Model Based Conditional Gradient Method with Armijo-like Line Search

Yura Malitsky [*1] **Peter Ochs** [*2]

Abstract

The Conditional Gradient Method is generalized to a class of non-smooth non-convex optimization problems with many applications in machine learning. The proposed algorithm iterates by minimizing so-called model functions over the constraint set. Complemented with an Armijo line search procedure, we prove that subsequences converge to a stationary point. The abstract framework of model functions provides great flexibility for the design of concrete algorithms. As special cases, for example, we develop an algorithm for additive composite problems and an algorithm for non-linear composite problems which leads to a Gauss–Newton-type algorithm. Both instances are novel in non-smooth non-convex optimization and come with numerous applications in machine learning. Moreover, we obtain a hybrid version of Conditional Gradient and Proximal Minimization schemes for free, which combines advantages of both. Our algorithm is shown to perform favorably on a sparse non-linear robust regression problem and we discuss the flexibility of the proposed framework in several matrix factorization formulations.

1. Introduction

A prominent algorithm for applications in machine learning and statistics, such as matrix learning, recommender systems, clustering, etc., is the Conditional Gradient Method (aka Frank–Wolfe Method). Its success is based on a low per-iteration complexity in several applications. For example, in low rank approximation (e.g. matrix completion), the main computational cost per iteration is the minimization of a linear function over a nuclear norm (trace norm or Schatten 1-norm) constraint, which can be solved efficiently by approximating the singular vector associated with the largest singular value of the gradient that defines the linear function. In contrast, related proximal minimization algorithms require a full singular value decomposition, which is significantly more expensive.

In this paper, we generalize the Conditional Gradient Method to non-smooth non-convex optimization problems and unify the convergence analysis for several algorithms. The classical convergence analysis relies on the Descent Lemma, in the case the objective f has Lipschitz continuous gradient. The Descent Lemma states that, for some $L > 0$,

$$|f(x)-f(\bar{x})-\langle \nabla f(\bar{x}), x-\bar{x}\rangle| \leq \frac{L}{2}\|x-\bar{x}\|_2^2 \quad \text{for all } x, \bar{x}\,.$$

This inequality can also be interpreted as a measure for the linearization error of f around $\bar{x}$, i.e., the approximation quality of f by a linear function. We emphasize the fact that such a measure for the approximation quality of f, rather than smoothness, is key for the convergence of the algorithm. We generalize the linear approximation to any *model function* $f_{\bar{x}}$ that obeys a certain approximation quality

$$|f(x) - f_{\bar{x}}(x)| \leq \omega(\|x-\bar{x}\|)\,,$$

measured by a growth function $\omega\colon \mathbb{R}_+ \to \mathbb{R}_+$ that controls the approximation error. Note that this inequality does not imply smoothness, even in the special case $\omega(t) = \frac{L}{2}t^2$. If $f = g + h$ with a smooth function h and a non-smooth function g, we can define $f_{\bar{x}}(x) = g(x) + h(\bar{x}) + \langle \nabla h(\bar{x}), x-\bar{x}\rangle$ and observe that the approximation error is only due to the linearization of the smooth part h of the objective, while $f_{\bar{x}}$ is non-smooth. There are many other situations of interest. We choose the properties of the growth function ω such that $f_{\bar{x}}$ mimics a first order oracle of f. The freedom to choose the model function depending on the problem structure at hand makes our approach a flexible and efficient way to solve structured non-smooth non-convex minimization problems.

In this model function framework, our generalized Conditional Gradient update step at x_k reads

$$\begin{aligned} y_k &\in \operatorname*{argmin}_{x\in C} f_{x_k}(x) \\ x_{k+1} &= \gamma_k y_k + (1-\gamma_k)x_k\,, \end{aligned}$$

where $\gamma_k \in [0,1]$ and C is a compact and convex constraint set. For f_{x_k} being the linearization of f around x_k, this is exactly the Conditional Gradient Method.

[*]Equal contribution [1]University of Göttingen, Göttingen, Germany [2]Saarland University, Saarbrücken, Germany. Correspondence to: Peter Ochs <ochs@math.uni-sb.de>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

As for all methods, the efficiency depends on the cost to evaluate the oracle, which in our case is the minimization of f_{x_k} over C and, for proximal minimization problems, the cost to solve subproblems of type

$$\min_{x\in\mathbb{R}^N} f_{x_k}(x) + \frac{1}{2\tau}\|x - x_k\|^2 ,$$

for some step size $\tau > 0$. The generalization achieved in this paper increases the modelling flexibility for practical applications by making them accessible with another (possibly much cheaper) oracle, or by combining the oracles to a hybrid Proximal–Conditional Gradient method. In particular, we show the favorable performance of our algorithm for a sparse non-linear robust regression problem and demonstrate the flexibility of the algorithm on several applications in matrix factorization.

2. Contributions and Related Work

The idea of model functions to unify and generalize algorithms has been used before in bundle methods (Noll, 2013; Noll et al., 2008), where only a lower bound on the approximation error with the model function is used, which is a different setup. In (Drusvyatskiy et al., 2016; Ochs et al., 2018), the same class of model functions is considered as in our paper. In (Ochs et al., 2018), a Bregman proximal minimization framework is developed and convergence to a stationary point with an Armijo-like line search strategy is proved under weak assumptions on the Bregman distances. Their work can be seen as the proximal analogue to our framework. Recently, the model function framework has been extended to stochastic optimization (Davis & Drusvyatski, 2018; Davis et al., 2018).

Both, (Ochs et al., 2018) and our work, present an *implementable algorithm of the model function framework*, which is motivated by the abstract consideration of (pure) sequential model minimization in (Drusvyatskiy et al., 2016). The goal of (Drusvyatskiy et al., 2016) is to devise a measure for proximity to a stationary point, which can be used as a stopping criterion in non-smooth optimization. However, their convergence result depends on assumptions that are not automatically satisfied in practice. In (Ochs et al., 2018), model functions are complemented with additional structure (the Bregman proximity term) and Armijo line search. Once the model functions are selected, convergence of subsequences to a stationary point is guaranteed. We substitute the Bregman proximity by minimization of model functions over a compact set, and also obtain convergence of subsequences to a stationary point without additional assumptions.

A special case of our framework yields the Conditional Gradient Method (aka Frank–Wolfe method (Frank & Wolfe, 1956)) with Armijo line search. Convergence has been analyzed in (Bertsekas, 1999) for smooth constrained optimization and in (Reddi et al., 2016) for smooth stochastic problems. While, in convex optimization, convergence of the method is fairly well understood (Bach, 2015; Jaggi, 2013; Lacoste-Julien et al., 2013; Lacoste-Julien & Jaggi, 2015; Silveti-Falls et al., 2019; Yurtsever et al., 2018; Nesterov, 2018), little is known in the non-smooth non-convex setting. To the best of our knowledge, our work is *the first to generalize the Conditional Gradient minimization strategy to constrained non-smooth non-convex optimization with provable convergence (of subsequences) to a stationary point*. In this way, we contribute to the increase in modelling flexibility for problems in machine learning, computer vision, and statistics. In particular, we explore this *flexibility in an example from non-linear robust regression and several formulations from matrix factorization*.

As specific instances of our algorithmic framework, we *obtain new algorithms*. For example, we consider non-linear composite problems of type $\min_{x\in C} g(F(x))$ where F is sufficiently smooth and g is convex. Our iterative model function minimization over a convex constraint set yields an algorithm of Gauss-Newton type (Nocedal & Wright, 2006). Alternative strategies that use a proximal minimization strategy, which leads to Levenberg–Marquardt algorithm (Levenberg, 1944; Marquardt, 1963) in a certain special case, is explored, for example, in (Lewis & Wright, 2016; Drusvyatskiy et al., 2016; Ochs et al., 2018). The problems that can be modelled in this form is immense (Lewis & Wright, 2016). Using specific approximations of the objective by model functions, we also propose a *hybrid Proximal–Conditional Gradient minimization scheme* that combines the advantages of both worlds. In the convex setting, such a hybrid method was used in (Argyriou et al., 2014). However, their analysis was tailored to exactly this hybrid algorithm, whereas we obtain it from the model function framework for free and in the non-convex setting.

3. The Model Based Minimization Algorithm

We consider optimization problems of the form

$$\min_{x\in C} f(x) \tag{3}$$

with the following properties:

Assumption 1. *(i) C is a non-empty compact convex set in $\mathbb{R}^N$;*

(ii) $f\colon \mathbb{R}^N \to (-\infty, +\infty]$ is a proper lower semi-continuous (lsc) function that is bounded from below with $\operatorname{dom} f \subset C$.

As motivated in the introduction, the proposed algorithm is based on iteratively minimizing model functions of the objective in (3) over the constraint set C. These model functions obey a certain approximation quality with respect

Algorithm 1 (Model Based Conditional Gradient Method with Line Search)**.**

- **Optimization Problem:** *Problem* (3).
- **Initialization:** $x_0 \in \mathbb{R}^N$ *and set* $\rho \in (0, 1)$.
- **Update** $(k \geq 0)$*:*
  - *Find* $y_k \in C$ *such that the model improvement is positive, i.e.*

    $$\Delta(x_k, y_k) = f_{x_k}(x_k) - f_{x_k}(y_k) > 0, \tag{1}$$

    and compute

    $$x_{k+1} = x_k + \gamma_k(y_k - x_k) \tag{2}$$

    with $\gamma_k \in [0, 1]$ *determined by Algorithm* 2 *such that the following holds:*

    $$\text{(Armijo line search)} \quad \gamma_k \quad \text{satisfies} \quad f(x_{k+1}) \leq f(x_k) - \rho\gamma_k\Delta(x_k, y_k) \tag{ALS}$$

  - *If* (1) *cannot be satisfied (i.e.,* $\max_{y \in C} \Delta(x_k, y) \leq 0$*), then terminate the algorithm.*

Algorithm 2 (Armijo Line Search for Algorithm 1)**.**

- **Parameters:** *Fix* $\rho, \delta \in (0, 1)$ *and* $\tilde{\gamma} \in (0, 1]$.
- **Input:** $x_k, y_k \in C$ *that satisfy* (1).
- **Line Search:** *Find the smallest integer* $j \geq 0$ *such that* $\gamma_k = \tilde{\gamma}\delta^j$ *satisfies* (ALS).

to the objective function, which we measure in general using an (error) growth function:

Definition 3.1 (growth function)**.** *A continuous function* $\omega\colon \mathbb{R}_+ \to \mathbb{R}_+$ *is called growth function if it satisfies* $\omega(0) = 0$ *and* $\omega'_+(0) := \lim_{t \searrow 0} \omega(t)/t = 0$.

The standard example of a growth function is $\omega(t) = L \cdot t^r$ with $L > 0$ and $r > 1$. However, we may easily generate more examples using the concept of ψ-uniform continuity as in (Ochs et al., 2018), which generalizes Lipschitz and Hölder continuity. Note that from the definition of growth function, ones has $\omega(t) = o(t)$.

In this paper, we consider model functions that satisfy the following assumption.

Assumption 2 (model assumption)**.** *There exists a growth function* $\omega\colon \mathbb{R}_+ \to \mathbb{R}_+$ *such that for each* $\bar{x} \in \mathbb{R}^N$*, there exists a proper lsc convex function* $f_{\bar{x}}\colon \mathbb{R}^N \to (-\infty, +\infty]$ *such that* $\operatorname{dom} f = \operatorname{dom} f_{\bar{x}}$*, called model function, with the following property:*

$$|f(x) - f_{\bar{x}}(x)| \leq \omega(\|x - \bar{x}\|)\,, \quad \forall x \in C\,.$$

For examples of model functions, we refer to Section 4. The Model Assumption 2 preserves up to the first order information of the objective function in the following sense (see Lemma A.1 in the supplementary material)

$$f_{\bar{x}}(\bar{x}) = f(\bar{x}) \quad \text{and} \quad \widehat{\partial} f(\bar{x}) = \partial f_{\bar{x}}(\bar{x})\,, \tag{4}$$

where $\widehat{\partial} f$ denotes the Fréchet subdifferential (Definition 8.3 in (Rockafellar & Wets, 1998)) of f and ∂f the (convex) subdifferential, which coincides with the Fréchet subdifferential for convex functions (Proposition 8.12 in (Rockafellar & Wets, 1998)). The *Fréchet subdifferential* is defined at a point $\bar{x}$, at which f is finite, as $v \in \widehat{\partial} f(\bar{x})$ if and only if $f(x) \geq f(\bar{x}) + \langle v, x - \bar{x}\rangle + o(\|x - \bar{x}\|)$, and $\widehat{\partial} f(\bar{x}) = \emptyset$ for $\bar{x} \notin \operatorname{dom} f$.

Minimizing model functions from Assumption 2 provides a generic way to define algorithms with a first order oracle (possibly non-smooth). We seek to find a *(Fréchet) stationary point* $\bar{x}$ of (3), characterized by

$$0 \in \widehat{\partial} f(\bar{x})\,.$$

In Algorithm 1, the proposed algorithm is defined. The practical realization of the Armijo condition in (ALS), requires an algorithmic procedure. We propose the backtracking line search in Algorithm 2 as subroutine for (ALS).

Key for measuring the progress of the algorithm is the *model improvement*, which we define as

$$\Delta(x, y) := f_x(x) - f_x(y)\,, \quad \text{for all } x, y \in \mathbb{R}^N\,. \tag{5}$$

We show that this is a natural measure of stationarity.

In order to obtain a "stable" algorithm, in the sense that objective values are non-increasing, the choice of y_k satisfying (1) is arbitrary. However, the proof that all limit points of the sequence generated by Algorithm 1 are stationary points requires an additional assumption. We must assert that the error in solving the model subproblem vanishes for k tending towards infinity.

Assumption 3 (optimality of y_k)**.** *There exists $(\varepsilon_k)_{k\in\mathbb{N}}$ with $\varepsilon_k \searrow 0$ such that*

$$f_{x_k}(y_k) \le \min_{x\in C} f_{x_k}(x) + \varepsilon_k .$$

For each $k \in \mathbb{N}$, we denote by $\hat{y}_k$ any element in $\operatorname{argmin}_{x\in C} f_{x_k}(x)$.

Remark 3.2. *One option to choose y_k in (1) is to set $y_k = \hat{y}_k \in \operatorname{argmin}_{x\in C} f_{x_k}(x)$. Observe that this is equivalent to $y_k \in \operatorname{argmax}_{y\in C} \Delta(x_k, y)$. On the other hand, our framework is general enough to allow one solving $\min_{y\in C} f_{x_k}(y)$ with errors as in Assumption 3.*

3.1. Analysis of the Algorithm

3.1.1. Finite Termination of Line Search

We show that Algorithm 1 is well-defined, i.e., Algorithm 2 terminates after a finite number of iterations. We verify that $y_k - x_k$ is a *descent direction*, i.e., all sufficiently small choices of γ_k satisfy (ALS). Therefore, reducing γ_k according to the rule in Algorithm 2, it eventually enters a neighborhood of 0 after finitely many steps.

Proposition 3.3. *Fix $k \in \mathbb{N}$. There exists $\tilde{\gamma} \in (0, 1]$ such that (ALS) is satisfied for all $\gamma_k \in (0, \tilde{\gamma})$.*

The proof is in Section[1] A.1.

3.1.2. Finite Termination of the Algorithm

In case, the algorithm terminates after a finite number of iterations, i.e., (1) cannot be satisfied for any y_k, we have already found a stationary point.

Proposition 3.4. *Let $k \in \mathbb{N}$ be such that the model improvement is zero, i.e., $\max_{y\in C} \Delta(x_k, y) = 0$. Then, x_k is a stationary point of (3).*

The proof is in Section A.2.

Proposition 3.4 identifies the model improvement $\Delta(x_k, y_k)$ as a suitable measure for stationarity. For smooth functions, this is an obvious fact, as the following example shows.

Example 3.5. *If f is sufficiently smooth, a suitable model function is $f_{x_k}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k\rangle$, and the model improvement becomes*

$$\Delta(x_k, y_k) = \langle \nabla f(x_k), x_k - y_k\rangle > 0 ,$$

which is the characterization of a descent direction $v = y_k - x_k$ in classical smooth optimization. If there is no $y_k \in C$ along which the value of f_{x_k} can be reduced, then $\langle \nabla f(x_k), y - x_k\rangle \ge 0$ for all $y \in C$, which is the standard characterization of a stationary point x_k for constrained smooth optimization.

3.1.3. Asymptotic Analysis

In this section, we present the following main theorem.

Theorem 3.6 (convergence to a stationary point)**.** *Let Assumptions 1, 2 and 3 be satisfied and let $(x_k)_{k\in\mathbb{N}}$ be a sequence that is generated by Algorithm 1. Then, every limit point of $(x_k)_{k\in\mathbb{N}}$ is a stationary point of (3) and $(f(x_k))_{k\in\mathbb{N}}$ converges to the value of f at the limit point. Moreover, there exists at least one converging subsequence of $(x_k)_{k\in\mathbb{N}}$.*

The proof is in Section A.3.

Remark 3.7. *Theorem 3.6 guarantees to find a stationary point of the minimization problem in (3). Note, that we do not intend to guarantee that a global minimizer of (3) is found or approximated. This would ask for too much considering the broadness of the class of non-smooth non-convex optimization problems that (3) deals with. In this general framework, convergence of subsequences to a stationary point is quite satisfying and is the objective of most first order optimization schemes in non-convex optimization.*

Remark 3.8. *We can easily derive the following convergence rate from the Armijo line search condition (ALS):*

$$\min_{0\le i\le k} \Delta(x_i, y_i) \le \frac{f(x_0) - \inf f}{\rho \sum_{i=0}^{k} \gamma_k} , \quad \forall k \in \mathbb{N} .$$

However, considering practice experiments, we observed that the convergence rate is too conservative and does not reflect the actual performance of our algorithm.

4. Examples of Model Functions

As the assumption of model functions is the same as in (Ochs et al., 2018), the same examples may be incorporated here. However, we consider minimization of model functions over the constraint set C instead of (Bregman) proximal minimization. In order to make this work self contained, we mention their models (and some new ones) and discuss the algorithmic difference. For presentation, we focus on the case of maximal model improvement in (1). Let Γ_0 denote the class of proper lsc convex functions and $\mathscr{C}^{1,\psi}(C)$ be the class of smooth functions with ψ-uniformly continuous gradient relative to C, i.e., $f \in \mathscr{C}^{1,\psi}$ if and only if

$$\|\nabla f(x) - \nabla f(y)\| \le \psi(\|x - y\|) , \quad \forall x, y \in C ,$$

for some continuous function $\psi\colon \mathbb{R}_+ \to \mathbb{R}_+$ with $\psi(0) = 0$. The Generalized Descent Lemma (Lemma 4 in (Ochs et al.,

[1] Section A is provided in the supplementary material.

2018)) shows that such a function obeys, for all $x, \bar{x} \in C$,

$$|f(x) - f(\bar{x}) - \langle \nabla f(\bar{x}), x - \bar{x} \rangle| \le \omega(\|x - \bar{x}\|) \qquad (6)$$

with growth function $\omega(t) := \int_0^1 \frac{\varphi(st)}{s}\, ds$ where $\varphi(s) = s\psi(s)$. The most important example is $\psi(s) = cs^\alpha$ for some $c > 0$, which is Hölder continuity for $\alpha \in (0, 1]$ and Lipschitz continuity for $\alpha = 1$. It results in $\omega(t) = \frac{c}{1+\alpha} t^{1+\alpha}$. By the compact constraint set in (3), Lipschitz or Hölder continuity can be assumed to be global (possibly with a different constant).

Note that in general the following examples account for non-smooth non-convex optimization problems.

Example 4.1 (additive composite problems)**.** *Many problems in image processing, signal analysis, or statistics (including image deblurring, denoising, robust PCA, support vector machines, LASSO, etc.) can be cast in the form*

$$\min_{x \in C} f(x)\,, \quad f := g + h \quad \text{with } g \in \Gamma_0 \text{ and } h \in \mathscr{C}^{1,\psi}(C)\,.$$

A suitable model function for such problems is the following

$$f_{\bar{x}}(x) = g(x) + h(\bar{x}) + \langle \nabla h(\bar{x}), x - \bar{x} \rangle\,, \quad \forall x \in C\,.$$

Using (6), Assumption 2 is clearly satisfied.

In the proximal minimization framework (Ochs et al., 2018), this choice requires to solve subproblems of the form

$$\min_{x \in C} g(x) + \langle \nabla h(\bar{x}), x - \bar{x} \rangle + D(x, \bar{x})\,,$$

which are known as (Bregman) Proximal Gradient Descent update steps (aka. Forward–Backward Splitting or Mirror Descent), where $D(x, \bar{x})$ is a Bregman distance[2]*. For $D(x, \bar{x}) = \frac{1}{2}\|x - \bar{x}\|^2$ with $\tau > 0$, the mapping that assigns to $\bar{x}$ the solution of this problem is known as the proximal gradient mapping $\mathrm{prox}_{\tau g + \delta_C}(\bar{x} - \tau \nabla h(\bar{x}))$ with respect to $g + \delta_C$, where δ_C is the indicator function of the set C.*

Instead, for our generalized Conditional Gradient type algorithm (Algorithm 1) with maximal model improvement, the update step requires solving subproblems of type

$$\min_{x \in C} g(x) + \langle \nabla h(\bar{x}), x \rangle\,.$$

Key in selecting the "better" algorithm depends on the computational cost for solving the subproblems.

Example 4.2 (hybrid Proximal–Conditional Gradient minimization)**.** *Motivated by the comparison of proximal minimization and our Conditional Gradient type minimization in Example 4.1, the model function could be defined as*

$$f_{\bar{x}}(x) = g(x) + h(\bar{x}) + \langle \nabla h(\bar{x}), x - \bar{x} \rangle + \frac{1}{2\tau}\|x - \bar{x}\|^2\,,$$

leading to a proximal subproblem over a constraint set C in our Algorithm 1. In this sense, our model function framework allows us to interpolate between proximal minimization algorithms and Conditional Gradient type algorithms.

We may also combine linearization and proximal linearization to devise a model function that yields a hybrid version of Conditional Gradient and proximal minimization. Consider the optimization problem in Example 4.1 with $x = (x_1, x_2) \in \mathbb{R}^N$ and $C = C_1 \times C_2$, where g is additively separable, i.e., $g(x_1, x_2) = g_1(x_1) + g_2(x_2)$ for functions $g_1, g_2 \in \Gamma_0$. Then, the model function

$$f_{\bar{x}}(x) = g_1(x_1) + g_2(x_2) + h(\bar{x}) + \langle \nabla h(\bar{x}), x - \bar{x} \rangle + \frac{1}{2\tau}\|x_1 - \bar{x}_1\|^2\,, \quad x = (x_1, x_2)\,,$$

where $\nabla h(\bar{x}) = (\nabla_{x_1} h(\bar{x}), \nabla_{x_2} h(\bar{x}))$, leads to a proximal gradient step with respect to x_1 and a Conditional Gradient step with respect to x_2:

$$\hat{y}_1 = \mathrm{prox}_{\tau g_1 + \delta_{C_1}}\Big(\bar{x}_1 - \tau \nabla_{x_1} h(\bar{x})\Big)$$
$$\hat{y}_2 \in \operatorname*{argmin}_{x_2 \in C_2} g_2(x_2) + \langle \nabla_{x_2} h(\bar{x}), x_2 \rangle$$

where $\hat{y} = (\hat{y}_1, \hat{y}_2)$ yields the maximal model improvement.

Example 4.3 (Newton-based Conditional Gradient)**.** *Suppose that h in Example 4.1 is at least twice continuously differentiable. In that case, a second order expansion in the model function is feasible*

$$f_{\bar{x}}(x) = g(x) + h(\bar{x}) + \langle \nabla h(\bar{x}), x - \bar{x} \rangle + \frac{1}{2}\langle x - \bar{x}, [\nabla^2 h(\bar{x})]_+ (x - \bar{x}) \rangle\,,$$

where $[\nabla^2 h(\bar{x})]_+$ is the projection of the Hessian of h at $\bar{x}$ onto the cone of positive semi-definite matrices. The convexity assumption of our model functions requires us to replace the Hessian matrix by a positive semi-definite approximation. However, in general, unlike proximal minimization methods, thanks to the compact constraint set, we do not need to enforce strong convexity of the subproblem, i.e., $[\nabla^2 h(\bar{x})]_+$ need not be positive definite. In (Ochs et al., 2018) with $D(x, \bar{x}) = \frac{1}{2\tau}\|x - \bar{x}\|^2$ and $g \equiv 0$, this choice leads to damped (projected) Newton steps of the form

$$\hat{y} = \mathrm{proj}_C\Big(\bar{x} - \tau(I + \tau[\nabla^2 h(\bar{x})]_+)^{-1} \nabla h(\bar{x})\Big)$$

with identity matrix I. Our Algorithm 1 leads to a projected Newton step in subproblem (1) without damping

$$\hat{y} = \mathrm{proj}_C\Big(\bar{x} - [\nabla^2 h(\bar{x})]_+^{-1} \nabla h(\bar{x})\Big)\,.$$

Note the abuse of notation, since $[\nabla^2 h(\bar{x})]_+$ might not be invertible. In that case, a constrained quadratic program needs to be solved to obtain a point $\hat{y}$ that yields the maximal model improvement.

[2] The considered Bregman distances have the form $D(x, \bar{x}) = \varphi(x) - \varphi(\bar{x}) - \langle \nabla \varphi(\bar{x}), x - \bar{x} \rangle$, if $\bar{x} \in \operatorname{int}\operatorname{dom}\varphi$, with a so-called Legendre function φ (see Section 26 in (Rockafellar, 1970)), and $D(x, \bar{x}) = +\infty$ if $\bar{x} \notin \operatorname{int}\operatorname{dom}\varphi$.

Example 4.4 (Gauss–Newton). *Consider*

$$\begin{aligned}&\min_{x\in C} g(F(x)) \quad \textit{with}\\ &g\in\Gamma_0(\mathbb{R}^M) \textit{ Lipschitz and } F\in\mathscr{C}^{1,\psi}(C,\mathbb{R}^M)\,.\end{aligned} \tag{7}$$

This class of problems includes non-linear inverse problems. We present a simple application of non-linear regression in Section 5.1. A suitable model function is the following:

$$f_{\bar x}(x) = g(F(\bar x) + DF(\bar x)(x-\bar x))\,,$$

which is motivated by the Gauss–Newton method (Nocedal & Wright, 2006). In the proximal minimization framework, it leads to the ProxLinear (or ProxDescent) algorithm (Lewis & Wright, 2016), which can solve a broad class of problems. Often, the arising subproblems do not have closed form solution. However, convexity allows for their efficient minimization. Due to the broad class of problems that is covered by (7), in general, no simpler algorithms are currently known. There are essentially two ways of incorporating line search: (i) line search in direction of the solution of the subproblem (Ochs et al., 2018) or (ii) line search of the scaling of the proximity term to successively push the new iterate closer to the old iterate (Lewis & Wright, 2016; Drusvyatskiy et al., 2016). Where (i) requires to solve the subproblem once, (ii) requires to solve the subproblem in each trial of a step size.

The line search strategy (i) is the same as in (ALS). As our subproblems do not involve the additional distance term, in contrast to proximal minimization subproblems, the search directions that we find are closer linked to the original problem, and hence we expect faster progress of our method. The experiment in Section 5.1 supports this intuition.

Example 4.5. *The flexibility of our algorithm allows model functions to be tailored to specific problems. Suppose g in (7) is additively separable, i.e., $g(y)=\sum_{i=1}^M g_i(y_i)$ and g_i is convex and non-decreasing, e.g., the hinge loss $g_i(y_i)=\max(y_i,0)$ that is used in support vector machines. Then, model functions with coordinate-wise higher order convex approximations of $F(x)=(F_1(x),\dots,F_M(x))$ can be used to devise higher order convex model functions.*

5. Applications

5.1. Sparse Robust Non-linear Regression

We consider a simple non-smooth non-convex sparse robust regression problem (Hampel et al., 1986) of the form

$$\min_{(a,b)\in C}\sum_{i=1}^M \|F_i(a,b)-y_i\|_1 + \mu\|a\|_1\,,$$
$$F_i(a,b) := \sum_{j=1}^P a_j\exp(-b_j x_i)\,,$$

similar to (Ochs et al., 2018), where the data $(x_i,y_i)\in\mathbb{R}^2$, $i=1,\dots,M$, is a sequence of covariate-observation pairs and $C=[0,\overline{a}]^P\times[0,\overline{b}]^P$ for some $\overline{a},\overline{b}>0$ and $P\in\mathbb{N}$. We assume that $y_i=F_i(a,b)+n_i$ where n_i are iid errors drawn from a Laplacian distribution, which motivates the usage of the ℓ_1-norm data fidelity term. Moreover, we assume that a large percentage of coefficients a_j are zero, which is the reason for penalizing also the ℓ_1-norm of the parameter vector $a\in\mathbb{R}^P$. By "symmetry" of F_i, the number of zero coordinates matters rather than the actual support.

We compare several algorithms with provable convergence of subsequences to a stationary point for solving the problem. The objective function falls into the class of problems of Example 4.4, for example, since F has bounded Hessian on C, hence, its gradient is Lipschitz continuous on C. All algorithms are based on that choice of model functions. We write the linearization of the inner functions around $u_k=(a_k,b_k)$ as follows: For all $i=1,\dots,M$,

$$F_i(a,b)-y_i\approx \mathcal{K}_i u - y_i^\diamond\,,\qquad \text{for all } u=(a,b)\in C\,,$$

where $\mathcal{K}_i = DF_i(u_k)$ and $y_i^\diamond := y_i - F_i(u_k) + DF_i(u_k)u_k$. Our Algorithm 1, denoted `FW-CompLinLS`[3], leads to subproblems of the form

$$\min_{u=(a,b)\in C}\sum_{i=1}^M \|\mathcal{K}_i u - y_i^\diamond\|_1 + \mu\|a\|_1\,,$$

the algorithm in (Ochs et al., 2018), denoted `ProxLinearLS`, and (Lewis & Wright, 2016), denoted `ProxLinearBT`, require to solve subproblems of the form

$$\min_{u=(a,b)\in C}\sum_{i=1}^M \|\mathcal{K}_i u - y_i^\diamond\|_1 + \mu\|a\|_1 + \frac{1}{2\tau}\|u-u_k\|^2\,.$$

We solve the inner problem using the Primal–Dual Hybrid Gradient Algorithm with preconditioning (Pock & Chambolle, 2011), which allows for step sizes that are automatically computed based on $\mathcal{K}_i$. We use warm starting for all methods. Our algorithm `FW-CompLinLS` and `ProxLinearLS` solve the subproblem up to a certain accuracy and perform an Armijo-like line search in the direction of the approximate solution. The backtracking of `ProxLinearBT` is with respect to the parameter τ and involves solving the subproblem for each trial "step size" τ until a sufficient improvement of the objective value is observed. All methods perform line search for improving the objective value relative to the model improvement $\Delta(x_k,y_k)$.

The data for the experiment is generated randomly with $P=100$, $M=1000$, $\mu=80$, $\overline{a}=20$, $\overline{b}=5$, and 80% of

[3] Abbreviation for Frank–Wolfe Composite Linear splitting with line search.

coefficients a_j are randomly set to 0. Figure 1 shows the data and the convergence of the objective value or the model improvement with respect to actual computation time.

5.2. Structured Matrix Factorization

Many applications in data analysis such as blind image deblurring (Kopriva & Nuzillard, 2006; Chaudhuri et al., 2014), clustering and principal component analysis (Duda et al., 2001; Murphy, 2013), source separation (Lee & Seung, 1999; Févotte et al., 2009; Cichocki et al., 2009), signal processing (Aharon et al., 2006; Starck et al., 2015), or dictionary learning (Mairal et al., 2010; Xu et al., 2017) can be formulated as structured matrix factorization problems. In this section, we demonstrate the flexible applicability of our algorithm to various formulations of matrix factorization problems. Most algorithms for solving such problems depend on alternating minimization techniques (Cichocki et al., 2009; Chaudhuri et al., 2014; Starck et al., 2015), sometimes with linearization (Bolte et al., 2014; Pock & Sabach, 2016). Algorithms are usually based on a proximal minimization oracle. In (Ochs et al., 2018), several formulations of matrix factorization are presented using Bregman proximal minimization steps. This approach has a great advantage for several constraint sets.

However, for example, proximal minimization of low rank constraints (e.g., constraints on the nuclear norm or 1-Schatten norm) require a full singular value decomposition (SVD), which can be expensive for large (or huge) scale data analysis problems (Cai et al., 2010). In these settings, a Conditional Gradient minimization oracle is favorable. It requires to estimate the singular vector corresponding to the largest singular value only, which is computationally significantly cheaper than a full SVD. While this technique has been used frequently in (convex) low rank approximation schemes (Jaggi, 2013), it has not been explored in detail for structured matrix factorization due to the non-convexity of the problem. We discuss several formulations of matrix factorization problems with focus on such low-rank constraints. Due to the favorable properties of the generalized Conditional Gradient minimization oracle as described above, we believe that benchmarking is not required.

We highlight the flexible applicability of our framework to non-convex problems of the form

$$\min_{X,Y} \frac{1}{2}\|A - XY\|_F^2 + g(X)\,, \quad \text{s.t. } X \in \mathcal{X},\ Y \in \mathcal{Y}\,,$$

where the goal is to represent a matrix A as a product XY with matrices $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are (convex and compact) constraint sets that encode some problem specific characteristics and g is a convex regularization function. We propose to use the additive composite splitting model from Example 4.1, i.e., we set $C = \mathcal{X} \times \mathcal{Y}$, $h(X, Y) = \frac{1}{2}\|A - XY\|_F^2$ and solve the following subproblems:

$$\min_{X,Y} g(X) + \langle X, (X_k Y_k - A) Y_k^\top \rangle \\ + \langle Y, X_k^\top (X_k Y_k - A)\rangle_F\,, \quad \text{s.t. } X \in \mathcal{X},\ Y \in \mathcal{Y}\,. \quad (8)$$

Of course, the Frobenius norm in h could be replaced by any smooth function, for example, the log-student-t distribution $\sum_{i,j} \log(1 + (A - XY)_{i,j}^2)$ for robust estimations. The linearization of h makes the minimization separable, which allows us to discuss minimization steps with respect to X and Y independently.

Examples for $\mathcal{X}$. In dictionary learning, $\mathcal{X}$ describes the set of feasible atoms that may be used for reconstructing A. It is common to normalize the atoms, e.g.,

$$\mathcal{X}_1 = \left\{ X \,:\, \forall j\colon \sum_i X_{i,j}^2 \leq 1\,,\ \forall j > 2\colon \,:\sum_i X_{i,j} = 0 \right\}\,,$$

which is a classical choice for dictionary learning (Xu et al., 2017). For column $j = 1$, the update step in (8) is the projection of the 1st column (of the gradient) onto the ℓ_2-unit ball, and for $j > 1$, by projecting the mean-subtracted jth column onto the ℓ_2-unit ball. The choice

$$\mathcal{X}_2 = \left\{ X \,:\, \forall j\colon \sum_i X_{i,j} = 1\,,\ \forall i,j\colon X_{i,j} \geq 0 \right\}$$

enforces normalization and non-negativity, which is commonly used in non-negative matrix factorization (NMF) (Lee & Seung, 1999). The update step in (8) sets columnwise a smallest coordinates to 1 and all others to 0.

In (Ochs et al., 2018), a closed form update step with respect to $\mathcal{X}_2$ is derived by a suitable choice of Bregman distance. Proximal minimization with respect to the Euclidean distance requires an algorithmic approach, though, which is also simple, as it is just a projection onto a unit simplex.

Examples for $\mathcal{Y}$ and g. Sparsity is a favorable property for several matrix factorization problems. Conditional Gradient steps with respect to several norm constraints lead to simple updates (Jaggi, 2013; Bach, 2013). For example, for some $r > 0$, set $\mathcal{Y}_1 = \{Y \,:\, \|Y\|_1 \leq r\}$ to promote sparsity of the matrix Y. It can be used in dictionary learning (Xu et al., 2017) to express A with only a few atoms of X, i.e., many entries of Y shall be 0. Analogously, convex relaxations of rank-r constraints are commonly used, which can be modelled by $\mathcal{Y}_2 = \{Y \,:\, \|Y\|_* \leq r\}$. The nuclear norm $\|Y\|_*$ of Y enforces the columns of A to be spanned by at most r different linear subspaces, which is related to clustering problems. The Conditional Gradient subproblems with respect to both constraint sets $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are simple

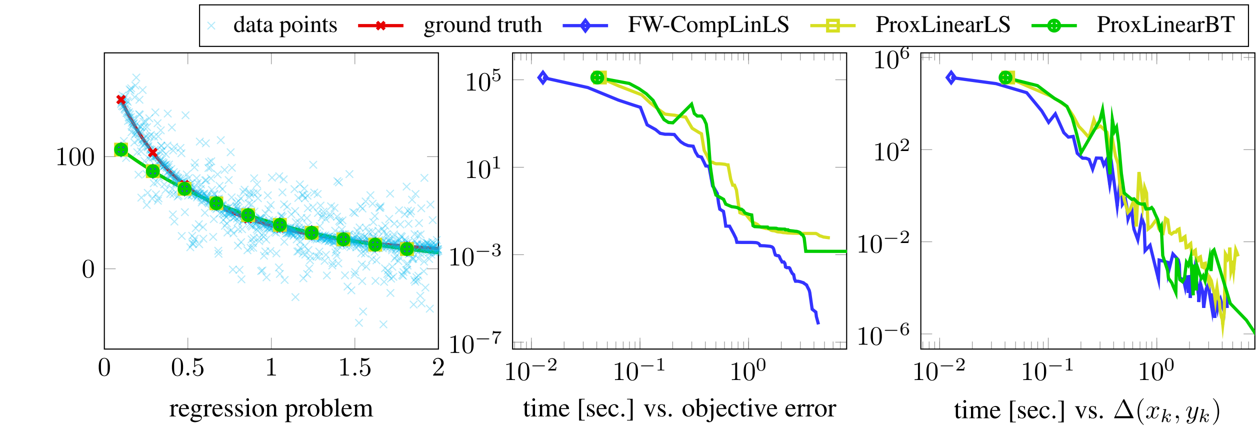

Figure 1. Regression function and convergence plots for solving the robust regression problem in Section 5.1. All methods find the same regression function, as the left plot shows. The plot in the middle shows $f(x_k) - \underline{f}$ where $\underline{f}$ is the smallest objective value found by any of the methods. The right plot shows the convergence of the model improvement, which is a measure for stationarity. The convergence is given with respect to actual computation time in seconds. Our method `FW-CompLinLS` outperforms the proximal line search `ProxLinearLS` and backtracking `ProxLinearBT` based methods.

(Jaggi, 2013), where the second one requires the estimation of the extreme singular vector as mentioned above.

However, on top of the constraint sets, we can use $g \not\equiv 0$, which may be used as penalty instead of a constraint, for example, penalizing the nuclear norm (Harchaoui et al., 2012) or structured sparsity (Bach et al., 2012). The convex subproblem that arise in this context have been studied in convex optimization (Dudik et al., 2012; Harchaoui et al., 2015; Nesterov, 2018). Also note that the solution of subproblems in (8) with respect to Y can be related to finding a subgradient in the subdifferential of the convex conjugate evaluated at the current gradient (Bach, 2015).

Hybrid Proximal–Conditional Gradient minimization. Finally, we discuss an alternative model function to (8), motivated by Example 4.2. We define the model function by linearization of the objective with respect to Y and a convex quadratic approximation with respect to X. This choice leads to subproblems of the following form for our Algorithm 1:

$$\begin{aligned}\min_{X,Y}\, & g(X) + \langle X, (X_kY_k - A)Y_k^\top \rangle \\ & + \langle Y, X_k^\top (X_kY_k - A)\rangle_F + \frac{1}{2\tau}\|Y - Y_k\|_F^2 \,, \\ & \qquad\qquad\qquad \text{s.t. } X \in \mathcal{X},\ Y \in \mathcal{Y}\,,\end{aligned}$$

for some $\tau > 0$, leading to Conditional Gradient type problems with respect to X and proximal minimization problems with respect to Y. The matrix factorization problem and the algorithm can be formulated to explore the advantages of both worlds. For example, a nuclear norm constraint with respect to Y should be handled by a Conditional Gradient step and an additional group-sparsity penalty on X can be efficiently handled by proximal minimization steps (Bach et al., 2012).

6. Conclusion

We have presented an algorithmic framework that generalizes the Conditional Gradient method from constrained convex or smooth minimization to a class of constrained non-smooth non-convex minimization problems. The algorithm is formulated with respect to sequential minimization of model functions over the constraint set, complemented with an Armijo line search procedure. Model functions are simple surrogates of the objective function that obey a certain approximation quality and capture first order information of the problem. We presented several examples of model functions, including examples for additive or nonlinear composite problems, which demonstrates the gain in flexibility for solving problems in machine learning, computer vision, and statistics. The possibility to tailor model functions to the specific structure of the optimization problem at hand, allows for efficient minimization. We also devise a hybrid method that combines Conditional Gradient type update steps with proximal minimization steps, which is particularly interesting for matrix factorization problems. In a numerical experiment for robust non-linear regression, the algorithm performs favorably compared to proximal minimization based algorithms.

Acknowledgements

Peter Ochs acknowledges funding by the German Research Foundation (DFG Grant OC 150/1-1).

References

Aharon, M., Elad, M., and Bruckstein, A. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Image Processing*, 54(11):4311–4322, November 2006.

Argyriou, A., Signoretto, M., and Suykens, J. Hybrid Conditional Gradient - Smoothing Algorithms with Applications to Sparse and Low Rank Regularization. *ArXiv e-prints*, April 2014. arXiv: 1404.3591.

Bach, F. Convex relaxations of structured matrix factorizations. *ArXiv e-prints*, September 2013. arXiv: 1309.3117.

Bach, F. Duality Between Subgradient and Conditional Gradient Methods. *SIAM Journal on Optimization*, 25(1): 115–129, January 2015.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. Optimization with Sparsity-Inducing Penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, January 2012.

Bertsekas, D. P. *Nonlinear Programming*. Athena Scientific, 2nd edition, September 1999. ISBN 1886529000.

Bolte, J., Sabach, S., and Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

Cai, J., Candès, E., and Shen, Z. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization*, 20(4):1956–1982, January 2010.

Chaudhuri, S., Velmurugan, R., and Rameshan, R. *Blind Image Deconvolution*. Springer, 2014.

Cichocki, A., Zdunek, R., Phan, A., and Amari, S. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Wiley, New York, 2009.

Davis, D. and Drusvyatski, D. Stochastic model-based minimization of weakly convex functions. *ArXiv e-prints*, 2018. arXiv: 1803.06523.

Davis, D., Drusvyatskiy, D., and MacPhee, K. Stochastic model-based minimization under high-order growth. *ArXiv e-prints*, 2018. arXiv: 1807.00255.

Drusvyatskiy, D., Ioffe, A. D., and Lewis, A. S. Nonsmooth optimization using Taylor-like models: error bounds, convergence, and termination criteria. *ArXiv e-prints*, October 2016. arXiv: 1610.03446.

Duda, R., Hart, P., and Stork, D. *Pattern Classification*. Wiley, New York, 2 edition, 2001.

Dudik, M., Harchaoui, Z., and Malick, J. Lifted coordinate descent for learning with trace-norm regularization. In *International Conference on Artificial Intelligence and Statistics*, volume 22, pp. 327–336, April 2012.

Févotte, C., Bertin, N., and Durrieu, J. Nonnegative Matrix Factorization with the Itakura-Saito Divergence: With Application to Music Analysis. *Neural Computation*, 21 (3):793–830, March 2009.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(12): 95–110, March 1956.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. *Robust Statistics: The Approach Based on Influence Functions*. MIT Press, Cambridge, MA, 1986.

Harchaoui, Z., Douze, M., Paulin, M., Dudik, M., and Malick, J. Large-scale image classification with trace-norm regularization. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3386–3393, June 2012.

Harchaoui, Z., Juditsky, A., and Nemirovski, A. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1-2):75–112, August 2015.

Jaggi, M. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *International Conference on Machine Learning (ICML)*, pp. 427–435, February 2013.

Kopriva, I. and Nuzillard, D. Non-negative Matrix Factorization Approach to Blind Image Deconvolution. In Rosca, J., Erdogmus, D., Príncipe, J., and Haykin, S. (eds.), *Independent Component Analysis and Blind Signal Separation*, Lecture Notes in Computer Science, pp. 966–973. Springer Berlin Heidelberg, 2006.

Lacoste-Julien, S. and Jaggi, M. On the Global Linear Convergence of Frank-Wolfe Optimization Variants. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems (NIPS)*, pp. 496–504. Curran Associates, Inc., 2015.

Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *International Conference on Machine Learning (ICML)*, pp. 53–61, February 2013.

Lee, D. and Seung, H. Learning the part of objects from nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

Levenberg, K. A method for the solution of certain nonlinear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

Lewis, A. and Wright, S. A proximal method for composite minimization. *Mathematical Programming*, 158(1-2): 501–546, 2016.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research*, 11:19–60, March 2010.

Marquardt, D. An algorithm for least-squares estimation of nonlinear parameters. *Society for Industrial and Applied Mathematics*, 11:431–441, 1963.

Murphy, K. *Machine learning: a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013.

Nesterov, Y. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming*, 171(1):311–330, September 2018.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

Noll, D. Convergence of non-smooth descent methods using the Kurdyka–Łojasiewicz inequality. *Journal of Optimization Theory and Applications*, 160(2):553–572, September 2013.

Noll, D., Prot, O., and Apkarian, P. A proximity control algorithm to minimize nonsmooth and nonconvex functions. *Pacific Journal of Optimization*, 4(3):571–604, 2008.

Ochs, P., Fadili, J., and Brox, T. Non-smooth non-convex bregman minimization: Unification and new algorithms. *Journal of Optimization Theory and Applications*, 181(1): 244–278, 2018.

Pock, T. and Chambolle, A. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *International Conference on Computer Vision (ICCV)*, 2011.

Pock, T. and Sabach, S. Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences*, 9 (4):1756–1787, January 2016.

Reddi, S., Sra, S., Póczos, B., and Smola, A. Stochastic Frank-Wolfe methods for nonconvex optimization. In *Conference on Communication, Control, and Computing*, pp. 1244–1251, September 2016.

Rockafellar, R. T. *Convex Analysis*. Princeton University Press, Princeton, 1970.

Rockafellar, R. T. and Wets, R. J.-B. *Variational Analysis*, volume 317. Springer Berlin Heidelberg, Heidelberg, 1998.

Silveti-Falls, A., Molinari, C., and Fadili, J. Generalized Conditional Gradient with Augmented Lagrangian for Composite Minimization. *ArXiv e-prints*, January 2019. arXiv: 1901.01287.

Starck, J.-L., Murtagh, F., and Fadili, J. *Sparse image and signal processing: wavelets, curvelets, morphological diversity*. Cambridge University Press, 2nd edition, 2015.

Xu, Y., Li, Z., Yang, J., and Zhang, D. A survey of dictionary learning algorithms for face recognition. *IEEE Access*, 5: 8502–8514, 2017.

Yurtsever, A., Fercoq, O., Locatello, F., and Cevher, V. A Conditional Gradient Framework for Composite Convex Minimization with Applications to Semidefinite Programming. *International Conference on Machine Learning (ICML)*, July 2018.