Frank-Wolfe Splitting via Augmented Lagrangian Method

Gauthier Gidel
MILA, DIRO Université de Montréal

Fabian Pedregosa
UC Berkeley & ETH Zurich

Simon Lacoste-Julien
MILA, DIRO Université de Montréal

Abstract

Minimizing a function over an intersection of convex sets is an important task in optimization that is often much more challenging than minimizations over each individual constraint set. While traditional methods such as Frank-Wolfe (FW) or proximal gradient descent assume access to a linear or quadratic oracle on the intersection, splitting techniques take advantage of the structure of each sets, and only require access to the oracle on the individual constraints. In this work we develop and analyze the Frank-Wolfe Augmented Lagrangian (FW-AL) algorithm, a method for minimizing a smooth function over convex compact sets related by a linear consistency constraint that only requires access to a linear minimization oracle over the individual constraints. It is based on the Augmented Lagrangian Method (ALM), also known as Method of Multipliers, but unlike most existing splitting methods it only requires access to linear (instead of quadratic) minimization oracles. We use recent advances in the analysis of Frank-Wolfe and the alternating direction method of multipliers algorithms to prove a sublinear convergence rate over general convex compact sets and a linear convergence rate of our algorithm over for polytopes.

1 Introduction

The Frank-Wolfe (FW) or conditional gradient algorithm has seen an impressive revival in recent years, notably due to its very favorable properties for the optimization of sparse problems (Jaggi, 2013). This algorithm assumes knowledge of a linear minimization oracle (LMO) over the set of constraints. This oracle is inexpensive to compute for sets such as the ℓ_1 or trace norm ball. However, inducing complex priors often requires to consider *multiple* constraints, leading to a constraint set formed by the intersection of the original constraints. Unfortunately, evaluating the LMO over this intersection can be very challenging even if the LMOs on the individual sets are inexpensive.

The problem of minimizing over an intersection of convex constraints is pervasive in machine learning and signal processing. For example, one can seek for a matrix that is both sparse and low rank by constraining the solution to have *both* small ℓ_1 and trace norm (Richard et al., 2012) or find a set of brain maps which are both sparse and piecewise constant by constraining both the ℓ_1 and total variation pseudonorm (Gramfort et al., 2013). Furthermore, some challenging optimization problems such as multiple sequence alignment are naturally expressed over an intersection of constraints (Yen et al., 2016a) or more generally as a linear relationship between these sets (Huang et al., 2017).

The objective of this paper is to describe and analyze FW-AL, an optimization method that can solve convex optimization problems subject to multiple constraint sets, assuming we have access to a LMO on each of the set.

Previous work. The vast majority of methods proposed to solve optimization problems over an intersection of sets of constraints rely on the availability of a projection operator onto each set (see e.g. the recent reviews (Glowinski et al., 2017; Ryu and Boyd, 2016), which cover the more general proximal splitting framework). One of the most popular algorithm in this framework is the alternating direction method of multipliers (ADMM), proposed by Glowinski and Marroco (1975), studied by Gabay and Mercier (1976), and revisited many times; see for instance (Boyd et al., 2011; Yan and Yin, 2016). On some cases, such as constraints on the trace norm (Cai et al., 2010) or the latent group lasso (Obozinski et al., 2011), the projection step can be a time-consuming operation, while

Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

the Frank-Wolfe LMO is much cheaper in both cases. Moreover, for some highly structured polytopes such as those appearing in alignment constraints (Alayrac et al., 2016) or Structured SVM (Lacoste-Julien et al., 2013), there exists a fast and elegant dynamic programming algorithm to compute the LMO, while there is no known practical algorithm to compute the projection. Hence, the development of splitting methods that use the LMO instead of the proximal operator is of key practical interest.

Recently, Yen et al. (2016a) proposed a FW variant for objectives with a linear loss function over an intersection of polytopes named Greedy Direction Method of Multipliers (GDMM). A similar version of GDMM is also used in (Yen et al., 2016b; Huang et al., 2017) to optimize a function over a Cartesian product of spaces related to each other by a linear consistency constraint. The constraints are incorporated through the augmented Lagrangian method and its convergence analysis crucially uses recent progress in the analysis of ADMM by Hong and Luo (2017). Nevertheless, we argue in Sec. C.1 that there are technical issues in these analysis since some of the properties used have only been proven for ADMM and do not hold in the context of GDMM. Furthermore, even though GDMM provides good experimental results in these papers, the practical applicability of the method to other problems is dampened by overly restrictive assumptions: the loss function is required to be linear or quadratic, leaving outside loss functions such as logistic regression, and the constraint needs to be a polytope, leaving outside domains such as the trace norm ball.

Yurtsever et al. (2015) propose an algorithm (UniPDGrad) based on Lagrangian method holding splitting with LMO as a particular case. We develop the comparison with FW-AL in App. B.2.

Contributions. Our main contribution is the development of a novel variant of FW for the optimization of a function over product of spaces related to each other by a linear consistency constraint and its rigorous analysis. We name this method Frank-Wolfe via Augmented Lagrangian method (FW-AL). With respect to Yen et al. (2016a,b); Huang et al. (2017), our framework generalizes GDMM by providing a method to optimize a general class of functions over an intersection of an arbitrary number of compact sets, which are *not* restricted to be polytopes. Moreover, we argue that the previous proofs of convergence were incomplete: in this paper, we prove a new challenging technical lemma providing a growth condition on the augmented dual function which allows us to fix the missing parts.

We show that FW-AL converges for any smooth objective function. We prove that a standard gap measure converges linearly (i.e., with a geometric rate) when the constraint sets are polytopes, and sublinearly for general compact convex sets. We also show that when the function is strongly convex, the sum of this gap measure and the feasibility gives a bound on the distance to the set of optimal solutions. This is of key practical importance since the applications that we consider (e.g., minimization with trace norm constraints) verify these assumptions.

The paper is organized as follows. In Section 2, we introduce the general setting, provide some motivating applications and present the augmented Lagrangian formulation of our problem. In Section 3, we describe the algorithm FW-AL and provide its analysis in Section 4. Finally, we present in Section 5 illustrative experiments.

2 Problem Setting

We will consider the following minimization problem,

$$\begin{aligned} &\underset{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(k)}}{\text{minimize}} \; f(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(k)})\,, \\ &\text{s.t. } \boldsymbol{x}^{(k)} \in \mathcal{X}_k,\; k \in [K],\; \sum_{k=1}^{K} A_k \boldsymbol{x}^{(k)} = 0\,, \end{aligned} \tag{OPT}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ is a convex differentiable function and for $k \in [K]$, $\mathcal{X}_k \subset \mathbb{R}^{d_k}$ are convex compact sets and A_k matrices of size $d \times d_k$. We will call the constraint $\sum_{k=1}^{K} A_k \boldsymbol{x}^{(k)} = 0$ the *linear consistency constraint*. We assume that we have access to the linear minimization oracle $\mathrm{LMO}_k(\boldsymbol{r}) \in \arg\min_{\boldsymbol{s}\in\mathcal{X}_k} \langle \boldsymbol{s}, \boldsymbol{r}\rangle$, $k \in [K]$. In Section 2.1 we detail some arising problem in machine learning and signal processing modeled by this framework. We denote by $\mathcal{X}^*$ the set of optimal points of the optimization problem (OPT) and we will assume that this problem is feasible, i.e., the set of solutions is non empty.

2.1 Motivating Applications

We now present some motivating applications of our problem setting, including examples where special case versions of FW-AL were used. This previous work provide additional evidence for the practical significance of the FW-AL algorithm.

Multiple sequence alignment and motif discovery (Yen et al., 2016a) are problems in which the domain is described as an intersection of alignment constraints and consensus constraints, two highly structured polytopes.

The linear oracle on both sets can be solved by dynamic programing whereas there is no known practical algorithm to project onto. A factorwise approach to the dual of the structured SVM objective (Yen et al., 2016b) can also be cast as constrained problem over a Cartesian product of polytopes related to each other by a linear consistency constraint. As often in structured prediction, the output domain grows exponentially, leading to extremely high dimensional polytopes. Once again, dynamic programming can be used to compute the linear oracle in structured SVMs at a lower computational cost than the potentially intractable projection. The algorithms proposed by Yen et al. (2016a) and Yen et al. (2016b) are in fact a particular instance of FW-AL, where the objective function is respectively linear and quadratic.

Finally, simultaneously sparse (ℓ_1 norm constraint) and low rank (trace norm constraint) matrices (Richard et al., 2012) is another class of problems where the constraints consists of an intersection of sets with simple linear oracle but expensive projection. This example is a novel application of FW-AL and is developed in Section 5.

2.2 Augmented Lagrangian Reformulation

It is possible to reformulate (OPT) into the problem of finding a saddle point of an augmented Lagrangian (Bertsekas, 1996), in order to split the constraints in a way in which the linear oracle is computed over a product space. We first rewrite (OPT) as follows:

$$\min_{\boldsymbol{x}^{(k)} \in \mathcal{X}_k,\, k \in [K]} f(\boldsymbol{x}) \quad \text{s.t.} \quad M\boldsymbol{x} = 0\,, \tag{1}$$

where $\boldsymbol{x} := \left(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)}\right)$, and M is such that

$$M\boldsymbol{x} = 0 \Leftrightarrow \sum_{k=1}^{K} A_k \boldsymbol{x}^{(k)} = 0\,. \tag{2}$$

We can now consider the augmented Lagrangian formulation of (1):

$$\begin{aligned} &\underset{(\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(K)})}{\text{minimize}} \ \max_{\boldsymbol{y} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)}, \boldsymbol{y}) \\ &\text{s.t.} \quad \boldsymbol{x}^{(k)} \in \mathcal{X}_k, \quad k \in \{1, \dots, K\} \\ &\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := f(\boldsymbol{x}) + \langle \boldsymbol{y}, M\boldsymbol{x} \rangle + \tfrac{\lambda}{2}\|M\boldsymbol{x}\|^2. \end{aligned} \tag{OPT2}$$

For notational simplicity, we denote $\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_K \subset \mathbb{R}^{d_1 + \ldots + d_K} = \mathbb{R}^m$. This formulation is the one onto which our algorithm FW-AL is applied.

Intersection of sets. One potential application of our work is the optimization over the intersection of an arbitrary number of sets $\cap_{k=1}^{K} \mathcal{X}_k$. In that case the matrix M of the ALM formulation is such that $M\boldsymbol{x} = 0 \Leftrightarrow \boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k+1)}$, $k \in [K-1]$.

Notation and assumption. In this paper, we denote by $\|\cdot\|$ the ℓ_2 norm and $\operatorname{dist}(\boldsymbol{x}, \mathcal{C}) := \inf_{\boldsymbol{x}' \in \mathcal{C}} \|\boldsymbol{x} - \boldsymbol{x}'\|$ its associated distance to a set. Note that in finite dimensional space this infimum is achieved when $\mathcal{C}$ is a non-empty closed set. We assume that f is L-smooth on $\mathbb{R}^p$, i.e., differentiable with L-Lipschitz continuous gradient. This is characterized by the following inequality for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^p$:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leqslant L\|\boldsymbol{x} - \boldsymbol{x}'\|\,. \tag{3}$$

This assumption is standard in convex optimization (Nesterov, 2004). Notice that the FW algorithm does not converge if the objective function is not at least continuously differentiable (Nesterov, 2016, Example 1). In our analysis, we will also use the observation that $\frac{\lambda}{2}\|M \cdot\|^2$ is generalized strongly convex.[1] We say that a function h is *generalized strongly convex* when it takes the following general form:

$$h(\boldsymbol{x}) := g(A\boldsymbol{x}) + \langle \boldsymbol{b}, \boldsymbol{x} \rangle\,, \quad \forall \boldsymbol{x} \in \mathbb{R}^p\,, \tag{4}$$

where $A \in \mathbb{R}^{d \times p}$ and g is μ_g-*strongly convex* w.r.t. the Euclidean norm on $\mathbb{R}^d$ with $\mu_g > 0$. Recall that a μ_g-strongly (differentiable) convex function $g : \mathbb{R}^d \to \mathbb{R}$ is one such that, $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$,

$$g(\boldsymbol{x}) \geqslant g(\boldsymbol{x}') + \left\langle \boldsymbol{x} - \boldsymbol{x}', \nabla g(\boldsymbol{x}') \right\rangle + \frac{\mu_g}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2\,.$$

3 FW-AL Algorithm

Our algorithm takes inspiration from both Frank-Wolfe and the augmented Lagrangian method. The augmented Lagrangian method alternates a primal update on $\boldsymbol{x}$ (approximately) minimizing[2] the augmented Lagrangian $\mathcal{L}(\cdot, \boldsymbol{y}_t)$, with a dual update on $\boldsymbol{y}$ by taking a gradient ascent step on $\mathcal{L}(\boldsymbol{x}_{t+1}, \cdot)$. The FW-AL algorithm follows the general iteration of the augmented Lagrangian method, but with the crucial difference that Lagrangian minimization is replaced by one Frank-Wolfe step on $\mathcal{L}(\cdot, \boldsymbol{y}_t)$. The algorithm is thus composed by two loops: an outer loop presented in (5) and an inner loop noted $\mathcal{FW}$ which can be chosen to be one of the FW step variants described in Alg. 1 or 2.

[1] This notion has been studied by Wang and Lin (2014) and in the Frank-Wolfe framework by Beck and Shtern (2016) and Lacoste-Julien and Jaggi (2015).

[2] An example of approximate minimization is taking one proximal gradient step, as used for example in the Linearized ADMM algorithm (Goldfarb et al., 2013; Yang and Yuan, 2013).

FW Augmented Lagrangian method (FW-AL)

At each iteration $t \geqslant 1$, we first update the primal variable blocks $\boldsymbol{x}_t$ with a Frank-Wolfe step and then update the dual multiplier $\boldsymbol{y}_t$ using the updated primal variable:

$$\begin{cases} \boldsymbol{x}_{t+1} = \mathcal{FW}(\boldsymbol{x}_t; \mathcal{L}(\cdot, \boldsymbol{y}_t)) \,, \\ \boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \eta_t M \boldsymbol{x}_{t+1} \,, \end{cases} \tag{5}$$

where $\eta_t > 0$ is the step size for the dual update and $\mathcal{FW}$ is either Alg. 1 or Alg. 2 (see more in App. A).

FW steps. In FW-AL we need to ensure that the $\mathcal{FW}$ inner loop makes sufficient progress. For general sets, we can use one iteration of the classical Frank-Wolfe algorithm with line-search (Jaggi, 2013) as given in Algorithm 2. When working over polytopes, we can get faster (linear) convergence by taking one *non-drop* step (defined below) of the away-step variant of the FW algorithm (AFW) (Lacoste-Julien and Jaggi, 2015), as described in Algorithm 1). Other possible variants are discussed in Appendix A. We denote by $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ the iterates computed by FW-AL after t steps and by $\mathcal{A}_t$ the set of atoms previously given by the FW oracle (including the initialization point). If the constraint set is the convex hull of a set of atoms $\mathcal{A}$, the iterate $\boldsymbol{x}_t$ has a sparse representation as a convex combination of the first iterate and the atoms previously given by the FW oracle. The set of atoms which appear in this expansion with non-zero weight is called the *active set* $\mathcal{S}_t$. Similarly, since $\boldsymbol{y}_t$ is by construction in the cone generated by $\{M\boldsymbol{x}_s\}_{s \leqslant t}$, the iterate $\boldsymbol{y}_t$ is in the span of $M\mathcal{A}_t$, that is, they both have the sparse expansion:

$$\boldsymbol{x}_t = \sum_{\boldsymbol{v} \in \mathcal{S}_t} \alpha_{\boldsymbol{v}}^{(t)} \boldsymbol{v}, \quad \text{and} \quad \boldsymbol{y}_t = \sum_{\boldsymbol{v} \in \mathcal{A}_t} \xi_{\boldsymbol{v}}^{(t)} M\boldsymbol{v} \,, \tag{6}$$

When we choose to use the AFW Alg. 1 as inner loop algorithm, it can choose an *away* direction to remove mass from "bad" atoms in the active set, i.e. to reduce $\alpha_{\boldsymbol{v}}^{(t)}$ for some $\boldsymbol{v}$ (see L11 of Alg. 1), thereby avoiding the zig-zagging phenomenon that prevents FW from achieving a better convergence rate (Lacoste-Julien and Jaggi, 2015). On the other hand, the maximal step size for an *away* step can be quite small ($\gamma_{\max} = \alpha_{\boldsymbol{v}}^{(t)}/1-\alpha_{\boldsymbol{v}}^{(t)}$, where $\alpha_{\boldsymbol{v}}^{(t)}$ is the weight of the away vertex in (6)), yielding to arbitrary small suboptimality progress when the line-search is truncated to such small step-sizes. A step removing an atom from the active set is called a *drop step* (this is further discussed in Appendix A), and Alg. 1 loops until a non-drop step is obtained. In App. A.1 we prove that the number of drop steps after t iterations cannot be larger than $t+1$. This bound is crucial in order to make the convergence results theoretically significant. Indeed, if we were not able to upper bound the cumulative number of drop steps, FW-AL with Alg. 1 as inner loop could be stuck (since Alg. 1 only returns when it performs a non-drop step).

Algorithm 1 Away-step Frank-Wolfe (one non-drop step) : (Lacoste-Julien and Jaggi, 2015)

input: $(\boldsymbol{x}, \mathcal{S}, \mathcal{A}, \varphi)$
`drop_step` $\leftarrow$ `true` (initialization of the boolean)
while `drop_step` = `true` **do**
  $\boldsymbol{s} \leftarrow \text{LMO}\left(\nabla\varphi(\boldsymbol{x})\right)$
  $\boldsymbol{v} \in \arg\max_{\boldsymbol{v} \in \mathcal{S}} \langle \nabla\varphi(\boldsymbol{x}), \boldsymbol{v} \rangle$
  $g^{FW} \leftarrow \langle \nabla\varphi(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{s} \rangle$ *(Frank-Wolfe gap)*
  $g^{A} \leftarrow \langle \nabla\varphi(\boldsymbol{x}), \boldsymbol{v} - \boldsymbol{x} \rangle$ *(Away gap)*
  if $g^{FW} \geqslant g^{A}$ **then** *(FW direction is better)*
    $\boldsymbol{d} \leftarrow \boldsymbol{s} - \boldsymbol{x}$ and $\gamma_{\max} \leftarrow 1$
  else *(Away direction is better)*
    $\boldsymbol{d} \leftarrow \boldsymbol{x} - \boldsymbol{v}$ and $\gamma_{\max} \leftarrow \alpha_{\boldsymbol{v}}/(1 - \alpha_{\boldsymbol{v}})$
  end if
  Compute $\gamma \in \arg\min_{\gamma \in [0, \gamma_{\max}]} \varphi\left(\boldsymbol{x} + \gamma \boldsymbol{d}\right)$
  if $\gamma < \gamma_{\max}$ **then** *(first non-drop step)*
    `drop_step` $\leftarrow$ `false`
  end if
  Update $\boldsymbol{x} \leftarrow \boldsymbol{x} + \gamma \boldsymbol{d}$
  Update $\alpha_{\boldsymbol{v}}$ according to (6)
  Update $\mathcal{S} \leftarrow \{\boldsymbol{v} \in \mathcal{A} \text{ s.t. } \alpha_{\boldsymbol{v}} > 0\}$ *(active set)*
end while
return: $(\boldsymbol{x}, \mathcal{S})$

Algorithm 2 FW(one step) : (Frank and Wolfe, 1956)

input: $(\boldsymbol{x}, \varphi)$
Compute $\boldsymbol{r} \leftarrow \nabla\varphi(\boldsymbol{x})$
Compute $\boldsymbol{s} \leftarrow \arg\min_{\boldsymbol{s} \in \mathcal{X}} \langle \boldsymbol{s}, \boldsymbol{r} \rangle$
$\gamma \in \arg\min_{\gamma \in [0,1]} \varphi(\boldsymbol{x} + \gamma(\boldsymbol{s} - \boldsymbol{x}))$
Update $\boldsymbol{x} \leftarrow (1 - \gamma)\boldsymbol{x} + \gamma \boldsymbol{s}$
return: $\boldsymbol{x}$

4 Analysis of FW-AL

Solutions of (OPT2) are called saddle points, equivalently a vector $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathcal{X} \times \mathbb{R}^d$ is said to be a saddle point if the following is verified for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathbb{R}^d$,

$$\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}) \leqslant \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*) \leqslant \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^*) \,. \tag{7}$$

Our assumptions (convexity of f and $\mathcal{X}$, feasibility of $M\boldsymbol{x} = 0$, and crucially boundedness of $\mathcal{X}$) are sufficient for strong duality to hold (Boyd and Vandenberghe, 2004, Exercise 5.25(e)). Hence, the set of saddle points is not empty and is equal to $\mathcal{X}^* \times \mathcal{Y}^*$, where $\mathcal{X}^*$ is the set of minimizer of (OPT) and $\mathcal{Y}^*$ the set of maximizers of the augmented dual function d:

$$d(\boldsymbol{y}) := \min_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) \,. \tag{8}$$

One of the issue of ALM is that it is a non feasible method and consequently the function suboptimality is no longer a satisfying convergence criterion (since it can be negative). In the following section we wonder what could be the quantities to look at in order to get a sufficient condition of convergence.

4.1 Convergence Measures

Variants of ALM (also known as the methods of multipliers) update at each iteration both the primal variable and the dual variable. For the purpose of analyzing the popular ADMM algorithm, Hong and Luo (2017) introduced two positive quantities which they called primal and dual gaps that we re-use in the analysis of our algorithm. Let $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ be the current primal and dual variables after t iterations of the FW-AL algorithm (5), the dual gap is defined as

$$\Delta_t^{(d)} := d^* - d(\boldsymbol{y}_t) \quad \text{where} \quad d(\boldsymbol{y}_t) := \min_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}_t) \quad (9)$$

and $d^* := \max_{\boldsymbol{y}\in\mathbb{R}^d} d(\boldsymbol{y})$. It represents the dual suboptimality at the t-th iteration. On the other hand, the "primal gap" at iteration t is defined as

$$\Delta_t^{(p)} := \mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - d(\boldsymbol{y}_t), \quad t \geqslant 0\,. \quad (10)$$

Notice that $\Delta_t^{(p)}$ is *not* the suboptimality associated with the primal function $p(\cdot) := \max_{\boldsymbol{y}\in\mathbb{R}^d} \mathcal{L}(\cdot, \boldsymbol{y})$ (which is infinite for every $\boldsymbol{x}$ non feasible). In this paper, we also define

$$\Delta_t := \Delta_t^{(p)} + \Delta_t^{(d)}\,. \quad (11)$$

It is important to realize that since ALM is a non-feasible method, the standard convergence convex minimization certificates could become meaningless. In particular, the quantity $f(\boldsymbol{x}_t) - f^*$ might be negative since $\boldsymbol{x}_t$ does not necessarily belong to the constraint set of (OPT). Without any additional assumption on f, the primal and dual gap introduced in (9) and (10) can be small whereas the current iterate can be arbitrary far from the optimal point.

Illustrative example. Previous theoretical results for GDMM (Yen et al., 2016a,b; Huang et al., 2017) only provided a rate on both gaps (9) and (10) which is not sufficient to derive guarantees on how close is iterate to the optimal point. In this paper, we are able to prove that the feasibility $\|M\boldsymbol{x}\|^2$ converges to 0 as fast as Δ_t. But even with these quantities vanishing, the suboptimality of the closest feasible point can be arbitrary bigger than the suboptimality of a point ϵ-close to the optimum. To illustrate this point, we will propose an objective function and two constraint sets following the intuition that a small value of $f(\boldsymbol{x}_t) - f^*$ can allow the point $\boldsymbol{x}_t$ to be really far from $\boldsymbol{x}^*$ where the gradient of f is larger that around the optimum, implying that feasible points close to $\boldsymbol{x}_t$ can have large suboptimality. Concretely, let $r \gg \epsilon > 0$ two fixed variables and let,

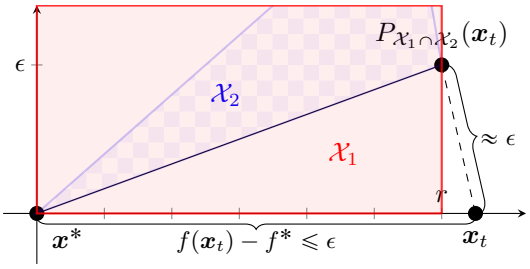


Figure 1: Even with $f(\boldsymbol{x}_t) - f^*$ small, an $\boldsymbol{x}_t$ far from $\boldsymbol{x}^*$ can lead to large value for $f(P_{\mathcal{X}_1\cap\mathcal{X}_2}(\boldsymbol{x}_t)) - f^*$

$$f(u,v) = \epsilon\tfrac{u^2 e^{r^4 v}}{(r+\epsilon)^2}, \quad \mathcal{X}_1 := [0,r]^2 \quad \text{and}$$
$$\mathcal{X}_2 := \{(u,v)\,;\, \|(u,v)\|_1 \leqslant r+\epsilon\,;\, v \geqslant \tfrac{\epsilon}{r}u,\ v \leqslant u\}\,.$$

The sets are compact and the function f is Lipschitz on these sets. The sets and the position of the non feasible iterate $\boldsymbol{x}_t$ are represented in Figure 1. We clearly have $f^* = 0$ and $\boldsymbol{x}^* = (0,0)$. We set $\boldsymbol{x}_t := (0, r')$ such that its projection $P_{\mathcal{X}_1\cap\mathcal{X}_2}(\boldsymbol{x}_t)$ is (r,ϵ), explicitly we can show that $r' = r + \epsilon^2/r$. We then have

$$f(\boldsymbol{x}_t) \leqslant \epsilon, \quad \text{and} \quad f(P_{\mathcal{X}_1\cap\mathcal{X}_2}(\boldsymbol{x}_t)) = \epsilon\tfrac{r^2 e^{r^4\epsilon}}{(r+\epsilon)^2} \geqslant \tfrac{\epsilon}{2} e^{r^4\epsilon}\,,$$

whereas $\forall \boldsymbol{x} \in B(\boldsymbol{x}^*, \epsilon)$, $f(\boldsymbol{x}) \leqslant \frac{\epsilon^3 e^{r^4\epsilon/2}}{r^2}$ which can be arbitrary smaller than $f(P_{\mathcal{X}_1\cap\mathcal{X}_2}(\boldsymbol{x}_t))$.

In conclusion, a rate on Δ_t and on the feasibility can lead to a bad approximate solution for the original problem (OPT). In order to obtain a convergence certificate leading to a "good" approximate solution we will also consider strongly convex functions. With this additional assumption, we proved a convergence rate on the distance of the iterates to the solution of (OPT).

4.2 Properties of the augmented Lagrangian dual function

The augmented dual function plays a key role in our convergence analysis. One of our main technical contribution is the proof of a new property of this function which can be understood as a growth condition. This property is due to the smoothness of the objective function and the compactness of the constraint set. We will need an additional technical assumption called *interior qualification* (a.k.a *Slater's conditions*).

Assumption 1. *The sets $(\mathcal{X}_k)$ are convex compact sets such that there exist an interior feasible point, i.e., there exist $\boldsymbol{x}^{(k)} \in \operatorname{relint}(\mathcal{X}_k)$, $k \in [K]$ such that $\sum_{k=1}^K A_k \boldsymbol{x}^{(k)} = 0$.*

Recall that $\boldsymbol{x} \in \operatorname{relint}(\mathcal{X})$ if and only if $\boldsymbol{x}$ is an interior point relative to the affine hull of $\mathcal{X}$. This assumption is pretty standard and weak in practice. It is a particular case of constraint qualifications (Holmes, 1975; Gowda and Teboulle, 1990). With this assumption we can deduce a global property on the augmented dual function that can be summarized as quadratic growth condition on a ball of size $L_\lambda D^2$ and a linear growth condition outside of this ball. Optimization literature named such properties *error bounds* (Pang, 1997).

Theorem 1 (Error bound)**.** *Let d be the augmented dual function (8), if f is a L-smooth convex function, $\mathcal{X}$ a compact convex set and if Assump. 1 holds, then there exist a constant $\alpha > 0$ such that for all $\boldsymbol{y} \in \mathbb{R}^d$,*

$$d^* - d(\boldsymbol{y}) \geqslant \frac{\alpha^2}{2} \min\left\{ \frac{\operatorname{dist}(\boldsymbol{y}, \mathcal{Y}^*)^2}{L_\lambda D^2}, \operatorname{dist}(\boldsymbol{y}, \mathcal{Y}^*) \right\}, \quad (12)$$

where $D := \max_{(\boldsymbol{x},\boldsymbol{x}') \in \mathcal{X}^2} \|\boldsymbol{x} - \boldsymbol{x}'\|$ is the diameter of $\mathcal{X}$ and $L_\lambda := L + \lambda \|M^\top M\|$.

This theorem, proven in App. C.1, is crucial in our analysis. In our *descent lemma* (21), we want to relate the gap decrease to a quantity proportional to the gap. A consequence of (12) is a lower bound of interest: (22).

Issue in previous proofs. In previous work, Yen et al. (2016a, Theorem 2) have a constant called R_Y in the upper bound of Δ_t which may be infinite and lead to the trivial bound $\Delta_t \leqslant \infty$. Actually R_Y is an upper bound on the distance of the dual iterate $\boldsymbol{y}_t$ to the optimal solution set $\mathcal{Y}^*$ of the augmented dual function, since this quantity is not proven to be bounded an element is missing in the convergence analysis. In their convergence proof, Yen et al. (2016b) and Huang et al. (2017) use Lemma 3.1 in (Hong and Luo, 2012) (which also appears as Lemma 3.1 in the published version (Hong and Luo, 2017)). This lemma states a result not holding for all $\boldsymbol{y} \in \mathbb{R}^d$ but instead for $(\boldsymbol{y}_t)_{t \in \mathbb{N}}$, which is the sequence of dual variables computed by the ADMM algorithm used in (Hong and Luo, 2017). This sequence cannot be assimilated to the sequence of dual variables computed by the GDMM algorithm since the update rule for the primal variables in each algorithm is different, the primal variable are updated with FW steps in one algorithm and with a proximal step in the other. The properties of this proximal step are intrinsically different from the FW steps computing the updates on the primal variables of FW-AL. To our knowledge, there is no easy fix to get a result as the one claimed by Yen et al. (2016b, Lem. 4) and Huang et al. (2017, Lem. 4). More details are provided in App. B.1.

4.3 Specific analysis for FW-AL

Convergence over general convex sets. When $\mathcal{X}$ is a general convex compact set and f is L-smooth, Algorithms 1 and 2 are able to perform a decrease on the objective value proportional to the square of the suboptimality (Jaggi, 2013, Lemma 5), (Lacoste-Julien and Jaggi, 2015, (31)), we will call this a *sublinear decrease* since it leads to a sublinear rate for the suboptimality: for any $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{y} \in \mathbb{R}^d$ they compute $\boldsymbol{x}^+ := \mathcal{FW}(\boldsymbol{x}; \mathcal{L}(\cdot, \boldsymbol{y}))$, such that for all $\gamma \in [0, 1]$,

$$\mathcal{L}(\boldsymbol{x}^+, \boldsymbol{y}) - \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) \leqslant \gamma(d(\boldsymbol{y}) - \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})) + \frac{\gamma^2 L_\lambda D^2}{2}, \quad (13)$$

where L_λ is the Lipschitz constant of $\mathcal{L}$ and D the diameter of $\mathcal{X}$. Recall that $d(\boldsymbol{y}) := \min_{\boldsymbol{x}' \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}', \boldsymbol{y})$. Note that setting $\gamma = 0$ provides us $\mathcal{L}(\boldsymbol{x}^+, \boldsymbol{y}) \leqslant \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$ and optimizing the RHS respect to γ provides us a decrease proportional to $(d(\boldsymbol{y}) - \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}))^2$. The GDMM algorithm of (Yen et al., 2016a,b; Huang et al., 2017) relied on the assumption of $\mathcal{X}$ being polytope, hence we obtain from this sublinear decrease a completely new result on ALM with FW. This result covers the case of the simultaneously sparse and low rank matrices (29) where the trace norm ball is not a polytope.

Theorem 2 (Rate of FW-AL with Alg. 2)**.** *Under Assumption 1, if $\mathcal{X}$ is a convex compact set and f is a L-smooth convex function and M has the form described in (2), then using any algorithm with* sublinear decrease *(13) as inner loop in* FW-AL *(5) and $\eta_t := \min\left\{\frac{2}{\lambda}, \frac{\alpha^2}{2\delta}\right\} \frac{2}{t+2}$ we have that there exists a bounded $t_0 \geqslant 0$ such that $\forall t \geqslant t_0$,*

$$\Delta_t \leqslant \frac{4\delta(t_0 + 2)}{t + 2}, \quad \min_{t_0 \leqslant s-1 \leqslant t} \|M\boldsymbol{x}_s\|^2 \leqslant \frac{O(1)}{t - t_0 + 1} \quad (14)$$

where $D := \max_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|$ is the diameter of $\mathcal{X}$, $L_\lambda := L + \lambda \|M^\top M\|$ the Lipschitz constant of $\mathcal{L}$ the AL function, $\delta := L_\lambda D^2$ and α is defined in Thm.1.

In App. D.2 we provide an analysis for different step size schemes and explicit bounds on t_0.

Convergence over Polytopes. On the other hand, if $\mathcal{X}$ is a polytope, recent advances on FW proposed global linear convergence rates for a generalized strongly convex objective using FW with away steps (Lacoste-Julien and Jaggi, 2015; Garber et al., 2016). Note that in the augmented formulation, $\lambda > 0$ and thus $\frac{1}{2}\|M \cdot\|^2$ is a generalized strongly convex function, making $\mathcal{L}(\cdot, \boldsymbol{y})$ a generalized strongly convex function for any $\boldsymbol{y} \in \mathbb{R}^d$. We can then use such linearly convergent algorithms to improve the rate of FW-AL. More precisely, we will use the fact that Algorithm 1 performs *geometric decrease* (Lacoste-Julien and Jaggi, 2015, Theorem 1): for $\boldsymbol{x}^+ := \mathcal{FW}(\boldsymbol{x}; \mathcal{L}(\cdot, \boldsymbol{y}))$, there exists $\rho_A < 1$ such that for all $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathbb{R}^d$,

$$\mathcal{L}(\boldsymbol{x}^+, \boldsymbol{y}) - \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) \leqslant \rho_A \big[\min_{\boldsymbol{x}' \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}', \boldsymbol{y}) - \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) \big]. \quad (15)$$

The constant ρ_A (Lacoste-Julien and Jaggi, 2015) depends on the smoothness, the generalized strong convexity of $\mathcal{L}(\cdot, \boldsymbol{y})$ (does not depend on $\boldsymbol{y}$, but depends on M) and the condition number of the set $\mathcal{X}$ depending on its geometry (more details in App. A.3).

Theorem 3 (Rate of FW-AL with inner loop Alg. 1)**.** *Under the same assumptions as in Thm.2 and if moreover $\mathcal{X}$ is a polytope and f a generalized strongly convex function, then using Alg 1 as inner loop and a constant step size $\eta_t = \frac{\lambda \rho_A}{4}$, the quantity Δ_t decreases by a uniform amount for finite number of steps t_0 as,*

$$\Delta_{t+1} - \Delta_t \leqslant -\frac{\lambda \alpha^2 \rho_A}{8} \,, \tag{16}$$

until $\Delta_{t_0} \leqslant L_\lambda D^2$. Then for all $t \geqslant t_0$ we have that the gap and the feasibility violation decrease linearly as,

$$\Delta_t \leqslant \frac{\Delta_{t_0}}{(1+\kappa)^{t-t_0}} \,, \quad \|M\boldsymbol{x}_{t+1}\|^2 \leqslant \frac{16}{\lambda \cdot \rho_A} \frac{\Delta_{t_0}}{(1+\kappa)^{t-t_0}} \,,$$

where $\kappa := \min\left\{\frac{\rho_A}{2}, \frac{\rho_A \lambda \alpha^2}{8 L_\lambda D^2}\right\}$ and $L_\lambda := L + \lambda \|M^\top M\|$.

Strongly convex functions. When the objective function f is strongly convex, we are able to give a convergence rate for the distance of the primal iterate to the optimum. As argued in Sec. 4.1, an iterate close to the optimal point lead to a "better" approximate solution than an iterate achieving a small gap value.

Theorem 4. *Under the same assumptions as in Thm. 2, if f is a μ-strongly convex function, then the set of optimal solutions $\mathcal{X}^*$ is reduced to $\{\boldsymbol{x}^*\}$ and,*

$$\min_{8t_0+15 \leqslant s \leqslant t} \|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 \leqslant \frac{4K}{\mu} \frac{O(1)}{t - 8t_0 - 14} \,. \tag{17}$$

Moreover if $\mathcal{X}$ is a compact polytope the distance of the current point to the optimal set vanishes as,

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2 \leqslant \frac{2K\Delta_{t_0}(\sqrt{2}+1)}{\mu(\sqrt{1+\kappa})^{t-t_0}} \,. \tag{18}$$

For an intersection of sets, the three theorems above give stronger results than (Yen et al., 2016b; Huang et al., 2017) since we prove that the distance to the optimal point as well as the feasibility condition vanish linearly.

Proof sketch of Thm 2 and 3 Our goal is to obtain a convergence rate on the sum gaps (9) and (10). First we show that the dual gap verifies

$$\Delta^{(d)}_{t+1} - \Delta^{(d)}_t \leqslant -\eta_t \langle M\boldsymbol{x}_{t+1}, M\hat{\boldsymbol{x}}_{t+1}\rangle \tag{19}$$

where $\hat{\boldsymbol{x}}_{t+1} := \arg\min_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}_{t+1})$. Similarly, we prove the following inequality for the primal gap

$$\begin{aligned}\Delta^{(p)}_{t+1} - \Delta^{(p)}_t \leqslant \eta_t \|M\boldsymbol{x}_{t+1}\|^2 \\ + \mathcal{L}(\boldsymbol{x}_{t+2}, \boldsymbol{y}_{t+1}) - \mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) \\ - \eta_t \langle M\boldsymbol{x}_{t+1}, M\hat{\boldsymbol{x}}_{t+1}\rangle \,.\end{aligned} \tag{20}$$

Summing (19) and (20) and using that $\|M\boldsymbol{x}_{t+1} - M\hat{\boldsymbol{x}}_{t+1}\|^2 \leqslant \frac{2}{\lambda}\left(\mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) - \mathcal{L}(\hat{\boldsymbol{x}}_{t+1}, \boldsymbol{y}_{t+1})\right)$, we get the following *fundamental descent lemma*,

$$\begin{aligned}\Delta_{t+1} - \Delta_t \leqslant \mathcal{L}(\boldsymbol{x}_{t+2}, \boldsymbol{y}_{t+1}) - \mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) \\ + \frac{2\eta_t}{\lambda}\left(\mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) - \mathcal{L}(\hat{\boldsymbol{x}}_{t+1}, \boldsymbol{y}_{t+1})\right) \\ - \eta_t \|M\hat{\boldsymbol{x}}_{t+1}\|^2 \,.\end{aligned} \tag{21}$$

We now crucially combine (12) in Thm. 1 and the fact that $\Delta^{(d)}_t \leqslant \operatorname{dist}(\boldsymbol{y}^t, \mathcal{Y}^*)\|M\hat{\boldsymbol{x}}_{t+1}\|$ to obtain,

$$\tfrac{\alpha^2}{2L_\lambda D^2} \min\{\Delta^{(d)}_{t+1}, L_\lambda D^2\} \leqslant \|M\hat{\boldsymbol{x}}_{t+1}\|^2 \,, \tag{22}$$

and then,

$$\begin{aligned}\Delta_{t+1} - \Delta_t \leqslant \mathcal{L}(\boldsymbol{x}_{t+2}, \boldsymbol{y}_{t+1}) - \mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) \\ + \frac{2\eta_t}{\lambda}\left(\mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) - \mathcal{L}(\hat{\boldsymbol{x}}_{t+1}, \boldsymbol{y}_{t+1})\right) \\ - \eta_t \tfrac{\alpha^2}{2L_\lambda D^2} \min\{\Delta^{(d)}_{t+1}, L_\lambda D^2\} \,.\end{aligned} \tag{23}$$

Now the choice of the algorithm to get $\boldsymbol{x}_{t+2}$ from $\boldsymbol{x}_{t+1}$ and $\boldsymbol{y}_{t+1}$ is decisive:

If $\mathcal{X}$ is a polytope and if an algorithm with a *geometric decrease* (15) is used, setting $\eta_t = \frac{\lambda \cdot \rho_A}{4}$ we obtain

$$\begin{aligned}\Delta_{t+1} - \Delta_t \leqslant -\frac{\rho_A}{2}\left(\mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) - \mathcal{L}(\hat{\boldsymbol{x}}_{t+1}, \boldsymbol{y}_{t+1})\right) \\ - \frac{\lambda \cdot \rho_A}{4}\|M\boldsymbol{x}_{t+1}\|^2 \,.\end{aligned}$$

Since $\mathcal{L}(\boldsymbol{x}_{t+2}, \boldsymbol{y}_{t+1}) \leqslant \mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1})$ (L13), we have

$$\Delta^{(p)}_{t+1} \leqslant \mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1}) - \mathcal{L}(\hat{\boldsymbol{x}}_{t+1}, \boldsymbol{y}_{t+1}) \,, \tag{24}$$

leading us to a geometric decrease for all $t \geqslant t_0$,

$$\Delta_{t+1} \leqslant \frac{\Delta_t}{1+\kappa} \quad \text{where} \quad \kappa := \tfrac{\rho_A}{2} \min\left\{1, \tfrac{\lambda \alpha^2}{8 L_\lambda D^2}\right\}. \tag{25}$$

Additionally we can deduce from (21) that,

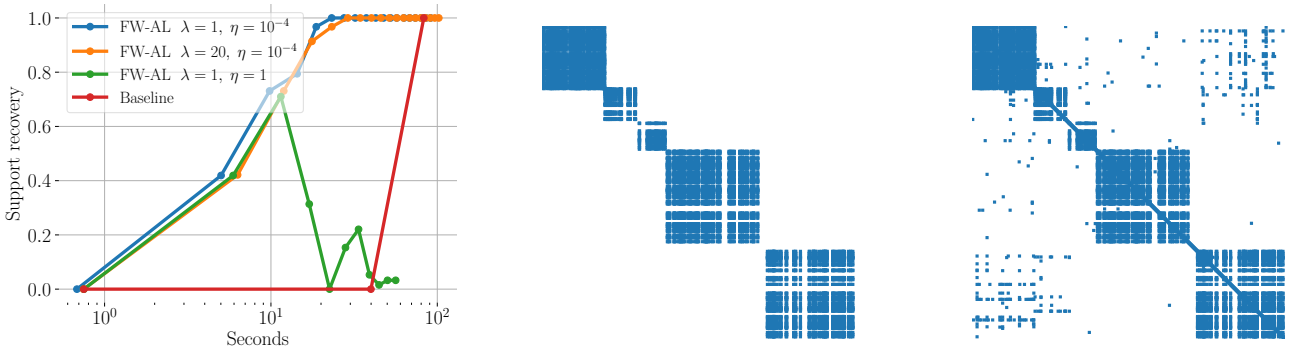
$$\eta_t \|M\hat{\boldsymbol{x}}_{t+1}\|^2 \leqslant \Delta_t \quad \text{and} \quad \eta_t \|M\boldsymbol{x}_{t+1}\|^2 \leqslant 4\Delta_t \,. \tag{26}$$

If $\mathcal{X}$ is not a polytope, we can use an algorithm with a *sublinear decrease* (13) to get from (23) that $\forall t \geqslant 0$,

$$\Delta_{t+1} - \Delta_t \leqslant -a\eta_t \min\{\Delta_{t+1}, \delta\} + (a\eta_t)^2 \tfrac{C}{2} \,, \tag{27}$$

where a, δ and C are three positive constants. Setting $\eta_t = \frac{2}{a(t+2)}$ we can prove that there exists $t_0 \geqslant \frac{C}{\delta}$ s.t.,

$$\Delta_{t+1} \leqslant \frac{4\delta(2+t_0)}{(t+2)}, \quad \forall t \geqslant t_0 \,. \tag{28}$$



(a) Fraction of the support recovered. (b) Matrix recovered with FW-AL. (c) Matrix recov. with the baseline.

Figure 2: Fig. 2a represent the fraction of the support of Σ recovered ($d^2 = 1.6 \cdot 10^7$ and the matrix computed is thresholded at 10^{-2}). The baseline is the generalized forward backward algorithm. Fig 2b and 2c represent the matrices recovered for $d^2 = 10^6$ after one minute of computation.

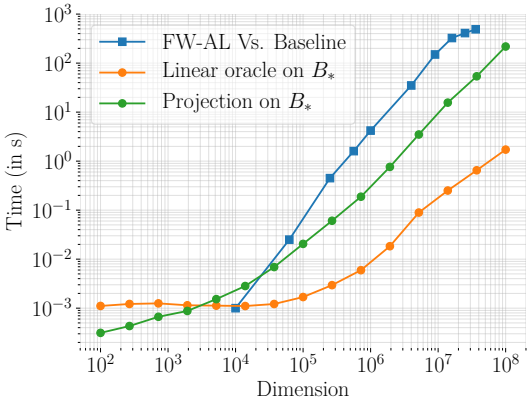


Figure 3: Time complexity of the linear oracle and the projection on the trace norm ball. The blue curve represent time spent by the generalized forward backward algorithm to reach a better point than the one computed by FW-AL.

5 Illustrative Experiments

Recovering a matrix that is simultaneously low rank and sparse has applications in problems such as covariance matrix estimation, graph denoising and link prediction (Richard et al., 2012). We compared our algorithm with proximal splitting method on a covariance matrix estimation problem. We define the $\|\cdot\|_1$ norm of a matrix S as $\|S\|_1 := \sum_{i,j} |S_{i,j}|$ and its trace norm as $\|S\|_* := \sum_{i=1}^{\operatorname{rank}(S)} \sigma_i$, where σ_i are the singular values of S. Given a symmetric positive definite matrix $\hat{\Sigma} \in \mathbb{R}^{d\times d}$ the objective function is defined as

$$\min_{S\succeq 0, \|S\|_1 \leqslant \beta_1, \|S\|_* \leqslant \beta_2} \|S - \hat{\Sigma}\|_2^2 \,. \tag{29}$$

The linear oracle for $\mathcal{X}_1 := \{S \succeq 0,\ \|S\|_1 \leqslant \beta_1\}$ is

$$\operatorname{LMO}_{\mathcal{X}_1}(D) := \beta_1 \tfrac{E_{ij}+E_{ji}}{2},\ (i,j) \in \operatorname*{arg\,min}_{(i,j)\in d\times d} D_{i,j} + D_{j,i}$$

where (E_{ij}) is the standard basis of $\mathbb{R}^{d\times d}$. The linear oracle for $\mathcal{X}_2 := \{S \succeq 0,\ \|S\|_* \leqslant \beta_2\}$ is

$$\operatorname{LMO}_{\mathcal{X}_2}(D) := \beta_2 \cdot U_1^\top U_1 \,, \tag{30}$$

where $D = [U_1, \dots, U_d] \operatorname{diag}(\sigma_1, \dots, \sigma_d) [U_1, \dots, U_d]^\top$. It can be computed efficiently by the Lanczos algorithm (Paige, 1971; Kuczyński and Woźniakowski, 1992) whereas the standard splitting method to solve (29) requires to compute projections over the trace norm ball via a complete diagonalization which is $O(d^3)$. For large d, the full diagonalization becomes untractable, while the Lanczos algorithm is more scalable and requires less storage (see Fig. 3).

The experimental setting is done following Richard et al. (2012): we generated a block diagonal covariance matrix Σ to draw n vectors $\boldsymbol{x}_i \sim \mathcal{N}(0, \Sigma)$. We use 5 blocks of the form $\boldsymbol{v}\boldsymbol{v}^\top$ where $\boldsymbol{v} \sim \mathcal{U}([-1, 1])$. In order to enforce sparsity we only kept the entries (i, j) such that $|\Sigma_{i,j}| > .9$. Finally, we add a gaussian noise $\mathcal{N}(0, \sigma)$ on each entry $\boldsymbol{x}_i$ and observe $\hat{\Sigma} = \sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^T$. In our experiment $n = d, \sigma = 0.6$. We apply our method, as well as the baseline from (Richard et al., 2012), which is the generalized forward backward splitting of Raguet et al. (2013), to optimize (29) performing projections over the constraint sets. The results are presented in Fig. 2 and 3 . The oracle for this algorithm is also slower in the large scale, as can be seen in Fig. 3. We can say that our algorithm performs better than the baseline for high dimensional problems for two reasons: in high dimensions, only one projection on the trace norm ball B_* can take hours (green curve) whereas solving a LMO over B_* takes few seconds, additionally, the iterates computed by FW-AL are naturally sparse and low rank, so we then get a better estimation at the beginning of the optimization as illustrated in Fig. 2b and 2c.

Acknowledgements

Thanks to an anonymous reviewer for helpful comments. Work partially supported by the European Union's Horizon 2020 research and innovation program under the Marie Skłodorowska-Curie grant agreement 748900.

References

J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016.

A. Beck and S. Shtern. Linearly convergent away-step conditional gradient for non-strongly convex functions. *Math. Program.*, 2016.

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

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 2011.

J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010.

D. J. M. Danskin. The directional derivative. In *The Theory of Max-Min and Its Application to Weapons Allocation Problems*. Springer Berlin Heidelberg, 1967.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics*, 1956.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 1976.

D. Garber, D. Garber, and O. Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *NIPS*, 2016.

R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis*, 1975.

R. Glowinski, S. J. Osher, and W. Yin. *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, 2017.

D. Goldfarb, S. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. *Mathematical Programming*, 2013.

M. S. Gowda and M. Teboulle. A comparison of constraint qualifications in infinite-dimensional convex programming. *SIAM Journal on Control and Optimization*, 1990.

A. Gramfort, B. Thirion, and G. Varoquaux. Identifying predictive regions from fMRI with TV-L1 prior. In *International Workshop on Pattern Recognition in Neuroimaging*. IEEE, 2013.

R. B. Holmes. Geometric functional analysis and its applications. 1975.

M. Hong and Z.-Q. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv:1208.3922*, 2012.

M. Hong and Z.-Q. Luo. On the linear convergence of the alternating direction method of multipliers. *Math. Program.*, 2017.

X. Huang, I. E.-H. Yen, R. Zhang, Q. Huang, P. Ravikumar, and I. Dhillon. Greedy direction method of multiplier for MAP inference of large output domain. In *AISTATS*, 2017.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.

S. Kakade, S. Shalev-Shwartz, and A. Tewari. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript*, 2009.

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. *ECML*, 2016.

J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM. J. Matrix Anal. & Appl.*, 1992.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. 2015.

S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *ICML*, 2013.

S. Łojasiewicz. A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 1963.

Y. Nesterov. *Introductory Lectures on Convex Optimization*. Applied Optimization. Springer US, 2004.

Y. Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. *CORE Discussion Paper*, 2016.

G. Obozinski, L. Jacob, and J.-P. Vert. Group lasso with overlaps: the latent group lasso approach. *arXiv:1110.0413*, 2011.

C. C. Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices.* PhD thesis, University of London, 1971.

J.-S. Pang. A posteriori error bounds for the linearly-constrained variational inequality problem. *Mathematics of Operations Research*, 1987.

J.-S. Pang. Error bounds in mathematical programming. *Math. Program.*, 1997.

B. T. Polyak. Gradient methods for minimizing functionals. *Zh. Vychisl. Mat. Mat. Fiz.*, 1963.

H. Raguet, J. Fadili, and G. Peyré. A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 2013.

E. Richard, P.-A. Savalle, and N. Vayatis. Estimation of simultaneously sparse and low rank matrices. In *ICML*, 2012.

R. T. Rockafellar and R. J. Wets. Variational analysis. 1998.

E. Ryu and S. Boyd. Primer on monotone operator methods. *Appl. Comput. Math*, 2016.

S. Shalev-Shwartz and Y. Singer. On the equivalence of weak learnability and linear separability: New relaxations and efficient boosting algorithms. *Mach. Learn.*, 2010.

P.-W. Wang and C.-J. Lin. Iteration complexity of feasible descent methods for convex optimization. *Journal of Machine Learning Research*, 2014.

M. Yan and W. Yin. Self equivalence of the alternating direction method of multipliers. In *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, 2016.

J. Yang and X. Yuan. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of computation*, 2013.

I. Yen, X. Huang, K. Zhong, R. Zhang, P. Ravikumar, and I. Dhillon. Dual decomposed learning with factorwise oracle for structural SVM with large output domain. In *NIPS*, 2016b.

I. E.-H. Yen, X. Lin, J. Zhang, P. Ravikumar, and I. Dhillon. A convex atomic-norm approach to multiple sequence alignment and motif discovery. In *ICML*, 2016a.

A. Yurtsever, Q. T. Dinh, and V. Cevher. A universal primal-dual convex optimization framework. In *NIPS*, 2015.