A Conditional Gradient Framework for Composite Convex Minimization with Applications to Semidefinite Programming

Alp Yurtsever [1] **Olivier Fercoq** [2] **Francesco Locatello** [3 4] **Volkan Cevher** [1]

Abstract

We propose a conditional gradient framework for a composite convex minimization template with broad applications. Our approach combines smoothing and homotopy techniques under the CGM framework, and provably achieves the optimal $\mathcal{O}(1/\sqrt{k})$ convergence rate. We demonstrate that the same rate holds if the linear subproblems are solved approximately with additive or multiplicative error. In contrast with the relevant work, we are able to characterize the convergence when the non-smooth term is an indicator function. Specific applications of our framework include the non-smooth minimization, semidefinite programming, and minimization with linear inclusion constraints over a compact domain. Numerical evidence demonstrates the benefits of our framework.

1. Introduction

The importance of convex optimization in machine learning has increased dramatically in the last decade due to the new theory in structured sparsity, rank minimization and statistical learning models like support vector machines. Indeed, a large class of learning formulations can be addressed by the following composite convex minimization template:

$$\min_{x\in\mathcal{X}} F(x) := f(x) + g(Ax), \tag{1.1}$$

where $\mathcal{X} \subset \mathbb{R}^n$ is compact (nonempty, bounded, closed) and its 0-dimensional faces (*i.e.*, its vertices) are called *atoms*. $f : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ is a smooth proper closed convex function, $A \in \mathbb{R}^{d\times n}$, and $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper closed convex function which is possibly non-smooth.

[1]LIONS, Ecole Polytechnique Fédérale de Lausanne, Switzerland [2]LTCI, Télécom ParisTech, Université Paris-Saclay, France [3]BMI, Dept. of Computer Science, ETH Zurich, Switzerland [4]Empirical Inference, Max Planck Institute for Intelligent Systems, Germany. Correspondence to: Alp Yurtsever <alp.yurtsever@epfl.ch>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

By using the powerful proximal gradient framework, problems belonging to the template (1.1) can be solved nearly as efficiently as if they were fully smooth with fast convergence rates. By proximal (prox) operator, we mean:

$$\mathrm{prox}_{\varphi}(v) = \arg\min_{x}\ \varphi(x) + \frac{1}{2}\|x - v\|^2.$$

These methods make use of the gradient of the smooth function f along with the prox of the non-smooth part $g(A\,\cdot) + \iota_{\mathcal{X}}$, where $\iota_{\mathcal{X}}$ denotes the indicator function of $\mathcal{X}$, and are optimal as they match the iteration lower-bounds.

Surprisingly, the proximal operator can impose an undesirable computational burden and even intractability on these gradient-based methods, such as the computation of a full singular value decomposition in the ambient dimension or the computation of proximal mapping for the latent group lasso (Jaggi, 2013). Moreover, the linear mapping A often complicates the computation of the prox itself, and require more sophisticated splitting or primal-dual methods.

As a result, the conditional gradient method (CGM, aka Frank-Wolfe method) has recently increased in popularity, since it requires only a linear minimization oracle (lmo). By lmo, we mean a resolvent of the following problem:

$$\mathrm{lmo}_{\mathcal{X}}(v) = \arg\min_{x\in\mathcal{X}} \langle x,\, v\rangle.$$

CGM features significantly reduced computational costs (*e.g.*, when $\mathcal{X}$ is the spectrahedron), tractability (*e.g.*, when $\mathcal{X}$ is a latent group lasso norm), and interpretability (*e.g.*, they generate solutions as a combination of atoms of $\mathcal{X}$). The method is shown in Algorithm 1 when $g(Ax) = 0$:

Algorithm 1 CGM for smooth minimization

Input: $x_1 \in \mathcal{X}$
for $k = 1, 2, \ldots,$ **do**
  $\eta_k = 2/(k+1)$
  $s_k = \arg\min_{x\in\mathcal{X}} \langle \nabla f(x_k),\, x\rangle$
  $x_{k+1} = x_k + \eta_k(s_k - x_k)$
end for

The method itself is optimal for this particular template (Lan, 2014). Unfortunately, CGM *provably* cannot handle the non-smooth term $g(Ax)$ in (1.1) via its subgradients (*cf.* Section 5.3 for a counter example by Nesterov (2017)).

When the non-smooth part is an indicator function, one could take the intersection between $\mathcal{X}$ and the set represented by g. Unfortunately, even the lmo itself can be a difficult optimization problem depending on the structure of the domain. On many domains of interest, that we can parametrize as a composition of simple sets, linear problems are infeasible (Richard et al., 2012; Yen et al., 2016).

In this paper, we propose a CGM framework for solving the composite problem (1.1) with rigorous convergence guarantees. Our approach retains the simplicity of projection free methods, but allows to disentangle the complexity of the feasible set in order to preserve the simplicity of the lmo.

Our method combines the ideas of smoothing (Nesterov, 2005) and homotopy under the CGM framework. Lan (2014) proposes a similar approach for non-smooth problems, which is extended for the conditional gradient sliding framework in (Lan & Zhou, 2016; Lan et al., 2017). Their analysis, however, is restricted by the Lipschitz continuity assumption. Consequently, it does not apply to the standard semidefinite programming template, or to the problems with affine inclusion constraints, limiting its applicability in machine learning (*cf.* Sections 5.5 and 5.6).

Our work covers in particular the case where non-smooth part is the indicator function of a convex set. Similar ideas can be found for the primal-dual subgradient method and the coordinate descent in (Tran-Dinh et al., 2018; Alacaoglu et al., 2017) via the projection onto $\mathcal{X}$.

Our contributions can be summarized as follows:

- ▷ We introduce a simple, easy to implement CGM framework for solving composite problem (1.1), and prove that it achieves the optimal $\mathcal{O}(1/\sqrt{k})$ rate.
- ▷ We prove $\mathcal{O}(1/\sqrt{k})$ rate both in the objective residual and the feasibility gap, when the non-smooth term is an indicator function.
- ▷ We analyze the convergence of our algorithm under inexact oracles with additive and multiplicative errors.
- ▷ We present key instances of our framework, including the non-smooth minimization, minimization with linear inclusion constraints, and convex splitting.
- ▷ We present empirical evidence supporting our findings.

Roadmap. Section 2 recalls some basic notions and presents the preliminaries about the smoothing technique. In Section 3, we present CGM for composite convex minimization along with the convergence guarantees, and we extend these results for inexact oracle calls in Section 4. We describe some important special applications of our framework in Section 5. We provide empirical evidence supporting our theoretical findings in Section 6. Finally, Section 7 draws the conclusions with a discussion on the future work. Proofs and technical details are deferred to the appendix.

2. Notation & Preliminaries

We use $\|\cdot\|$ to denote the Euclidean norm for vectors and the spectral norm (a.k.a. Schatten ∞-norm) for linear mappings. We denote the Frobenius norm by $\|\cdot\|_F$, and the nuclear norm (a.k.a. Schatten 1-norm or trace norm) by $\|\cdot\|_{S_1}$. The notation $\langle\cdot,\,\cdot\rangle$ refers the Euclidean or Frobenius inner product. The symbol $^\top$ denotes the adjoint of a linear map, and the symbol $\succcurlyeq$ denotes the semidefinite order. We denote the diameter of $\mathcal{X}$ by $D_{\mathcal{X}} = \max_{x_1,x_2\in\mathcal{X}} \|x_1 - x_2\|$, and the distance between a point y and a set $\mathcal{K}$ by $\mathrm{dist}(y,\mathcal{K})$.

Lipschitz continuity. We say that a function $g:\mathbb{R}^d\to\mathbb{R}$ is L-Lipschitz continuous if it satisfies

$$|g(x_1) - g(x_2)| \le L\|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathbb{R}^d.$$

Smoothness. A differentiable function $f:\mathcal{X}\to\mathbb{R}$ is L_f-smooth if the gradient ∇f is L_f-Lipschitz continuous:

$$\|\nabla f(x_1) - \nabla f(x_2)\| \le L_f\|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{X}.$$

Fenchel conjugate & Smoothing. We consider the smooth approximation of a non-smooth term g obtained using the technique described by Nesterov (2005) with the standard Euclidean proximity function $\frac{1}{2}\|\cdot\|^2$ and a smoothness parameter $\beta > 0$

$$g_\beta(z) = \max_{y\in\mathbb{R}^d} \langle z,\, y\rangle - g^*(y) - \frac{\beta}{2}\|y\|^2,$$

where g^* denotes the Fenchel conjugate of g

$$g^*(x) = \sup_v \langle x,\, v\rangle - g(v).$$

Note that g_β is convex and $\frac{1}{\beta}$-smooth.

Solution set. We denote an exact solution of (1.1) by $x^\star$, and the set of all solutions by $\mathcal{X}^\star$. Throughout the paper, we assume that the solution set $\mathcal{X}^\star$ is nonempty.

Given an accuracy level $\epsilon > 0$, we call a point $x\in\mathcal{X}$ as an ϵ-solution of (1.1) if

$$f(x) + g(Ax) - f^\star - g^\star \le \epsilon, \tag{2.1}$$

where we use the notation $f^\star = f(x^\star)$ and $g^\star = g(Ax^\star)$.

When g is the indicator function of a set $\mathcal{K}$, condition (2.1) is not well-defined for infeasible points. Hence, we refine our definition, and call a point $x\in\mathcal{X}$ as an ϵ-solution if

$$f(x) - f^\star \le \epsilon, \quad \text{and} \quad \mathrm{dist}(Ax,\mathcal{K}) \le \epsilon.$$

Here, we call $f(x) - f^\star$ as the objective residual and $\mathrm{dist}(Ax,\mathcal{K})$ as the feasibility gap. We use the same ϵ for the objective residual and the feasibility gap, since the distinct choices can be handled by scaling f.

Lagrange saddle point. Suppose that g is the indicator function of a convex set $\mathcal{K}$, and denote the Lagrangian of problem (1.1) by $\mathcal{L}(x, y)$:

$$\mathcal{L}(x, y) := f(x) + \langle y,\, Ax\rangle - g^*(y).$$

We can formulate primal and dual problems as follows:

$$\underbrace{\max_{y\in\mathbb{R}^d} \min_{x\in\mathcal{X}} \mathcal{L}(x, y)}_{\text{dual}} \leq \underbrace{\min_{x\in\mathcal{X}} \max_{y\in\mathbb{R}^d} \mathcal{L}(x, y)}_{\text{primal}}.$$

We assume that the strong duality holds, *i.e.*, the relation above holds with equality. Slater's condition is a sufficient condition for strong duality. By Slater's condition, we mean

$$\operatorname{relint}(\mathcal{X} \times \mathcal{K}) \cap \{(x, r) \in \mathbb{R}^n \times \mathbb{R}^d : Ax = r\} \neq \emptyset,$$

where relint stands for the relative interior.

Throughout, $y^\star$ denotes a solution of the dual problem.

3. Algorithm & Convergence

Our method is based on the simple idea of combining smoothing and homotopy. In our problem template, the objective function F is non-smooth. We define the smooth approximation of F with smoothness parameter $\beta > 0$ as

$$F_\beta(x) = f(x) + g_\beta(Ax).$$

Note that F_β is $(L_f + \|A\|^2/\beta)$-smooth.

The algorithm takes a conditional gradient step with respect to the smooth approximation F_{β_k} at iteration k, where β_k is gradually decreased towards 0. Let us denote by $y^*_{\beta_k}$

$$\begin{aligned} y^*_{\beta_k}(Ax) &= \arg\max_{y\in\mathbb{R}^d} \langle Ax,\, y\rangle - g^*(y) - \frac{\beta_k}{2}\|y\|^2 \\ &= \operatorname{prox}_{\beta_k^{-1} g^*}(\beta_k^{-1} Ax) \\ &= \frac{1}{\beta_k}\big(Ax - \operatorname{prox}_{\beta_k g}(Ax)\big), \end{aligned}$$

where the last equality is due to the Moreau decomposition. Then, we can compute the gradient of F_{β_k} as

$$\begin{aligned} \nabla F_{\beta_k}(x) &= \nabla f(x) + A^\top y^*_{\beta_k}(Ax) \\ &= \nabla f(x) + \frac{1}{\beta_k} A^\top \big(Ax - \operatorname{prox}_{\beta_k g}(Ax)\big). \end{aligned}$$

Based on this formulation, we present our CGM framework for composite convex minimization template (1.1) in Algorithm 2. The choice of β_k comes from the convergence analysis, which can be found in the supplements.

Theorem 3.1. *The sequence x_k generated by Algorithm 2 satisfies the following bound:*

$$F_{\beta_k}(x_{k+1}) - F^\star \leq 2D_{\mathcal{X}}^2 \left(\frac{L_f}{k+1} + \frac{\|A\|^2}{\beta_0\sqrt{k+1}}\right).$$

Algorithm 2 CGM for composite problems

Input: $x_1 \in \mathcal{X}$, $\beta_0 > 0$
for $k = 1, 2, \ldots,$ **do**
  $\eta_k = 2/(k+1)$, and $\beta_k = \beta_0/\sqrt{k+1}$
  $v_k = \beta_k \nabla f(x_k) + A^\top\big(Ax_k - \operatorname{prox}_{\beta_k g}(Ax_k)\big)$
  $s_k = \arg\min_{x\in\mathcal{X}} \langle v_k,\, x\rangle$
  $x_{k+1} = x_k + \eta_k(s_k - x_k)$
end for

Theorem 3.1 does not directly certify the convergence of x_k to a solution, since the bound is on the smoothed gap $F_{\beta_k}(x_k) - F^\star$. To relate F_{β_k} back to F, one usually assumes Lipschitz continuity. This well known perspective leads us to Theorem 3.2, which is a direct extension of Theorem 4 of (Lan, 2014) for composite functions.

Theorem 3.2. *Assume that $g : \mathbb{R}^d \to \mathbb{R}$ is L_g-Lipschitz continuous. Then, the sequence x_k generated by Algorithm 2 satisfies the following convergence bound:*

$$F(x_k) - F^\star \leq 2D_{\mathcal{X}}^2 \left(\frac{L_f}{k} + \frac{\|A\|^2}{\beta_0\sqrt{k}}\right) + \frac{\beta_0 L_g^2}{2\sqrt{k}}.$$

Furthermore, if the constants $D_{\mathcal{X}}$, $\|A\|$ and L_g are known or easy to approximate, we can choose $\beta_0 = 2D_{\mathcal{X}}\|A\|/L_g$ to get the following convergence rate:

$$F(x_k) - F^\star \leq \frac{2D_{\mathcal{X}}^2 L_f}{k} + \frac{2D_{\mathcal{X}}\|A\|L_g}{\sqrt{k}}.$$

Lipschitz continuity assumption in Theorem 3.2 leaves many important applications out (*cf.* Sections 5.5 and 5.6). In Theorem 3.3, we take a step further and characterize the convergence when the non-smooth part is an indicator function. Note that x_k is not guaranteed to be a feasible point, since the condition $Ax_k \in \mathcal{K}$ is not guaranteed, but it converges to the feasible set, and it becomes feasible asymptotically.

Theorem 3.3. *Assume that $g : \mathbb{R}^d \to \mathbb{R}$ is the indicator function of a simple convex set $\mathcal{K}$. Then, the sequence x_k generated by Algorithm 2 satisfies:*

$$\begin{aligned} f(x_k) - f^\star &\geq -\|y^\star\| \operatorname{dist}(Ax_k, \mathcal{K}) \\ f(x_k) - f^\star &\leq 2D_{\mathcal{X}}^2 \left(\frac{L_f}{k} + \frac{\|A\|^2}{\beta_0\sqrt{k}}\right) \\ \operatorname{dist}(Ax_k, \mathcal{K}) &\leq \frac{2\beta_0}{\sqrt{k}} \left(\|y^\star\| + D_{\mathcal{X}}\sqrt{\frac{C_0}{\beta_0}}\right) \end{aligned}$$

where $C_0 = L_f + \|A\|^2/\beta_0$.

Remark 3.4. Similar to the classical CGM, we can consider variants of Algorithm 2 with line-search (which replaces the step size by $\eta_k = \min_{\eta\in[0,1]} F_{\beta_k}(x_k + \eta(s_k - x_k))$), and fully corrective updates (which replaces the last step by $x_{k+1} = \arg\min_{x\in\operatorname{conv}(s_1,\ldots,s_k)} F_{\beta_k}(x)$). All results presented in this paper remain valid for these variants.

4. Convergence with Inexact Oracles

Finding an exact solution of the lmo can be expensive in practice, especially when it involves a matrix factorization as in the SDP examples. On the other hand, approximate solutions can be much more efficient.

Different notions of inexact lmo are already explored in CGM and greedy optimization frameworks, *cf.*(Lacoste-Julien et al., 2013; Locatello et al., 2017a;b). We revisit the notions of additive and multiplicative errors which we adapt here for our setting.

4.1. Inexact Oracle with Additive Error

At iteration k, for the given direction v_k, we assume that the approximate lmo returns an element $\tilde{s}_k \in \mathcal{X}$ such that:

$$\langle v_k,\, \tilde{s}_k \rangle \leq \langle v_k,\, s_k \rangle + \delta \frac{\eta_k}{2} D_{\mathcal{X}}^2 \left(L_f + \frac{\|A\|^2}{\beta_k} \right) \tag{4.1}$$

for some $\delta > 0$, where s_k denotes the exact solution of the lmo. Note that as in (Jaggi, 2013), we require the accuracy of lmo to increase as the algorithm progresses.

Replacing the exact lmo with the approximate oracles of the form (4.1) in Algorithm 2, we get the convergence guarantees in Theorems 4.1 to 4.3.

Theorem 4.1. *The sequence x_k generated by Algorithm 2 with approximate* lmo *(4.1) satisfies:*

$$F_{\beta_k}(x_{k+1}) - F^\star \leq 2 D_{\mathcal{X}}^2 \left(\frac{L_f}{k+1} + \frac{\|A\|^2}{\beta_0 \sqrt{k+1}} \right) (1+\delta).$$

Theorem 4.2. *Assume that g is L_g-Lipschitz continuous. Then, the sequence x_k generated by Algorithm 2 with approximate* lmo *(4.1) satisfies:*

$$F(x_k) - F^\star \leq 2 D_{\mathcal{X}}^2 \left(\frac{L_f}{k} + \frac{\|A\|^2}{\beta_0 \sqrt{k}} \right) (1+\delta) + \frac{\beta_0 L_g^2}{2\sqrt{k}}.$$

We can optimize β_0 from this bound if δ is known.

Theorem 4.3. *Assume that g is the indicator function of a simple convex set $\mathcal{K}$. Then, the sequence x_k generated by Algorithm 2 with approximate* lmo *(4.1) satisfies:*

$$\begin{aligned}
f(x_k) - f^\star &\geq -\|y^\star\| \operatorname{dist}(Ax_k, \mathcal{K}) \\
f(x_k) - f^\star &\leq 2 D_{\mathcal{X}}^2 \left(\frac{L_f}{k} + \frac{\|A\|^2}{\beta_0 \sqrt{k}} \right) (1+\delta) \\
\operatorname{dist}(Ax_k, \mathcal{K}) &\leq \frac{2\beta_0}{\sqrt{k}} \left(\|y^\star\| + D_{\mathcal{X}} \sqrt{\frac{C_0}{\beta_0}(1+\delta)} \right)
\end{aligned}$$

where $C_0 = L_f + \|A\|^2/\beta_0$.

4.2. Inexact Oracle with Multiplicative Error

We consider the multiplicative inexact oracle:

$$\langle v_k,\, \tilde{s}_k - x_k \rangle \leq \delta \langle v_k,\, s_k - x_k \rangle \tag{4.2}$$

where $\delta \in (0, 1]$. Replacing the exact lmo with the approximate oracles of the form (4.2) in Algorithm 2, we get the convergence guarantees in Theorems 4.4 to 4.6.

Theorem 4.4. *The sequence x_k generated by Algorithm 2 with approximate* lmo *of the form (4.2), and modifying $\eta_k = \frac{2}{\delta(k-1)+2}$ and $\beta_k = \frac{\beta_0}{\sqrt{\delta k+1}}$ satisfies:*

$$F_{\beta_k}(x_{k+1}) - F^\star \leq \frac{2}{\delta} \left(\frac{D_{\mathcal{X}}^2 L_f + \delta \mathcal{E}}{\delta k + 2} + \frac{D_{\mathcal{X}}^2 \|A\|^2}{\beta_0 \sqrt{\delta k + 2}} \right)$$

where $\mathcal{E} = F(x_1) - F^\star$.

Theorem 4.5. *Assume that g is L_g-Lipschitz continuous. Then, the sequence x_k generated by Algorithm 2 with approximate* lmo *(4.2), and modifying $\eta_k = \frac{2}{\delta(k-1)+2}$ and $\beta_k = \frac{\beta_0}{\sqrt{\delta k+1}}$ satisfies:*

$$F(x_k) - F^\star \leq \frac{2}{\delta} \left(\frac{D_{\mathcal{X}}^2 L_f + \delta \mathcal{E}}{\delta k + 1} + \frac{D_{\mathcal{X}}^2 \|A\|^2}{\beta_0 \sqrt{\delta k + 1}} \right) + \frac{\beta_0 L_g^2}{2\sqrt{\delta k + 1}},$$

where $\mathcal{E} = F(x_1) - F^\star$. We can optimize β_0 from this bound if δ is known.

Theorem 4.6. *Assume that g is the indicator function of a simple convex set $\mathcal{K}$. Then, the sequence x_k generated by Algorithm 2 with approximate* lmo *(4.2), and modifying $\eta_k = \frac{2}{\delta(k-1)+2}$ and $\beta_k = \frac{\beta_0}{\sqrt{\delta k+1}}$ satisfies:*

$$\begin{aligned}
f(x_k) - f^\star &\geq -\|y^\star\| \operatorname{dist}(Ax_k, \mathcal{K}) \\
f(x_k) - f^\star &\leq \frac{2}{\delta} \left(\frac{D_{\mathcal{X}}^2 L_f + \delta \mathcal{E}}{\delta k + 1} + \frac{D_{\mathcal{X}}^2 \|A\|^2}{\beta_0 \sqrt{\delta k + 1}} \right) \\
\operatorname{dist}(Ax_k, \mathcal{K}) &\leq \frac{2\beta_0}{\sqrt{\delta k + 1}} \left(\|y^\star\| + \sqrt{\frac{D_{\mathcal{X}}^2 C_0 + \delta \mathcal{E}}{\beta_0 \delta}} \right)
\end{aligned}$$

where $\mathcal{E} = F(x_1) - F^\star$ and $C_0 = L_f + \|A\|^2/\beta_0$.

5. Applications & Related Work

CGM is proposed for the first time in the seminal work of Frank & Wolfe (1956) for solving smooth convex optimization on a polytope. It is then progressively generalized for more general settings in (Levitin & Polyak, 1966; Dunn & Harshbarger, 1978; Dunn, 1979; 1980). Nevertheless, with the introduction of the fast gradient methods with $\mathcal{O}(1/k^2)$ rate by Nesterov (1987), the development of CGM-type methods entered into a stagnation period.

The recent developments in machine learning applications with vast data brought the scalability of the first order methods under scrutiny. As a result, there has been a renewed interest in CGM in the last decade. We compare our framework with the recent developments of CGM literature in different camps of problem templates below.

5.1. Smooth Problems

CGM is extended for the smooth convex minimization over the simplex by Clarkson (2010), for the spactrahedron by Hazan (2008), and for an arbitrary compact convex set by Jaggi (2013). Online, stochastic and block coordinate variants of CGM are introduced by Hazan & Kale (2012), Hazan & Luo (2016) and Lacoste-Julien et al. (2013) respectively.

When applied to smooth problems, Algorithm 2 is equivalent to the classical CGM, and Theorem 3.2 recovers the known optimal $\mathcal{O}(1/k)$ convergence rate. We refer to (Jaggi, 2013) for a review of applications of this template.

It needs to be mentioned that Nesterov (2017) relaxes the smoothness assumption showing that CGM converges for weakly-smooth objectives (*i.e.*, with Hölder continuous gradients of order $\nu \in (0, 1]$).

5.2. Regularized Problems

CGM for composite problems is considered recently by Nesterov (2017) and Xu (2017). A similar but slightly different template, where $\mathcal{X}$ and g are assumed to be a closed convex cone and a norm respectively, is also studied by Harchaoui et al. (2015). However, these works are based on the resolvents of a modified oracle,

$$\arg\min_{x\in\mathcal{X}} \langle x,\, v\rangle + g(Ax),$$

which can be expensive, unless $\mathcal{X} \equiv \mathbb{R}^n$, or $g = 0$.

Algorithm 2 applies to the problem template (1.1) by leveraging prox of the regularizer and lmo of the domain independently. This allows us to consider additional sparsity, group sparsity and structured sparsity promoting regularizations, elastic-net regularization, total variation regularization and many others under the CGM framework.

Semi-proximal mirror-prox proposed by He & Harchaoui (2015) is also based on the smoothing technique, yet the motivation is fundamentally different. This method considers the regularizers for which the prox is difficult to compute, but can be approximated via CGM.

5.3. Non-Smooth Problems

Template (1.1) covers the non-smooth convex minimization template as a special case:

$$\min_{x\in\mathcal{X}} g(Ax). \tag{5.1}$$

Unfortunately, the classical CGM (Algorithm 1) cannot handle the non-smooth minimization, as shown by Nesterov (2017) with the following counter-example.

Example. Let $\mathcal{X}$ be the unit Euclidean norm ball in $\mathbb{R}^2$, and $g(x) = \max\{x_{(1)}, x_{(2)}\}$. Clearly, $x^\star = [-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]^\top$. Choose an initial point $x_0 \neq x^\star$. We can use an oracle that returns a subgradient $\nabla f(x) \in \{\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\}$ at any point $x \in \mathcal{X}$. Therefore, lmo returns $\left[\begin{smallmatrix}-1\\0\end{smallmatrix}\right]$ or $\left[\begin{smallmatrix}0\\-1\end{smallmatrix}\right]$ at each iteration, and x_k belongs to the convex hull of $\{x_0, \left[\begin{smallmatrix}-1\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}0\\-1\end{smallmatrix}\right]\}$ which does not contain the solution.

Our framework escapes such issues by leveraging prox of the objective function g (see appendix for numerical illustration). In this pathological example, prox_g corresponds to the projection onto the simplex. Often times the cost of prox_g is negligible in comparison to the cost of $\mathrm{lmo}_{\mathcal{X}}$ (*cf.* Section 6.2 for a robust PCA example).

Assume that $g : \mathbb{R}^d \to \mathbb{R}$ is L_g-Lipschitz continuous. As a consequence of Theorem 3.2, Algorithm 2 for solving (5.1) by choosing $\beta_0 = 2D_{\mathcal{X}}\|A\|/L_g$ satisfies

$$g(Ax_k) - g^\star \leq \frac{2D_{\mathcal{X}}\|A\|L_g}{\sqrt{k}}.$$

We recover the method proposed by Lan (2014) in this specific setting. Lan (2014) shows that this rate is optimal for algorithms approximating the solution of (5.1) as a convex combination of lmo outputs.

We extend the analysis in this setting for inexact oracles. In contrast to the smooth case, where the additive error should decrease by $\mathcal{O}(1/k)$ rate, definition (4.1) implies that we can preserve the convergence rate in the non-smooth case if the additive error is $\mathcal{O}(1/\sqrt{k})$.

5.4. Minimax Problems

We consider the minimax problems of the following form:

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \mathcal{L}(Ax, y)$$

where $\mathcal{L}$ is a smooth convex-concave function, *i.e.*, $\mathcal{L}(\cdot, y)$ is convex $\forall y \in \mathcal{Y}$ and $\mathcal{L}(Ax, \cdot)$ is concave $\forall x \in \mathcal{X}$. Note that this formulation is a special instance of (5.1) with $g(Ax) = \max_{y\in\mathcal{Y}} \mathcal{L}(Ax, y)$. Consequently, we can apply Algorithm 2 if prox_g is tractable.

When $\mathcal{Y}$ admits an efficient projection oracle, prox_g is also efficient for bilinear saddle point problems $\mathcal{L}(Ax, y) = \langle Ax,\, y\rangle$. By Moreau decomposition, we have

$$\mathrm{prox}_g(Ax_k) = Ax_k - \mathrm{proj}_{\mathcal{Y}}(Ax_k),$$

hence v_k takes the form

$$v_k = \beta_k \nabla f(x_k) + A^\top \mathrm{proj}_{\mathcal{Y}}(Ax_k).$$

Gidel et al. (2017) proposes a CGM variant for the smooth convex-concave saddle point problems. This method processes both x and y via the lmo, and hence it also requires $\mathcal{Y}$ to be bounded. Our method, on the other hand, is more suitable when $\mathrm{proj}_{\mathcal{Y}}$ is easy.

Bilinear saddle point problem covers the maximum margin estimation of structured output models (Taskar et al., 2006) and minimax games (Von Neumann & Morgenstern, 1944). In particular, it also covers an important semidefinite programming formulation (Garber & Hazan, 2016), where $\mathcal{X}$ is a spactrahedron and $\mathcal{Y}$ is the simplex. Our framework fits perfectly here since the projection onto the simplex can be computed efficiently. We defer the extension of our framework with the entropy Bregman smoothing for future.

Note that CGM is applicable also for the variational inequality problems beyond (1.1), see (Hammond, 1984), (Juditsky & Nemirovski, 2016) and (Cox et al., 2017).

5.5. Problems with Affine Constraints

Algorithm 2 also applies to smooth convex minimization problems with affine constraints over a convex compact set:

$$\min_{x\in\mathcal{X}} f(x) \quad \text{subject to} \quad Ax = b, \tag{5.2}$$

by setting $g(Ax)$ in (1.1) as indicator function of set $\{b\}$, where $b \in \mathbb{R}^d$ is a known vector.

Since the prox operator of the indicator function of a convex set is the projection, v_k in Algorithm 2 becomes

$$v_k = \beta_k \nabla f(x_k) + A^\top (Ax_k - b).$$

Gidel et al. (2018) recently proposed a CGM framework via augmented Lagrangian for $Ax = 0$ constraint. Their method achieves $\mathcal{O}(1/\sqrt{k})$ rate in feasibility gap, and $\mathcal{O}(1/k)$ rate in augmented Lagrangian residual. Note however, these alone do not imply convergence in the objective residual. Moreover, the step-size of the proposed method depends on the unknown error bound constant (*cf.* (Bolte et al., 2017) for more details on error bounds).

Liu et al. (2018) introduces an inexact augmented Lagrangian method, where the subproblems are approximately solved by CGM up to an accuracy. This method produces an ϵ-solution in $\mathcal{O}(1/\epsilon^2)$ iterations (see Corollary 4.4 in (Liu et al., 2018)), but it calls lmo an unknown number of times bounded by $\mathcal{O}(k^2)$ at k^{th} iteration.

Another relevant approach here is the universal primal-dual gradient method (UPD) proposed by Yurtsever et al. (2015). UPD takes advantage of Fenchel-type oracles, which can be thought as a generalization of lmo.

Unfortunately, UPD iterations explicitly depend on the target accuracy level ϵ, which is difficult to tune without a rough knowledge of the optimal value. Moreover, the method converges only up to ϵ-suboptimality. There is no known analysis with inexact oracle calls for UPD, and errors in function evaluation can cause the algorithm to get stuck in the line-search procedure.

We can generalize (5.2) for the problems with affine inclusion constraints:

$$\min_{x\in\mathcal{X}} f(x) \quad \text{subject to} \quad Ax - b \in \mathcal{K}, \tag{5.3}$$

where $\mathcal{K}$ is a simple closed convex set. In this case, v_k takes the following form:

$$v_k = \beta_k \nabla f(x_k) + A^\top \big(Ax_k - b - \mathrm{proj}_{\mathcal{K}}(Ax_k - b)\big).$$

We implicitly assume that $\mathrm{proj}_{\mathcal{K}}$ is tractable. We can use a splitting framework when it is computationally more advantageous to use $\mathrm{lmo}_{\mathcal{K}}$ instead (*cf.* Section 5.6).

This template covers the standard semidefinite programming in particular. Applications include clustering (Peng & Wei, 2007), optimal power-flow (Lavaei & Low, 2012), sparse PCA (d'Aspremont et al., 2007), kernel learning (Lanckriet et al., 2004), blind deconvolution (Ahmed et al., 2014), community detection (Bandeira et al., 2016), *etc.* Besides machine learning applications, this formulation has a crucial role in the convex relaxation of combinatorial problems.

A significant example is the problems over the doubly nonnegative cone (*i.e.*, the intersection of the positive semidefinite cone and the positive orthant) with a bounded trace norm (Yoshise & Matsukawa, 2010). Note that the lmo over this domain can be costly since the lmo can require full dimensional updates (Hamilton-Jester & Li, 1996; Locatello et al., 2017b).

Our framework can handle these problems ensuring the positive semidefiniteness by $\mathrm{lmo}_{\mathcal{X}}$, and can still ensure the convergence to the first orthant via $\mathrm{proj}_{\mathcal{K}}$. To the best of our knowledge, our framework is the first CGM extension that can handle affine constraints.

5.6. Minimization via Splitting

We can take advantage of splitting since we can handle affine constraints. This lets us to disentangle the complexity of the constraints. Consider the following optimization template:

$$\begin{aligned} \min_{x\in\mathcal{X}_1\cap\mathcal{X}_2} \quad & f(x) + g(Ax) \\ \text{subject to} \quad & Bx - b \in \mathcal{K}, \quad Cx - c \in \mathcal{S} \end{aligned}$$

where $\mathcal{X}_1$ and $\mathcal{X}_2 \subset \mathbb{R}^n$ are two convex compact sets, A, B, C are known matrices and b, c are given vectors.

Suppose that
- $\mathrm{lmo}_{\mathcal{X}_1}$ and $\mathrm{lmo}_{\mathcal{X}_2}$ are easy to compute, but not $\mathrm{lmo}_{\mathcal{X}_1\cap\mathcal{X}_2}$
- prox_g is easy to compute
- $\mathcal{K}$ is a simple convex set and $\mathrm{proj}_{\mathcal{K}}$ is efficient
- $\mathcal{S}$ is a convex compact set with an efficient lmo.

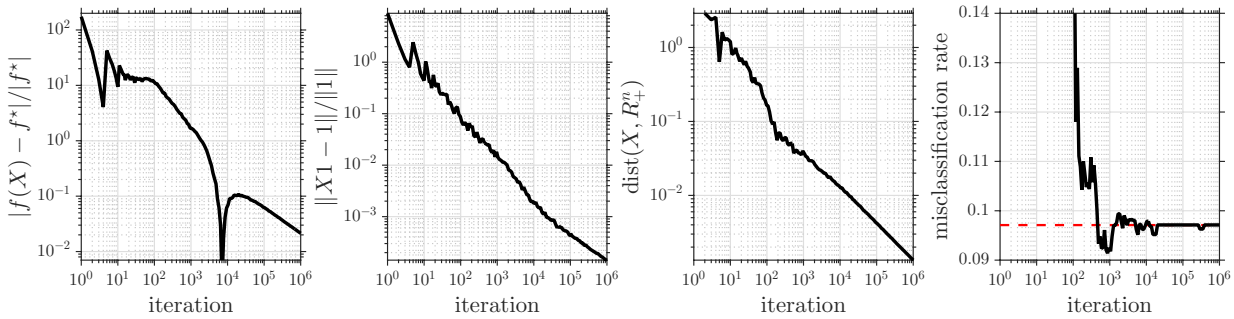

Figure 1. Clustering MNIST dataset: Convergence of our framework in function value and the feasibility gap. Red dashed line on the misclassification plot represents the value reported by Mixon et al. (2017).

We can reformulate this problem introducing slack variables $\xi \in \mathcal{X}_2$ and $\psi \in \mathcal{S}$ as follows:

$$\begin{aligned} \min_{\substack{x\in\mathcal{X}_1\ \xi\in\mathcal{X}_2\\ \psi\in\mathcal{S}}} \quad & f(x) + g(Ax) \\ \text{subject to} \quad & Bx - b \in \mathcal{K}, \quad Cx - c = \psi, \quad x = \xi. \end{aligned}$$

This formulation is in the form of (5.3) with respect to the variable $(x, \xi, \psi) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{S}$. It is easy to verify that Algorithm 2 leverages $\mathrm{lmo}_{\mathcal{X}_1}$, $\mathrm{lmo}_{\mathcal{X}_2}$, $\mathrm{lmo}_{\mathcal{S}}$, prox_g and $\mathrm{proj}_{\mathcal{K}}$ separately. This approach can be generalized for an arbitrary finite number of non-smooth terms in a straightforward way.

6. Numerical Experiments

This section presents numerical experiments supporting our theoretical findings in clustering and robust PCA examples. The non-smooth parts in the chosen examples consist of indicator functions, for which the dual domain is unbounded. Hence, to the best of our knowledge, other CGM variants in the literature are not applicable.

6.1. Clustering the MNIST dataset

We consider the model-free k-means clustering based on the semidefinite relaxation of Peng & Wei (2007):

$$\min_{X\in\mathcal{X}} \langle D, X\rangle \quad \text{subject to} \quad \underbrace{X1 = 1, \quad X \geq 0}_{g}, \qquad (6.1)$$

where $\mathcal{X} = \{X \in \mathbb{R}^{n\times n} : X \succcurlyeq 0,\ \mathrm{tr}(X) \leq \rho\}$ is the set of positive semidefinite matrices with a bounded trace norm, and $D \in \mathbb{R}^{n\times n}$ is the Euclidean distance matrix.

We use the setup described and published online by Mixon et al. (2017), which can be briefly described as follows: First, meaningful features from MNIST dataset (LeCun & Cortes), which consists of 28×28 grayscale images that can be stacked as 784×1 vectors, are extracted using a one-layer neural network. This gives us a weight matrix $W \in \mathbb{R}^{784\times 10}$ and a bias vector $b \in \mathbb{R}^{10}$. Then, the trained neural network is applied to the first 1000 elements of the test set, which gives the probability vectors for these 1000 test points, where each entry represents the probability of being each digit.

Mixon et al. (2017) runs a relax-and-round algorithm which solves (6.1) by SDPNAL+ (Yang et al., 2015) followed by a rounding scheme (see Section 5 of (Mixon et al., 2017) for details), and compares the results against MATLAB's built-in k-means++ implementation. Relax-and-round method is reported to achieve a misclassification rate of 0.0971. This rate matches with the all-time best rate for k-means++ after 100 different runs with random initializations.

For this experiment, we solve (6.1) by using Algorithm 2. Then, we cluster data using the same rounding scheme as (Mixon et al., 2017). We initialize our method from zeros, and we choose $\beta_0 = 1$. We implement lmo using the built-in MATLAB function `eigs` with tolerance parameter 10^{-9}.

We present the results of this experiment in Figure 1. We observe empirical $\mathcal{O}(1/\sqrt{k})$ rate both in the objective residual and the feasibility gap. Surprisingly, the method achieves the best test error around 1000 iterations achieveing the misclassification rate of 0.0914. This improves the value reported by Mixon et al. (2017) by 5.8%.

This example suggests that the slow convergence rate is not a major problem in many machine learning problems, since a low accuracy solution can generalize as well as the optimal point in terms of the test error, if not better.

6.2. Robust PCA

Suppose that we are given a large matrix that can be decomposed as the summation of a low-rank and a sparse (in some representation) matrix. Robust PCA aims to recover these components accurately. Robust PCA has many applications in machine learning and data science, such as collaborative filtering, system identification, genotype imputation, etc. Here, we focus on an image decomposition problem so that we can visualize the decomposition error results.

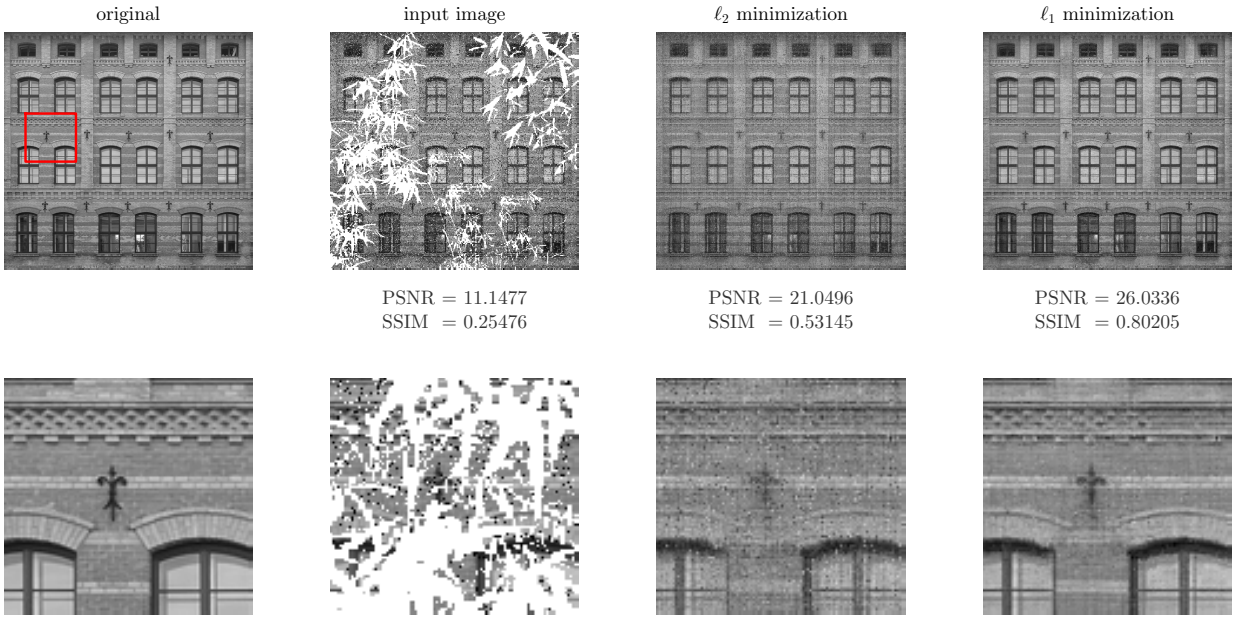

Figure 2. Image inpainting from noisy test image (493×517): Robust PCA recovers a better approximation with 5dB higher PSNR.

Our setting is similar to the setup described in (Zeng & So, 2018). We consider a scaled grayscale photograph with pattern from (Liu et al., 2013), and we assume that we only have access to an occluded image. Moreover, the image is contaminated by salt and pepper noise of density $1/10$. We seek to approximate the original from this noisy image.

This is essentially a matrix completion problem, and most of the scalable techniques rely on the Gaussian noise model. Note however the corresponding least-squares formulation is a good model against outliers:

$$\min_{X \in \mathcal{X}} \frac{1}{2}\|A(X) - b\|^2 \quad \text{subject to} \quad 0 \leq X \leq 1,$$

where $\mathcal{X} = \{X \in \mathbb{R}^{n \times n} : \|X\|_{S_1} \leq \rho\}$ is a scaled nuclear norm ball, and $A : \mathbb{R}^{n \times n} \to \mathbb{R}^d$ is the sampling operator.

Our framework also covers the following least absolute deviations formulation which is known to be more robust:

$$\min_{X \in \mathcal{X}} \|A(X) - b\|_1 \quad \text{subject to} \quad 0 \leq X \leq 1.$$

We solve both formulations with our framework, starting from all zero matrix, running 1000 iterations, and assuming that we know the true nuclear norm of the original image. We choose $\beta_0 = 1$ in both cases.

This experiment demonstrates the implications of the flexibility of our framework in a simple machine learning setup. We compile the results in Figure 2, where the non-smooth formulation recovers a better approximation with 5dB higher peak signal to noise ratio (PSNR) and 0.27 higher structural similarity index (SSIM). Evaluation of PSNR and SSIM vs iteration counter are shown in Figure 3.

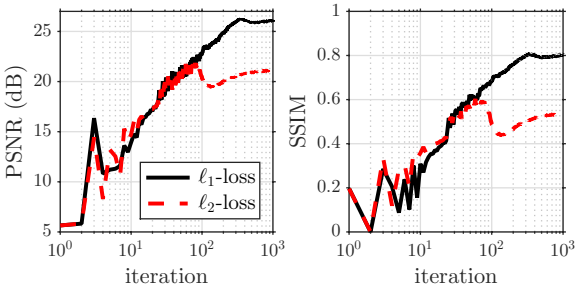

Figure 3. PSNR and SSIM vs iteration counter for formulations with ℓ_1 and ℓ_2 loss.

7. Conclusions

We presented a CGM framework for the composite convex minimization template, that provably achieves the optimal rate. This rate also holds under approximate oracle calls with additive or multiplicative errors.

Apart from its generalizations for various templates, there has been many attempts to improve the convergence rate, the arithmetic and the storage cost, or the proof techniques of CGM under some specific settings, *cf.* (Dunn, 1979; Guélat & Marcotte, 1986; Beck, 2004; Garber & Hazan, 2015; Lacoste-Julien & Jaggi, 2015; Odor et al., 2016; Freund & Grigas, 2016; Yurtsever et al., 2017) and references therein.

Many of these techniques can be adapted to our framework, since we preserve key features of CGM, such as the reduced costs and atomic representations. The only seeming drawback is the loss of affine invariance, left for future, which is fundamentally challenging due to smoothing technique.

Acknowledgements

[1]This work was supported by the Swiss National Science Foundation (SNSF) under grant number $200021\_178865/1$. [1]This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no 725594 - time-data). [2]This work was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH, in a joint call with PGMO. [3] [4]This project has received funding from the Max Planck ETH Center for Learning Systems and by ETH core funding (to Gunnar Rätsch).

References

Ahmed, A., Recht, B., and Romberg, J. Blind deconvolution using convex programming. *IEEE Trans. on Inf. Theory*, 60(3): 1711–1732, 2014.

Alacaoglu, A., Tran-Dinh, Q., Fercoq, O., and Cevher, V. Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization. In *Advances in Neural Information Processing Systems 30*, 2017.

Bandeira, A., Boumal, N., and Voroninski, V. On the low-rank approach for semidefinite programs arising in synchronization and community detection. *JMLR: Workshop and Conf. Proc.*, 49:1–22, 2016.

Beck, A. A conditional gradient method with linear rate of convergence for solving convex linear systems. *Math. Methods Oper. Res.*, 59(2):235–247, 2004.

Bolte, J., Nguyen, T. P., Peypouquet, J., and Suter, B. W. From error bounds to the complexity of first-order descent methods for convex functions. *Math. Program.*, 165(2):471–507, 2017.

Clarkson, K. L. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4), 2010.

Cox, B., Juditsky, A., and Nemirovski, A. Decomposition techniques for bilinear saddle point problems and variational inequalities with affine monotone operators. *J. Optim. Theory Appl.*, 2(402–435), 2017.

d'Aspremont, A., Ghaoui, L., Jordan, M., and Lanckriet, G. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

Dunn, J. Rates of convergence for conditional gradient algorithms near singular and nonsinglular extremals. *SIAM J. Control Optim.*, 17(2):187–211, 1979.

Dunn, J. Convergence rates for conditional gradient sequences generated by implicit step length rules. *SIAM J. Control Optim.*, 18(5):473–487, 1980.

Dunn, J. and Harshbarger, S. Conditional gradient algorithms with open loop step size rules. *J. Math. Anal. Appl.*, 62(2):432–444, 1978.

Dünner, C., Forte, S., Takác, M., and Jaggi, M. Primal–dual rates and certificates. In *Proc. 33rd Int. Conf. Machine Learning*, 2016.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

Freund, R. M. and Grigas, P. New analysis and results for the Frank–Wolfe method. *Math. Program.*, 155(1):199–230, 2016.

Garber, D. and Hazan, E. Faster rates for the Frank-Wolfe method over strongly-convex sets. In *Proc. 32nd Int. Conf. Machine Learning*, 2015.

Garber, D. and Hazan, E. Sublinear time algorithms for approximate semidefinite programming. *Math. Program., Ser. A*, (158): 329–361, 2016.

Gidel, G., Jebara, T., and Lacoste-Julien, S. Frank-Wolfe algorithms for saddle point problems. In *Proc. 20th Int. Conf. Artificial Intelligence and Statistics*, 2017.

Gidel, G., Pedregosa, F., and Lacoste-Julien, S. Frank-Wolfe splitting via augmented Lagrangian method. In *Proc. 21st Int. Conf. Artificial Intelligence and Statistics*, 2018.

Guélat, J. and Marcotte, P. Some comments on Wolfe's 'away step'. *Math. Program.*, 35(1):110–119, 1986.

Hamilton-Jester, C. L. and Li, C.-K. Extreme vectors of doubly nonnegative matrices. *Rocky Mountain J. Math.*, 26(4):1371–1383, 1996.

Hammond, H. J. *Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms*. PhD thesis, MIT, 1984.

Harchaoui, Z., Juditsky, A., and Nemirovski, A. Conditional gradient algorithms for norm–regularized smooth convex optimization. *Math. Program., Ser. A*, (152):75–112, 2015.

Hazan, E. Sparse approximate solutions to semidefinite programs. In *Proc. 8th Latin American Conf. Theoretical Informatics*, pp. 306–316, 2008.

Hazan, E. and Kale, S. Projection–free online learning. In *Proc. 29th Int. Conf. Machine Learning*, 2012.

Hazan, E. and Luo, H. Variance-reduced and projection-free stochastic optimization. In *Proc. 33rd Int. Conf. Machine Learning*, 2016.

He, N. and Harchaoui, Z. Semi–proximal mirror–prox for nonsmooth composite minimization. In *Advances in Neural Information Processing Systems 28*, 2015.

Jaggi, M. Revisiting Frank–Wolfe: Projection–free sparse convex optimization. In *Proc. 30th Int. Conf. Machine Learning*, 2013.

Juditsky, A. and Nemirovski, A. Solving variational inequalities with monotone operators on domains given by Linear Minimization Oracles. *Math. Program.*, 156(1-2):221–256, 2016.

Lacoste-Julien, S. and Jaggi, M. On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems 28*, 2015.

Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *Proc. 30th Int. Conf. Machine Learning*, 2013.

Lan, G. The complexity of large–scale convex programming under a linear optimization oracle. arXiv:1309.5550v2, 2014.

Lan, G. and Zhou, Y. Conditional gradient sliding for convex optimization. *SIAM J. Optim.*, 26(2):1379–1409, 2016.

Lan, G., Pokutta, S., Zhou, Y., and Zink, D. Conditional accelerated lazy stochastic gradient descent. In *Proc. 34th Int. Conf. Machine Learning*, 2017.

Lanckriet, G., Cristianini, N., Ghaoui, L., Bartlett, P., and Jordan, M. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

Lavaei, J. and Low, H. Zero duality gap in optimal power flow problem. *IEEE Trans. on Power Syst.*, 27(1):92–107, February 2012.

LeCun, Y. and Cortes, C. MNIST handwritten digit database, Accessed: Jan. 2016 . URL http://yann.lecun.com/exdb/mnist/.

Levitin, E. and Polyak, B. Constrained minimization methods. *USSR Comput. Math. & Math. Phys.*, 6(5):1–50, 1966.

Liu, J., Musialski, P., Wonka, P., and Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell*, 35(1):208–230, 2013.

Liu, Y.-F., Liu, X., and Ma, S. On the non-ergodic convergence rate of an inexact augmented Lagrangian framework for composite convex programming. *to appear in Math. Oper. Res.*, 2018.

Locatello, F., Khanna, R., Tschannen, M., and Jaggi, M. A unified optimization view on generalized matching pursuit and Frank-Wolfe. In *Proc. 20th Int. Conf. Artificial Intelligence and Statistics*, 2017a.

Locatello, F., Tschannen, M., Rätsch, G., and Jaggi, M. Greedy algorithms for cone constrained optimization with convergence guarantees. In *Advances in Neural Information Processing Systems 30*, 2017b.

Mixon, D. G., Villar, S., and Ward, R. Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA*, 6(4):389–415, 2017.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1987.

Nesterov, Y. Smooth minimization of non-smooth functions. *Math. Program.*, 103:127–152, 2005.

Nesterov, Y. Complexity bounds for primal-dual methods minimizing the model of objective function. *Math. Program.*, 2017.

Odor, G., Li, Y.-H., Yurtsever, A., Hsieh, Y.-P., Tran-Dinh, Q., El Halabi, M., and Cevher, V. Frank-Wolfe works for non-lipschitz continuous gradient objectives: Scalable poisson phase retrieval. In *41st IEEE Int. Conf. Acoustics, Speech & Signal Processing*, 2016.

Peng, J. and Wei, Y. Approximating K–means–type clustering via semidefinite programming. *SIAM J. Optim.*, 18(1):186–205, 2007.

Richard, E., Savalle, P.-A., and Vayatis, N. Estimation of simultaneously sparse and low rank matrices. In *Proc. 29th Int. Conf. Machine Learning*, 2012.

Taskar, B., Lacoste-Julien, S., and Jordan, M. Structured prediction, dual extragradient and Bregman projections. *J. Mach. Learn. Res.*, 2006.

Tran-Dinh, Q., Fercoq, O., and Cevher, V. A smooth primal-dual optimization framework for nonsmooth composite convex minimization. *SIAM J. Optim.*, 28(1):96–134, 2018.

Von Neumann, J. and Morgenstern, O. *Theory of games and economic behavior*. Princeton press, 1944.

Xu, H.-K. Convergence analysis of the Frank–Wolfe algorithm and its generalization in Banach spaces. arXiv:1710.07367v1, 2017.

Yang, L., Sun, D., and Toh, K.-C. SDPNAL+: A majorized semismooth Newton–CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Math. Program. Comput.*, 7(3):331–366, 2015.

Yen, I. E.-H., Lin, X., Zhang, J., Ravikumar, P., and Dhillon, I. S. A convex atomic–norm approach to multiple sequence alignment and motif discovery. In *Proc. 33rd Int. Conf. Machine Learning*, 2016.

Yoshise, A. and Matsukawa, Y. On optimization over the doubly nonnegative cone. In *IEEE International Symposium on Computer-Aided Control System Design*, pp. 13–18, Yokohama, 2010.

Yurtsever, A., Tran-Dinh, Q., and Cevher, V. A universal primal-dual convex optimization framework. In *Advances in Neural Information Processing Systems 28*, 2015.

Yurtsever, A., Udell, M., Tropp, J., and Cevher, V. Sketchy decisions: Convex low-rank matrix optimization with optimal storage. In *Proc. 20th Int. Conf. Artificial Intelligence and Statistics*, 2017.

Zeng, W.-J. and So, H. C. Outlier–robust matrix completion via ℓ_p-minimization. *IEEE Trans. on Sig. Process*, 66(5):1125–1140, 2018.

Appendix

A1 Preliminaries

The following properties of smoothing are key to derive the convergence rate of our algorithm.

Let $g : \mathbb{R}^d \to \mathbb{R} \cap \{+\infty\}$ be a a proper, closed and convex function, and denote its smooth approximation by

$$g_\beta(z) = \max_{y \in \mathbb{R}^d} \langle z,\, y \rangle - g^*(y) - \frac{\beta}{2}\|y\|^2$$

where g^* represents the Fenchel conjugate of g and $\beta > 0$ is the smoothing parameter. Then, g_β is convex and $\frac{1}{\beta}$-smooth. Let us denote the unique maximizer of this concave problem by

$$\begin{aligned} y^*_\beta(z) &= \arg\max_{y \in \mathbb{R}^d} \langle z,\, y \rangle - g^*(y) - \frac{\beta}{2}\|y\|^2 \\ &= \arg\min_{y \in \mathbb{R}^d} \frac{1}{\beta} g^*(y) - \frac{1}{\beta}\langle z,\, y \rangle + \frac{1}{2}\|y\|^2 + \frac{1}{2}\|\frac{1}{\beta} z\|^2 \\ &= \arg\min_{y \in \mathbb{R}^d} \frac{1}{\beta} g^*(y) + \frac{1}{2}\|y - \frac{1}{\beta} z\|^2 \\ &= \operatorname{prox}_{\beta^{-1} g^*}(\beta^{-1} z) = \frac{1}{\beta}\big(z - \operatorname{prox}_{\beta g}(z)\big) \end{aligned}$$

where the last equality is known as the Moreau decomposition. Then, the followings hold for $\forall z_1, z_2 \in \mathbb{R}^d$ and $\forall \beta, \gamma > 0$

$$g_\beta(z_1) \geq g_\beta(z_2) + \langle \nabla g_\beta(z_2),\, z_1 - z_2 \rangle + \frac{\beta}{2}\|y^*_\beta(z_2) - y^*_\beta(z_1)\|^2 \tag{7.1}$$

$$g(z_1) \geq g_\beta(z_2) + \langle \nabla g_\beta(z_2),\, z_1 - z_2 \rangle + \frac{\beta}{2}\|y^*_\beta(z_2)\|^2 \tag{7.2}$$

$$g_\beta(z_1) \leq g_\gamma(z_1) + \frac{\gamma - \beta}{2}\|y^*_\beta(z_1)\|^2 \tag{7.3}$$

Proofs can be found in Lemma 10 from (Tran-Dinh et al., 2018).

Suppose that g is L_g-Lipschitz continuous. Then, for any $\beta > 0$ and any $z \in \mathbb{R}^d$, the following bound holds:

$$g_\beta(z) \leq g(z) \leq g_\beta(z) + \frac{\beta}{2} L_g^2 \tag{7.4}$$

Proof follows from equation (2.7) in (Nesterov, 2005) with a remark on the duality between Lipshitzness and bounded support (*cf.* Lemma 5 in (Dünner et al., 2016)).

A2 Convergence analysis

This section presents the proof of our convergence results. We skip proofs of Theorems 3.1 to 3.3 since we can get these results as a special case by setting $\delta = 0$ in Theorems 4.1 to 4.3.

Proof of Theorem 4.1

First, we use the smoothness of F_{β_k} to upper bound the progress. Note that F_{β_k} is $(L_f + \|A\|^2/\beta_k)$-smooth.

$$\begin{aligned} F_{\beta_k}(x_{k+1}) &\leq F_{\beta_k}(x_k) + \eta_k\langle \nabla F_{\beta_k}(x_k),\, \tilde{s}_k - x_k\rangle + \frac{\eta_k^2}{2}\|\tilde{s}_k - x_k\|^2(L_f + \frac{\|A\|^2}{\beta_k}) \\ &\leq F_{\beta_k}(x_k) + \eta_k\langle \nabla F_{\beta_k}(x_k),\, \tilde{s}_k - x_k\rangle + \frac{\eta_k^2}{2}D_{\mathcal{X}}^2(L_f + \frac{\|A\|^2}{\beta_k}), \end{aligned} \tag{7.5}$$

where $\tilde{s}_k$ denotes the atom selected by the inexact lmo, and the second inequality follows since $\tilde{s}_k \in \mathcal{X}$.

By definition of inexact oracle (4.1), we have

$$\begin{aligned} \langle \nabla F_{\beta_k}(x_k),\, \tilde{s}_k - x_k\rangle &\leq \langle \nabla F_{\beta_k}(x_k),\, s_k - x_k\rangle + \delta\frac{\eta_k}{2}D_{\mathcal{X}}^2(L_f + \frac{\|A\|^2}{\beta_k}) \\ &\leq \langle \nabla F_{\beta_k}(x_k),\, x^\star - x_k\rangle + \delta\frac{\eta_k}{2}D_{\mathcal{X}}^2(L_f + \frac{\|A\|^2}{\beta_k}) \\ &= \langle \nabla f(x_k),\, x^\star - x_k\rangle + \langle A^\top \nabla g_{\beta_k}(Ax_k),\, x^\star - x_k\rangle + \delta\frac{\eta_k}{2}D_{\mathcal{X}}^2(L_f + \frac{\|A\|^2}{\beta_k}), \end{aligned}$$

where the second line follows since s_k is a solution of $\min_{x\in\mathcal{X}}\langle \nabla F_{\beta_k}(x_k),\, x\rangle$.

Now, convexity of f ensures $\langle \nabla f(x_k), x^\star - x_k\rangle \leq f(x^\star) - f(x_k)$. Using property (7.2), we have

$$\begin{aligned} \langle A^\top \nabla g_{\beta_k}(Ax_k),\, x^\star - x_k\rangle &= \langle \nabla g_{\beta_k}(Ax_k),\, Ax^\star - Ax_k\rangle \\ &\leq g(Ax^\star) - g_{\beta_k}(Ax_k) - \frac{\beta_k}{2}\|y^*_{\beta_k}(Ax_k)\|^2. \end{aligned}$$

Putting these altogether, we get the following bound

$$\begin{aligned} F_{\beta_k}(x_{k+1}) &\leq F_{\beta_k}(x_k) + \eta_k\left(f(x^\star) - f(x_k) + g(Ax^\star) - g_{\beta_k}(Ax_k) - \frac{\beta_k}{2}\|\nabla y^*_{\beta_k}(Ax_k)\|^2\right) \\ &\quad + \frac{\eta_k^2}{2}D_{\mathcal{X}}^2(L_f + \frac{\|A\|^2}{\beta_k})(1+\delta) \\ &= (1-\eta_k)F_{\beta_k}(x_k) + \eta_k F(x^\star) - \frac{\eta_k\beta_k}{2}\|\nabla y^*_{\beta_k}(Ax_k)\|^2 + \frac{\eta_k^2}{2}D_{\mathcal{X}}^2(L_f + \frac{\|A\|^2}{\beta_k})(1+\delta). \end{aligned} \tag{7.6}$$

Now, using (7.3), we get

$$\begin{aligned} F_{\beta_k}(x_k) &= f(x_k) + g_{\beta_k}(Ax_k) \\ &\leq f(x_k) + g_{\beta_{k-1}}(Ax_k) + \frac{\beta_{k-1} - \beta_k}{2}\|y^*_{\beta_k}(Ax_k)\|^2 \\ &= F_{\beta_{k-1}}(x_k) + \frac{\beta_{k-1} - \beta_k}{2}\|y^*_{\beta_k}(Ax_k)\|^2. \end{aligned}$$

We combine this with (7.6) and subtract $F(x^\star)$ from both sides to get

$$\begin{aligned} F_{\beta_k}(x_{k+1}) - F(x^\star) &\leq (1-\eta_k)\big(F_{\beta_{k-1}}(x_k) - F(x^\star)\big) + \frac{\eta_k^2}{2}D_{\mathcal{X}}^2(L_f + \frac{\|A\|^2}{\beta_k})(1+\delta) \\ &\quad + \big((1-\eta_k)(\beta_{k-1} - \beta_k) - \eta_k\beta_k\big)\frac{1}{2}\|y^*_{\beta_k}(Ax_k)\|^2. \end{aligned}$$

Let us choose η_k and β_k in a way to vanish the last term. By choosing $\eta_k = \frac{2}{k+1}$ and $\beta_k = \frac{\beta_0}{\sqrt{k+1}}$ for $k \geq 1$ with some $\beta_0 > 0$, we get $(1-\eta_k)(\beta_{k-1} - \beta_k) - \eta_k \beta_k < 0$. Hence, we end up with

$$F_{\beta_k}(x_{k+1}) - F(x^\star) \leq (1-\eta_k)\big(F_{\beta_{k-1}}(x_k) - F(x^\star)\big) + \frac{\eta_k^2}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1+\delta).$$

By recursively applying this inequality, we get

$$\begin{aligned} F_{\beta_k}(x_{k+1}) - F(x^\star) &\leq \prod_{j=1}^{k}(1-\eta_j)\left(F_{\beta_{j-1}}(x_k) - F(x^\star)\right) + \frac{1}{2} D_{\mathcal{X}}^2 (1+\delta) \sum_{\ell=1}^{k} \eta_\ell^2 \prod_{j=\ell}^{k} (1-\eta_j)(L_f + \frac{\|A\|^2}{\beta_\ell}) \\ &\leq \prod_{j=1}^{k}(1-\eta_j)\left(F_{\beta_{j-1}}(x_k) - F(x^\star)\right) + \frac{1}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1+\delta) \sum_{\ell=1}^{k} \eta_\ell^2 \prod_{j=\ell}^{k} (1-\eta_j) \\ &= \frac{1}{2} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1+\delta) \sum_{\ell=1}^{k} \eta_\ell^2 \prod_{j=\ell}^{k} (1-\eta_j), \end{aligned}$$

where the second line follows since $\beta_k \leq \beta_j$ for any positive integer $j \leq k$, and the third line since $\eta_1 = 1$.

Now, we use the following relation

$$\sum_{\ell=1}^{k} \eta_\ell^2 \prod_{j=\ell}^{k}(1-\eta_j) = \sum_{\ell=1}^{k} \frac{4}{(\ell+1)^2} \prod_{j=\ell}^{k} \frac{j-1}{j+1} = \sum_{\ell=1}^{k} \frac{4}{(\ell+1)^2} \frac{(\ell-1)\ell}{k(k+1)} \leq \frac{4}{k+1},$$

which yields the first result of Theorem 4.1 as

$$F_{\beta_k}(x_{k+1}) - F(x^\star) \leq \frac{2}{k+1} D_{\mathcal{X}}^2 (L_f + \frac{\|A\|^2}{\beta_k})(1+\delta) = 2D_{\mathcal{X}}^2 (\frac{L_f}{k+1} + \frac{\|A\|^2}{\beta_0 \sqrt{k+1}})(1+\delta).$$

Proof of Theorem 4.2

Now, we further assume that $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is L_g-Lipschitz continuous. From (7.4), we get

$$g(Ax_{k+1}) \leq g_{\beta_k}(Ax_{k+1}) + \frac{\beta_k L_g^2}{2} = g_{\beta_k}(Ax_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{k+1}}.$$

We complete the proof by adding $f(x_{k+1}) - F(x^\star)$ to both sides:

$$F(x_{k+1}) - F(x^\star) \leq F_{\beta_k}(x_{k+1}) - F(x^\star) + \frac{\beta_0 L_g^2}{2\sqrt{k+1}}.$$

Proof of Theorem 4.3

From the Lagrange saddle point theory, we know that the following bound holds $\forall x \in \mathcal{X}$ and $\forall r \in \mathcal{K}$:

$$f^\star \leq \mathcal{L}(x, r, y^\star) = f(x) + \langle y_\star,\, Ax - r \rangle \leq f(x) + \|y_\star\| \|Ax - r\|,$$

Since $x_{k+1} \in \mathcal{X}$, we get

$$f(x_{k+1}) - f^\star \geq -\min_{r \in \mathcal{K}} \|y^\star\| \|Ax_{k+1} - r\| = -\|y^\star\| \mathrm{dist}(Ax_{k+1}, \mathcal{K}). \tag{7.7}$$

This proves the first bound in Theorem 4.3.

The second bound directly follows by Theorem 4.1 as

$$f(x_{k+1}) - f^\star \leq \underbrace{f(x_{k+1}) - f^\star + \frac{1}{2\beta_k} \mathrm{dist}^2(Ax_{k+1}, \mathcal{K})}_{F_{\beta_k}(x_{k+1}) - F^\star} \leq 2D_{\mathcal{X}}^2 (\frac{L_f}{k+1} + \frac{\|A\|^2}{\beta_0 \sqrt{k+1}})(1+\delta).$$

Now, we combine this with (7.7), and we get

$$-\|y^\star\|\mathrm{dist}(Ax_{k+1},\mathcal{K})+\frac{1}{2\beta_k}\mathrm{dist}^2(Ax_{k+1},\mathcal{K}) \le 2D_{\mathcal{X}}^2\left(\frac{L_f}{k+1}+\frac{\|A\|^2}{\beta_0\sqrt{k+1}}\right)(1+\delta)$$
$$\le 2D_{\mathcal{X}}^2\frac{\beta_k}{\beta_0}\left(L_f+\frac{\|A\|^2}{\beta_0}\right)(1+\delta).$$

This is a second order inequality in terms of $\mathrm{dist}(Ax_k,\mathcal{K})$. Solving this inequality, we get

$$\mathrm{dist}(Ax_{k+1},\mathcal{K}) \le \beta_k\left(\|y^\star\|+\sqrt{\|y^\star\|^2+4D_{\mathcal{X}}^2\frac{1}{\beta_0}\left(L_f+\frac{\|A\|^2}{\beta_0}\right)(1+\delta)}\right)$$
$$\le \frac{2\beta_0}{\sqrt{k+1}}\left(\|y^\star\|+D_{\mathcal{X}}\sqrt{\frac{1}{\beta_0}\left(L_f+\frac{\|A\|^2}{\beta_0}\right)(1+\delta)}\right).$$

Proof of Theorem 4.4

Let us define the multiplicative error δ of the LMO:

$$\left\langle v_k,\,\tilde{s}_k-x_k\right\rangle \le \delta\left\langle v_k,\,s_k-x_k\right\rangle \tag{7.8}$$

For the proof we assume that x_1 is feasible. First, we use the smoothness of F_{β_k} to upper bound the progress. Note that F_{β_k} is $(L_f+\|A\|^2/\beta_k)$-smooth.

$$F_{\beta_k}(x_{k+1}) \le F_{\beta_k}(x_k)+\eta_k\left\langle\nabla F_{\beta_k}(x_k),\,\tilde{s}_k-x_k\right\rangle+\frac{\eta_k^2}{2}\|\tilde{s}_k-x_k\|^2\left(L_f+\frac{\|A\|^2}{\beta_k}\right)$$
$$\le F_{\beta_k}(x_k)+\eta_k\left\langle\nabla F_{\beta_k}(x_k),\,\tilde{s}_k-x_k\right\rangle+\frac{\eta_k^2}{2}D_{\mathcal{X}}^2\left(L_f+\frac{\|A\|^2}{\beta_k}\right), \tag{7.9}$$

where $\tilde{s}_k$ denotes the atom selected by the inexact linear minimization oracle, and the second inequality follows since $\tilde{s}_k\in\mathcal{X}$.

By definition of inexact oracle (7.8), we have

$$\left\langle\nabla F_{\beta_k}(x_k),\,\tilde{s}_k-x_k\right\rangle \le \delta\left\langle\nabla F_{\beta_k}(x_k),\,s_k-x_k\right\rangle$$
$$\le \delta\left\langle\nabla F_{\beta_k}(x_k),\,x^\star-x_k\right\rangle$$
$$= \delta\left\langle\nabla f(x_k),\,x^\star-x_k\right\rangle+\delta\left\langle A^\top\nabla g_{\beta_k}(Ax_k),\,x^\star-x_k\right\rangle,$$

where the second line follows since s_k is a solution of $\min_{x\in\mathcal{X}}\left\langle\nabla F_{\beta_k}(x_k),\,x\right\rangle$.

Now, convexity of f ensures $\langle\nabla f(x_k),x^\star-x_k\rangle\le f(x^\star)-f(x_k)$. Using property (7.2), we have

$$\left\langle A^\top\nabla g_{\beta_k}(Ax_k),\,x^\star-x_k\right\rangle = \left\langle\nabla g_{\beta_k}(Ax_k),\,Ax^\star-Ax_k\right\rangle$$
$$\le g(Ax^\star)-g_{\beta_k}(Ax_k)-\frac{\beta_k}{2}\|y_{\beta_k}^*(Ax_k)\|^2.$$

Putting these altogether, we get the following bound

$$F_{\beta_k}(x_{k+1}) \le F_{\beta_k}(x_k)+\eta_k\delta\left(f(x^\star)-f(x_k)+g(Ax^\star)-g_{\beta_k}(Ax_k)-\frac{\beta_k}{2}\|\nabla y_{\beta_k}^*(Ax_k)\|^2\right)$$
$$+\frac{\eta_k^2}{2}D_{\mathcal{X}}^2\left(L_f+\frac{\|A\|^2}{\beta_k}\right) \tag{7.10}$$
$$= (1-\delta\eta_k)F_{\beta_k}(x_k)+\delta\eta_k F(x^\star)-\frac{\delta\eta_k\beta_k}{2}\|\nabla y_{\beta_k}^*(Ax_k)\|^2+\frac{\eta_k^2}{2}D_{\mathcal{X}}^2\left(L_f+\frac{\|A\|^2}{\beta_k}\right).$$

Now, using (7.3), we get

$$\begin{aligned} F_{\beta_k}(x_k) &= f(x_k) + g_{\beta_k}(Ax_k) \\ &\leq f(x_k) + g_{\beta_{k-1}}(Ax_k) + \frac{\beta_{k-1}-\beta_k}{2}\|y^*_{\beta_k}(Ax_k)\|^2 \\ &= F_{\beta_{k-1}}(x_k) + \frac{\beta_{k-1}-\beta_k}{2}\|y^*_{\beta_k}(Ax_k)\|^2. \end{aligned}$$

We combine this with (7.10) and subtract $F(x^\star)$ from both sides to get

$$\begin{aligned} F_{\beta_k}(x_{k+1}) - F(x^\star) \leq (1-\delta\eta_k)\big(F_{\beta_{k-1}}(x_k) - F(x^\star)\big) + \frac{\eta_k^2}{2}D^2_{\mathcal{X}}(L_f + \frac{\|A\|^2}{\beta_k}) \\ + \big((1-\delta\eta_k)(\beta_{k-1}-\beta_k) - \delta\eta_k\beta_k\big)\frac{1}{2}\|\nabla y^*_{\beta_k}(Ax_k)\|^2. \end{aligned}$$

By choosing $\eta_k = \frac{2}{\delta(k-1)+2}$ and $\beta_k = \frac{\beta_0}{\sqrt{\delta k+1}}$ for some $\beta_0 > 0$, we get $(1-\delta\eta_k)(\beta_{k-1}-\beta_k) - \delta\eta_k\beta_k < 0$ for any $k \geq 1$, hence we end up with

$$F_{\beta_k}(x_{k+1}) - F(x^\star) \leq (1-\delta\eta_k)\big(F_{\beta_{k-1}}(x_k) - F(x^\star)\big) + \frac{\eta_k^2}{2}D^2_{\mathcal{X}}(L_f + \frac{\|A\|^2}{\beta_k}).$$

Let us call for simplicity $C := D^2_{\mathcal{X}}(L_f + \frac{\|A\|^2}{\beta_k})$, $E_{k+1} := F_{\beta_k}(x_{k+1}) - F(x^\star)$ Therefore, we have

$$E_{k+1} \leq (1-\delta\eta_k)E_k + \frac{\eta_k^2}{2}C \tag{7.11}$$

We now show by induction that:

$$E_k \leq 2\frac{\frac{1}{\delta}C + E_1}{\delta(k-1)+2}$$

The base case $k = 1$ is trivial as $C > 0$. Call for simplicity $K := \delta(k-1)+2$. Note that $K \geq 2$. Under this notation we can write $\eta_k = \frac{2}{\delta(k-1)+2} = \frac{2}{K}$ For the induction step, we add a positive term (E_1 is positive as x_1 is assumed feasible) to (7.11) and use the induction hypothesis:

$$\begin{aligned} E_{k+1} &\leq (1-\delta\eta_k)E_k + \frac{\eta_k^2}{2}C + 2\delta\frac{E_1}{K^2} \\ &\leq (1-\delta\frac{2}{K})E_k + \frac{2}{K^2}C + 2\delta\frac{E_1}{K^2} \\ &\leq (1-\delta\frac{2}{K})2\frac{\frac{1}{\delta}C+E_1}{K} + \frac{2}{K^2}C + 2\delta\frac{E_1}{K^2} \\ &= (1-\delta\frac{2}{K})2\frac{\frac{1}{\delta}C+E_1}{K} + 2\delta\left(\frac{\frac{1}{\delta}C}{K^2} + \frac{E_1}{K^2}\right) \\ &= 2\frac{\frac{1}{\delta}C+E_1}{K}\left(1-\delta\frac{2}{K}+\frac{\delta}{K}\right) \\ &= 2\frac{\frac{1}{\delta}C+E_1}{K}\left(1-\frac{\delta}{K}\right) \\ &\leq 2\frac{\frac{1}{\delta}C+E_1}{K+\delta} \end{aligned}$$

noting that $K + \delta = \delta k + 2$ concludes the proof.

Proof of Theorems 4.5 and 4.6 follows similarly to the proofs of Theorems 4.2 and 4.3.

A3 Numerical illustration for the pathological example in Section 5.3

This section considers the pathological example by Nesterov (2017) that we present in Section 5.3. We numerically demonstrate that our framework successfully finds the solution in contrast to the classical CGM.

Figure 4 illustrates the paths followed by classical CGM vs our framework, starting both methods from $[1,0]^\top$. Recall that the unique solution is $x^\star = [-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]^\top$, but classical CGM converges to $[-\frac{1}{2}, -\frac{1}{2}]^\top$, which belongs to the convex hull of $\{x_0, \left[\begin{smallmatrix}-1\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}0\\-1\end{smallmatrix}\right]\}$. Our framework, on the other hand, converges to $x^\star$.

Finally, Figure 5 plots the evolution of the objective residual as a function of the iteration counter. We observe an empirical $\mathcal{O}(1/k^2)$ convergence rate in this particular instance.

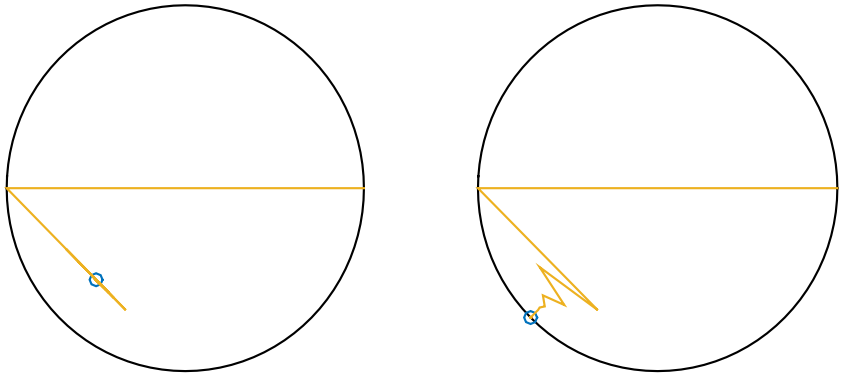

Figure 4. Classical CGM (*left*) vs our framework (*right*) for the pathological example, starting from $\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$.

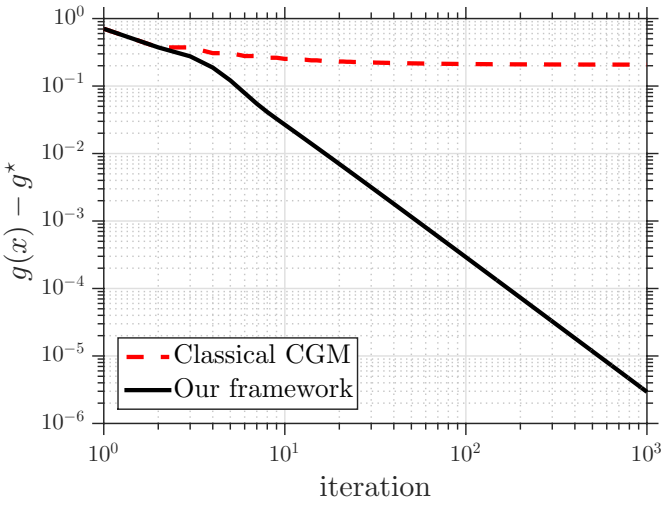

Figure 5. Evolution of the objective residual as a function of the iteration counter for the pathological example, starting from $\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$.