\mathcal{X}}^{\mathbf{S}} \in [0,1]$ after or during the update of $(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1})$ as in (2P1D) if we know the estimate $D_{\mathcal{X}}^{\mathbf{S}}$.

5 Numerical illustrations

5.1. Theoretical vs. practical bounds. We demonstrate the empirical performance of Algorithm 1 w.r.t. its theoretical bounds via a basic non-overlapping sparse-group basis pursuit problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n}\big\{\sum\nolimits_{i=1}^{n_g} w_i\|\mathbf{x}_{g_i}\|_2 : \mathbf{A}\mathbf{x}=\mathbf{b},\ \|\mathbf{x}\|_\infty \leq \rho\big\}, \tag{19}$$

where $\rho > 0$ is the signal magnitude, and g_i and w_i's are the group indices and weights, respectively.

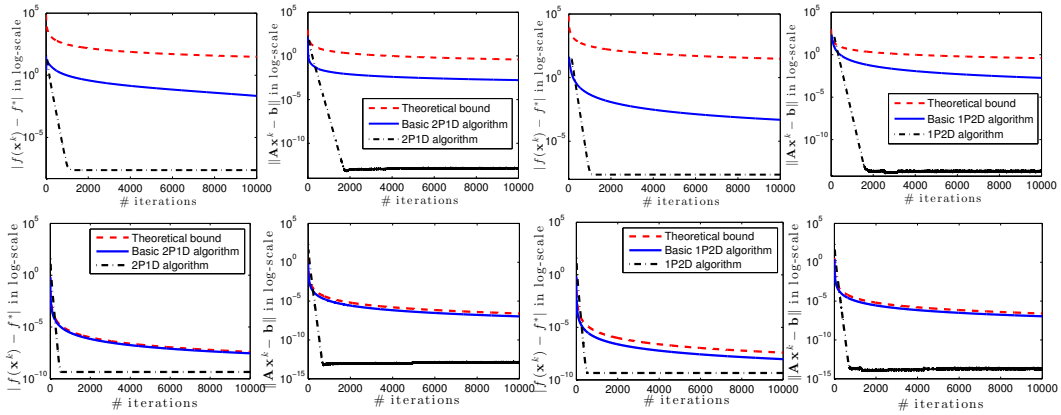

Figure 1: Actual performance vs. theoretical bounds: [top row] – the *decomposable* Bregman distance smoother ($\mathbf{S} = \mathbb{I}$), and [bottom row] – the augmented Lagrangian smoother ($\mathbf{S} = \mathbf{A}$).

In this test, we fix $\mathbf{x}^c = \mathbf{0}^n$ and $d_b(\mathbf{x}, \mathbf{x}^c) := (1/2)\|\mathbf{x}\|^2$. Since ρ is given, we can evaluate $D_{\mathcal{X}}$ numerically. By solving (19) with the SDPT3 interior-point solver [31] up to the accuracy 10^{-8}, we can numerically estimate $D_{\mathcal{Y}}^\star$ and $f^\star$. In the (2P1D) scheme, we set $\gamma_0 = \beta_0 = \sqrt{L_g}$, while, in the (1P2D) scheme, we set $\gamma_0 := 2\sqrt{2}\|\mathbf{A}\|(K+1)^{-1}$ with $K := 10^4$ and generate the theoretical bounds defined in Theorem 1.

We test the performance of the four variants using a synthetic data: $n = 1024$, $m = \lfloor n/3 \rfloor = 341$, $n_g = \lfloor n/8 \rfloor = 128$, and $\mathbf{x}^\natural$ is a $\lfloor n_g/8 \rfloor$-sparse vector. Matrix $\mathbf{A}$ is generated randomly using the iid standard Gaussian and $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural$. The group indices g_i is also generated randomly ($i = 1, \cdots, n_g$).

The empirical performance of two variants: (2P1D) and (1P2D) of Algorithm 1 is shown in Figure 1. The basic algorithm refers to the case when $\mathbf{x}_k^c := \mathbf{x}^c = \mathbf{0}^n$ and the parameters are not tuned. Hence, at each iteration of the basic (1P2D), it requires only 1 proximal calculation and applies $\mathbf{A}$ and $\mathbf{A}^T$ once each, and at each iteration of the basic (2P1D), we use 2 proximal calculations and apply $\mathbf{A}$ twice and $\mathbf{A}^T$ once. In contrast, (2P1D) and (1P2D) variants whose iterations require one more application of $\mathbf{A}^T$ for adaptive parameter updates.

As can be seen from Figure 1 (row 1) that the empirical performance of the basic variants roughly follows the $\mathcal{O}(1/k)$ convergence rate in terms of $|f(\bar{\mathbf{x}}^k) - f^\star|$ and $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|$. The deviations from the bound are due to the increasing sparsity of the iterates, which improves empirical convergence. With a kick-factor of $c_k = -0.02/\tau_k$ and adaptive $\mathbf{x}_k^c$, both turned variants (2P1D) and (1P2D) significantly outperform theoretical predictions. Indeed, they approach $\mathbf{x}^\star$ up to 10^{-13} accuracy, i.e., $\|\bar{\mathbf{x}}^k - \mathbf{x}^\star\| \leq 10^{-13}$ after a few hundreds of iterations.

Similarly, Figure 1 (row 2) illustrates the actual performance vs. the theoretical bounds $\mathcal{O}(1/k^2)$ by using the augmented Lagrangian smoother. Here, we solve the subproblems (13) and (17) by using FISTA [32] up to 10^{-8} accuracy as suggested in [28]. In this case, the theoretical bounds and the actual performance of the basis variants are very close to each other both in terms of $|f(\bar{\mathbf{x}}^k) - f^\star|$ and $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|$. When the parameter γ_k is updated, the algorithms exhibit a better performance.

5.2. Binary linear support vector machine. This example is concerned with the following binary linear support vector machine problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \big\{F(\mathbf{x}) := \sum\nolimits_{j=1}^m \ell_j(y_j, \mathbf{w}_j^T\mathbf{x} - \mathbf{b}_j) + g(\mathbf{x})\big\}, \tag{20}$$

where $\ell_j(\cdot,\cdot)$ is the Hinge loss function given by $\ell_j(s,\tau) := \max\{0, 1 - s\tau\} = [1 - s\tau]_+$, $\mathbf{w}_j$ is the column of a given matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^n$ is the intercept vector, $\mathbf{y} \in \{-1, +1\}^m$ is a classifier vector g is a given regularization function, e.g., $g(\mathbf{x}) := (\lambda/2)\|\mathbf{x}\|^2$ for the ℓ_2-regularizer or $g(\mathbf{x}) := \lambda\|\mathbf{x}\|_1$ for the ℓ_1-regularizer, where $\lambda > 0$ is a regularization parameter.

By introducing a slack variable $\mathbf{r} = \mathbf{W}\mathbf{x} - \mathbf{b}$, we can write (20) in terms of (1) as:

$$\min_{\mathbf{x}\in\mathbb{R}^n, \mathbf{r}\in\mathbb{R}^m} \Big\{ \sum\nolimits_{j=1}^m \ell_j(y_j, \mathbf{r}_j) + g(\mathbf{x}) : \mathbf{W}\mathbf{x} - \mathbf{r} = \mathbf{b}\Big\}. \tag{21}$$

Now, we apply the (1P2D) variant to solve (21). We test this algorithm on (21) and compare it with LibSVM [33] using two problems from the LibSVM data set available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. The first problem is `a1a`, which has $p = 119$ features and $N = 1605$ data points, while the second problem is `news20`, which has $p = 1'355'191$ features and $N = 19'996$ data points.

We compare Algorithm 1 and the LibSVM solver in terms of the final value $F(\mathbf{x}^k)$ of the original objective function F, the computational time, and the classification accuracy $\mathrm{ca}_\lambda := 1 - N^{-1}\sum_{j=1}^{N}\left[\mathrm{sign}(\mathbf{W}\mathbf{x}^k - \mathbf{r}) \neq \mathbf{y}\right]$ of both training and test data set. We randomly select 30% data in `a1a` and `news20` to form a test set, and the remaining 70% data is used for training. We perform 10 runs and compute the average results. These average results are plotted in Fig. 2 for two separate problems, respectively. The upper and lower bounds show the maximum and minimum values of these 10 runs.

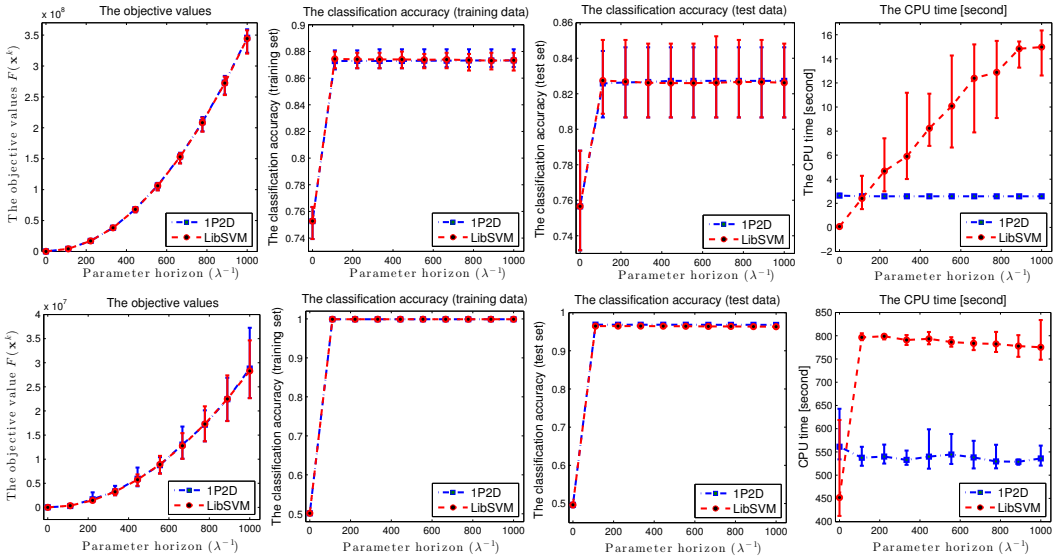

Figure 2: The average performance results of the two algorithms on the `a1a` (first row) and `news20` (second row) problems.

As can be seen from these results that both solvers give relatively the same objective values, the accuracy for these two problems, while the computational of (1P2D) is much lower than `LibSVM`. We note that `LibSVM` becomes slower when the parameter λ becomes smaller due to its active-set strategy. The (1P2D) algorithm is almost independent of the regularization parameter λ, which is different from active-set methods. In addition, the performance of (1P2D) can be improved by taking account its parallelization ability, which has not fully been exploited yet in our implementation.

6 Conclusions

We propose a model-based excessive gap (MEG) technique for constructing and analyzing first-order primal-dual methods that numerically approximate an optimal solution of the constrained convex optimization problem (1). Thanks to a combination of smoothing strategies and MEG, we propose, to the best of our knowledge, the first primal-dual algorithmic schemes for (1) that theoretically obtain optimal convergence rates directly without averaging the iterates and that seamlessly handle the p-decomposability structure. In addition, our analysis techniques can be simply adapt to handle inexact oracle produced by solving approximately the primal subproblems (c.f. [28]), which is important for the augmented Lagrangian versions with lower-iteration counts. We expect a deeper understanding of MEG and different smoothing strategies to help us in tailoring adaptive update strategies for our schemes (as well as several other connected and well-known schemes) in order to further improve the empirical performance.

Acknowledgments. This work is supported in part by the European Commission under the grants MIRG-268398 and ERC Future Proof, and by the Swiss Science Foundation under the grants SNF 200021-132548, SNF 200021-146750 and SNF CRSII2-147633.

References

[1] Y. Nesterov, "Excessive gap technique in nonsmooth convex minimization," *SIAM J. Optim.*, vol. 16, no. 1, pp. 235–249, 2005.

[2] D. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.

[3] V. Chandrasekaran, B. Recht, P. Parrilo, and A. Willsky, "The convex geometry of linear inverse problems," Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Tech. Report., 2012.

[4] M. B. McCoy, V. Cevher, Q. Tran-Dinh, A. Asaei, and L. Baldassarre, "Convexity in source separation: Models, geometry, and algorithms," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 87–95, 2014.

[5] M. J. Wainwright, "Structured regularizers for high-dimensional problems: Statistical and computational issues," *Annual Review of Statistics and its Applications*, vol. 1, pp. 233–253, 2014.

[6] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[7] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, pp. 1168–1200, 2005.

[8] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[9] T. Goldstein, E. Esser, and R. Baraniuk, "Adaptive primal-dual hybrid gradient methods for saddle point problems," *Tech. Report.*, vol. http://arxiv.org/pdf/1305.0546v1.pdf, pp. 1–26, 2013.

[10] B. He and X. Yuan, "On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers," Numer. Math., DOI 10.1007/s00211-014-0673-6, 2014.

[11] ——, "On the $\mathcal{O}(1/n)$ convergence rate of the Douglas-Rachford alternating direction method," *SIAM J. Numer. Anal.*, vol. 50, pp. 700–709, 2012.

[12] Y. Ouyang, Y. Chen, G. L. Lan., and E. J. Pasiliao, "An accelerated linearized alternating direction method of multiplier," *Tech*, 2014.

[13] R. Shefi and M. Teboulle, "Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization," *SIAM J. Optim.*, vol. 24, no. 1, pp. 269–297, 2014.

[14] H. Wang and A. Banerjee, "Bregman alternating direction method of multipliers," Tech. Report, pp. 1–18, 2013. Online at: *http://arxiv.org/pdf/1306.3203v1.pdf*.

[15] H. Ouyang, N. He, L. Q. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," *JMLR W&CP*, vol. 28, pp. 80–88, 2013.

[16] T. Goldstein, B. O. Donoghue, and S. Setzer, "Fast alternating direction optimization methods," SIAM J. Imaging Sci., vol. 7, no. 3, pp. 1588–1623, 2014.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[18] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Oper. Res. Letter*, vol. 42, no. 1, pp. 1–6, 2014.

[19] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Rice University CAAM, Tech. Rep., 2012, tR12-14.

[20] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Athena Scientific, 1996.

[21] R. T. Rockafellar, "Augmented lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, pp. 97–116, 1976.

[22] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.

[23] G. Lan and R. Monteiro, "Iteration-complexity of first-order augmented Lagrangian methods for convex programming," *Math. Program.*, DOI 10.1007/s10107-015-0861-x, 2015.

[24] V. Nedelcu, I. Necoara, and Q. Tran-Dinh, "Computational complexity of inexact gradient augmented Lagrangian methods: Application to constrained MPC," *SIAM J. Optim. Control*, vol. 52, no. 5, pp. 3109–3134, 2014.

[25] Y. Nesterov, *Introductory lectures on convex optimization: a basic course*, Kluwer Academic Publishers, 2004, vol. 87.

[26] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*, N. York, Ed. Springer-Verlag, 2003, vol. 1-2.

[27] A. Auslender, *Optimisation: Méthodes Numériques*. Paris: Masson, 1976.

[28] Q. Tran-Dinh and V. Cevher, "A primal-dual algorithmic framework for constrained convex minimization," *Tech. Report., LIONS*, pp. 1–54, 2014.

[29] Q. Tran-Dinh and V. Cevher, "Optimal-rate and tuning-free alternating algorithms for constrained convex optimization," *Tech. Report., LIONS*, 2015.

[30] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Math. Program.*, vol. 64, pp. 81–101, 1994.

[31] K.-C. Toh, M. Todd, and R. Tütüncü, "On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0," *NUS Singapore, Tech. Report*, 2010.

[32] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[33] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 27, pp. 1–27, 2011.