Dual subgradient algorithms for large-scale nonsmooth learning problems

Bruce Cox* Anatoli Juditsky† Arkadi Nemirovski‡

February 12, 2013

Abstract

"Classical" First Order (FO) algorithms of convex optimization, such as Mirror Descent algorithm or Nesterov's optimal algorithm of smooth convex optimization, are well known to have optimal (theoretical) complexity estimates which do not depend on the problem dimension. However, to attain the optimality, the domain of the problem should admit a "good proximal setup". The latter essentially means that 1) the problem domain should satisfy certain geometric conditions which we refer to as "favorable geometry", and 2) the practical use of these methods is conditioned by our ability to solve efficiently an auxiliary optimization task – computing *proximal transformation* – at each iteration of the method. More often than not these two conditions are satisfied in optimization problems arising in computational learning, what explains the fact that FO methods of proximal type recently became methods of choice when solving various learning problems. Yet, they meet their limits in several important problems such as multi-task learning with large number of tasks, where the problem domain does not exhibit favorable geometry, and learning and matrix completion problems with nuclear norm constraint, when the numerical cost of solving the auxiliary problem becomes prohibitive in large-scale problems.

We propose a novel approach to solving nonsmooth optimization problems arising in learning applications where Fenchel-type representation of the objective function is available. The approach is based on applying FO algorithms to the dual problem and using the *accuracy certificates* supplied by the method to recover the primal solution. While suboptimal in terms of accuracy guaranties, the proposed approach does not rely upon "good proximal setup" for the primal problem but requires the problem domain to admit a *Linear Optimization oracle* – the ability to efficiently maximize a linear form on the domaine of the primal problem.

1 Introduction

Motivation and background. The problem of interest in this paper is a convex optimization problem in the form

$$\mathrm{Opt}(P) = \max_{x \in X} f_*(x) \tag{P}$$

where X is a nonempty closed and bounded subset of Euclidean space E_x, and f_* is concave and Lipschitz continuous function on X. We are interested in the situation where the sizes of the problem put it beyond the "practical grasp" of polynomial time interior point methods with their rather computationally expensive in the large scale case iterations. In this case the

*US Air Force

†LJK, Université J. Fourier, B.P. 53, 38041 Grenoble Cedex 9, France, Anatoli.Juditsky@imag.fr

‡Georgia Institute of Technology, Atlanta, Georgia 30332, USA, nemirovs@isye.gatech.edu

Research of the third author was supported by the ONR grant N000140811104 and NSF grant DMS 0914785.

methods of choice are the First Order (FO) optimization techniques. The state of the art of these techniques can be briefly summarized as follows:

• The most standard FO approach to (P) requires to provide X, E_y with *proximal setup* $\|\cdot\|, \omega(\cdot)$, that is to equip the space E_x with a norm $\|\cdot\|$, and the domain X of the problem – with a *distance-generating function* (d.-g.f.) $\omega(x) : X \to \mathbf{R}$ which should be convex and continuous on X, admit a continuous in $x \in X^o = \{x \in X : \partial\omega(x) \neq \emptyset\}$ selection $\omega'(x)$ of subgradients, and be strongly convex, modulus 1, w.r.t. $\|\cdot\|$:

$$\langle \omega'(x) - \omega'(x'), x - x' \rangle \geq \|x - x'\|^2. \tag{1}$$

After such a setup is fixed, generating an ϵ-solution to the problem (i.e., a points $x_\epsilon \in X$ satisfying $\mathrm{Opt}(P) - f_*(x_\epsilon) \leq \epsilon$) costs at most $N(\epsilon)$ *steps*, where

- $N(\epsilon) = O(1)\frac{\Omega_X^2 L^2}{\epsilon^2}$ in the nonsmooth case, where f_* is Lipschitz continuous, with constant L w.r.t. $\|\cdot\|$ (Mirror Descent (MD) algorithm, see, e.g., [8, Chapter 5]), and
- $N(\epsilon) = O(1)\frac{D\Omega_X}{\sqrt{\epsilon}}$ in the smooth case, where f_* possesses Lipschitz continuous, with constant D^2, gradient: $\|f_*'(x) - f_*'(x')\|_* \leq D^2\|x - x'\|$, where $\|\cdot\|$ is the norm conjugate to $\|\cdot\|$ (Nesterov's optimal algorithm for smooth convex optimization, see, e.g., [11]).

In the above bounds, $\Omega_X = \Omega[X, \omega(\cdot)] = \sqrt{2[\max_{x\in X}\omega(x) - \min_{x\in X}\omega(x)]}$ is the *ω-diameter* of X; here and in the sequel $O(1)$'s stand for positive absolute constants.

A *step* of a FO method essentially reduces to a single computation of f_*, f_*' at a point and a single computation of the *prox-mapping*

$$\mathrm{Prox}_x(\xi) := \operatorname*{argmin}_{x'\in X} \left[\langle \xi - \omega'(x), x' \rangle + \omega(x')\right].$$

for a pair $x \in X^0, \xi \in E_x$.

• A different way of processing (P) by FO methods, originating in the breakthrough paper of Nesterov [11], is to use *Fenchel-type representation of f_**:

$$f_*(x) = \min_{y\in Y} \left[F(x, y) := \langle x, Ay + a\rangle + \psi(y)\right], \tag{2}$$

where Y is a closed and bounded subset of Euclidean space E_y and $\psi(y)$ is a convex function. Representations of this type are readily available for a wide family of "well-structured" nonsmooth objectives f_*; moreover, usually we can make ϕ to possess Lipschitz continuous gradient or even to be linear (for instructive examples, see, e.g., [11] or [8, Chapter 6]). Whenever this is the case, and given proximal setups $(\|\cdot\|_x, \omega_x(\cdot))$ and $(\|\cdot\|_y, \omega_y(\cdot))$ for (E_x, X) and for (E_y, Y), we can find an ϵ-solution to (P) in

$$N(\epsilon) \leq O(1)\frac{L_{xx}\Omega_X^2 + 2L_{xy}\Omega_X\Omega_Y + L_{yy}\Omega_Y^2}{\epsilon}$$

steps (Nesterov's smoothing [11] or the Mirror Prox algorithm, see, e.g., [8, Chapter 6]). Here $\Omega_X = \Omega[X, \omega_x(\cdot)]$, $\Omega_Y = \Omega[Y, \omega_y(\cdot)])$, and L_{xx}, L_{yy}, L_{xy} are the partial Lipschitz constants of $\nabla F(x, y)$, namely,

$$\begin{array}{l}\forall(x, x' \in X, y, y' \in Y):\\ \quad \|\nabla_x F(x', y') - \nabla_x F(x, y)\|_{x,*} \leq L_{xx}\|x' - x\|_x + L_{xy}\|y' - y\|_y,\\ \quad \|\nabla_y F(x', y') - \nabla_y F(x, y)\|_{y,*} \leq L_{xy}\|x' - x\|_x + L_{yy}\|y' - y\|_y,\end{array}$$

and $\|\cdot\|_{x,*}$, $\|\cdot\|_{y,*}$ are the norms conjugate to $\|\cdot\|_x$, $\|\cdot\|_y$, respectively. A step of the method requires a single computation of $\nabla F(\cdot)$ at a point and computing the values of $O(1)$ prox-mappings associated with $(X, \omega_x(\cdot))$ and $(Y, \omega_y(\cdot))$.

Clearly, to be practical, methods of the outlined type should rely on "good" proximal setups – those resulting in "moderate" values of Ω_X *and* Ω_Y and not too difficult to compute prox-mappings, associated with ω_X *and* ω_Y. This is indeed the case for domains X arising in numerous applications (for instructive examples, see, e.g., see [8, Chapter 5]). The question addressed in this paper is what to do when one of the domains, namely, X does not admit a "good" proximal setup. Here are two instructive examples:

A. X is the unit ball of the nuclear norm $\|\sigma(\cdot)\|_1$ in the space $\mathbf{R}^{p\times q}$ of $p \times q$ matrices (from now on, for a $p \times q$ matrix x, $\sigma(x) = [\sigma_1(x); ...; \sigma_{\min[p,q]}(x)]$ denotes the vector comprised by singular values of x taken in the non-ascending order). This domain arises in various low-rank-oriented problems of matrix recovery. In this case, X does admit a proximal setup with $\Omega_X = O(1)\sqrt{\ln(pq)}$. However, computing prox-mapping involves full singular value decomposition of a $p \times q$ matrix and becomes prohibitively time-consuming when p, q are in the range of tens of thousand. Note that this hardly is a shortcoming of the existing proximal setups, since already computing the nuclear norm (that is, checking the inclusion $x \in X$) requires an SVD decomposition of x.

B. X is a high-dimensional box – the unit ball of the $\|\cdot\|_\infty$-norm of $\mathbf{R}^m$ with large m, or, more generally, the unit ball of the ℓ_∞/ℓ_2 norm $\|x\|_{\infty|2} = \max_{1\le i\le m} \|x^i\|_2$, where $x = [x^1; ...; x^m] \in E = \mathbf{R}^{n_1} \times ... \times \mathbf{R}^{n_m}$. Here it is easy to point out a proximal setup with an easy-to-compute prox mapping (e.g., the *Euclidean setup* $\|\cdot\| = \|\cdot\|_2$, $\omega(x) = \frac{1}{2}\langle x, x\rangle$). However, it is easily seen that whenever $\|\cdot\|$ is a norm satisfying $\|e_j\| \ge 1$ for all basic orths e_j [1], one has $\Omega_X \ge O(1)\sqrt{m}$, that is, the (theoretical) performance of "natural" proximal FO methods deteriorates rapidly as m grows.

Note that whenever a prox-mapping associated with X is "easy to compute," it is equally easy to maximize over X a linear form (since $\mathrm{Prox}_x(-t\xi)$ converges to the maximizer of $\langle\xi, x\rangle$ over X as $t \to \infty$). In such a case, we have at our disposal an efficient *Linear Optimization (LO) oracle* – a routine which, given on input a linear form ξ, returns a point $x_X(\xi) \in \mathrm{Argmax}_{x\in X}\langle\xi, x\rangle$. This conclusion, however, cannot be reversed – our abilities to maximize, at a reasonable cost, linear functionals over X does not imply the possibility to compute a prox-mapping at a comparable cost. For example, when $X \in \mathbf{R}^{p\times q}$ is the unit ball of the nuclear norm, maximizing a linear function $\langle\xi, x\rangle = \mathrm{Tr}(\xi x^T)$ over X requires finding the largest singular value of a $p \times q$ matrix and associated left singular vector. For large p and q, solving the latter problem is by orders of magnitude cheaper than computing full SVD of a $p \times q$ matrix. This and similar examples motivate the interest, especially in Machine Learning community, in optimization techniques solving (P) via an LO oracle for X. In particular, the only "classical" technique of this type – the *Conditional Gradient (CG)* algorithm going back to Frank and Wolfe [7] – has attracted much attention recently. In the setting of CG method it is assumed that f is smooth (with Lipschitz/Hölder continuous gradient), and the standard result here (which is widely known, see, e.g., [4, 5, 13]) is the following.

[1] This is a natural normalization: indeed, $\|e_j\| \ll 1$ means that j-th coordinate of x has a large Lipschitz constant w.r.t. $\|\cdot\|$, in spite of the fact that this coordinate is "perfectly well behaved" on X – its variation on the set is just 2.

Proposition 1.1 *Let X be a closed and bounded convex set in a Euclidean space E_x such that $X - X$ linearly spans E_x. Assume that we are given a point $x_1 \in X$ and an LO oracle for X, and let f_* be a concave continuously differentiable function on X such that for some $\mathcal{L} < \infty$ and $q \in (1, 2]$ one has*

$$\forall x, x' \in X : f_*(y) \geq f_*(x) + \langle f'_*(x), x' - x\rangle - \frac{1}{q}\mathcal{L}\|x' - x\|_X^q, \tag{3}$$

where $\|\cdot\|_X$ is the norm on E_x with the unit ball $X - X$. Consider the recurrences

$$\begin{array}{llcl} (a) & x_{t+1} & \in & \mathrm{Argmax}_{x\in\Delta_t} f_*(x),\ \Delta_t = [x_t, x_X(f'_*(x_t))], \\ (b) & x_{t+1} & = & x_t + \lambda_t[x_X(f'_*(x_t)) - x_t],\ \lambda_t = \frac{2}{t+1}, \end{array} \tag{4}$$

where $x_X(\xi) \in \mathrm{Argmax}_{x\in X}\langle \xi, x\rangle$ and $x_1 \in X$. Then for all $t = 2, 3, ...$ one has

$$\epsilon_t := \max_{x\in X} f_*(x) - f_*(x_t) \leq \left\{ \begin{array}{ll} q^{q-1}D(t+q-2)^{1-q}, & \textit{recurrence } (a) \\ \max\left[3^{q-1}, \frac{2^q}{q(3-q)}\right] D(t+1)^{1-q}, & \textit{recurrence } (b) \end{array} \right. . \tag{5}$$

Contents of this paper. Assuming an LO oracle for X available, the major limitation in solving (P) by the Conditional Gradient method is the requirement for problem objective f_* to be smooth (otherwise, there are no particular requirements to the problem geometry). What to do if this requirement is not satisfied? In this paper, we investigate two simple options for processing this case, based on Fenchel-type representation (2) of f_*. Primarily, we focus on "nonsmooth" approach: we assume that such a representation is available and involves a Lipschitz continuous convex function ψ given by a First Order oracle (i.e., a black box which, given on input $y \in Y$, returns the value $\psi(y)$ and a subgradient $\psi'(y)$ of ψ at y). Besides this, we assume that Y (but not X!) *does admit a proximal setup* $(\|\cdot\|_y, \omega_y(\cdot))$. In this case, we can pass from the problem of interest (P) to its dual

$$\mathrm{Opt}(D) = \min_{y\in Y}\left[f(y) := \max_{x\in X} F(x, y)\right],\ F(x, y) = \langle x, Ay + a\rangle + \psi(y). \tag{D}$$

Clearly, the LO oracle for X along with the FO oracle for ψ provide a FO oracle for (D):

$$f(y) = \langle x(y), Ay + a\rangle + \psi(y),\ f'(y) = A^T x(y) + \psi'(y),\ x(y) := x_X(Ay + a).$$

Since Y admits a proximal setup, this is enough to allow to get an ϵ-solution to (D) in $N(\epsilon) = O(1)\frac{L^2\Omega_Y^2}{\epsilon^2}$ steps, L being the Lipschitz constant of f w.r.t. $\|\cdot\|_y$. Whatever slow the resulting rate of convergence could look, we shall see in the mean time that there are important applications where this rate seems to be the best known so far. When implementing the outlined scheme, the only nontrivial question is how to recover a good optimal solution to the problem (P) of actual interest from a good approximate solution to its dual problem (D). The proposed answer to this question stems from the recent (and pretty simple at the first glance) machinery of *accuracy certificates* proposed recently in [10], and closely related to the work [12]. The summary of our approach is as follows. When solving (D) by a FO method, we generate *search points* $y_\tau \in Y$ where the subgradients $f'(y_\tau)$ of f are computed; as a byproduct of the latter computation, we have at our disposal the points $x_\tau = x(y_\tau)$. As a result, after t steps we have at our disposal *execution protocol* $y^t = \{y_\tau, f'(y_\tau)\}_{\tau=1}^t$. An *accuracy certificate* associated with this protocol is, by definition, a collection $\lambda^t = \{\lambda_\tau^t\}_{\tau=1}^t$ of nonnegative weights λ_τ^t summing up to 1: $\sum_{\tau=1}^t \lambda_\tau^t = 1$. The *resolution* of the certificate is, by definition, the quantity

$$\epsilon(y^t, \lambda^t) = \max_{y\in Y}\sum_{\tau=1}^t \lambda_\tau^t\langle f'(y_\tau), y_\tau - y\rangle.$$

An immediate observation is (see section 2) that setting $\widehat{y}^t = \sum_{\tau=1}^t \lambda_\tau^t y_\tau$, $\widehat{x}^t = \sum_{\tau=1}^t \lambda_\tau^t x_\tau$, we get a pair of feasible solutions to (D) and to (P) such that

$$[f(\widehat{y}^t) - \mathrm{Opt}(D)] + [\mathrm{Opt}(P) - f_*(\widehat{x}^t)] \leq \epsilon(y^t, \lambda^t).$$

Thus, assuming that the FO method in question produces, in addition to search points, accuracy certificates for the resulting execution protocols and that *the resolution of these certificates goes to 0 as $t \to \infty$ at some rate*, we can use the certificates to build feasible approximate solutions to (D) *and to* (P) with nonoptimalities, in terms of the objectives of the respective problems, going to 0 at the same rate.

The scope of the outlined approach depends on whether we are able to equip known methods of nonsmooth convex minimization with computationally cheap mechanisms for building "good" accuracy certificates. The meaning of "good" in this context is exactly that the rate of convergence of the corresponding resolution to 0 is identical to the standard efficiency estimates of the methods (e.g., for MD this would mean that $\epsilon(y^t, \lambda^t) \leq O(1)L\Omega_Y t^{-1/2}$). [10] provides a positive answer to this question for the most attractive academically polynomial time oracle-oriented algorithms for convex optimization, like the Ellipsoid method. These methods, however, usually are poorly suited for large-scale applications. In this paper, we provide a positive answer to the above question for the three most attractive oracle-oriented FO methods for nonsmooth convex optimization known to us. Specifically, we consider

- MD (where accuracy certificates are easy to obtain, see also [12]),
- Full Memory Mirror Descent Level (MDL) method (a Mirror Descent extension to the Bundle-Level method [9]; to the best of our knowledge, this extension was not yet described in the literature), and
- Non-Euclidean Restricted Memory Level method (NERML) originating from [2], which we believe is the most attractive tool for large-scale nonsmooth oracle-based convex optimization. To the best of our knowledge, equipping NERML with accuracy certificates is a novel development.

We also consider a different approach to non-smooth convex optimization over a domain given by LO oracle, approach mimicking Nesterov's smoothing [11]. Specifically, assuming, as above, that f_* is given by Fenchel-type representation (2) with Y admitting a proximal setup, we use this setup, *exactly in the same way as in [11]*, to approximate f_* by a smooth function which then is minimized by the CG algorithm. Therefore, the only difference with [11] is in replacing Nesterov's optimal algorithm for smooth convex optimization (which requires a good proximal point setup for X) with although slower, but less demanding (just LO oracle for X is enough) CG method. We shall see in the mean time that, unsurprisingly, the theoretical complexity of the two outlined approaches – "nonsmooth" and "smoothing" one – are essentially the same.

The main body of the paper is organized as follows. In section 2, we develop the components of the approach related to duality and show how an accuracy certificate with small resolution yields a pair of good approximate solutions to (P) and (D). In section 3, we show how to equip the MD, MDL and NERML algorithms with accuracy certificates. In section 4, we investigate the outlined smoothing approach. In section 5, we consider examples, primarily of Machine Learning origin, where we prone the usage of the proposed algorithms. Some technical proofs are relegated to the appendix.

2 Duality and accuracy certificates

2.1 Situation

Let E_x be an Euclidean space, $X \subset E_x$ be a nonempty closed and bounded convex set equipped with *LO oracle* – a procedure which, given on input $\xi \in E_x$, returns a maximizer $x_X(\xi)$ of the linear form $\langle \xi, x\rangle$ over $x \in X$. Let $f_*(x)$ be a concave function given by *Fenchel-type representation*:

$$f_*(x) = \min_{y\in Y} \left[\langle x, Ay + a\rangle + \psi(y)\right], \tag{6}$$

where Y is a closed compact subset of an Euclidean space E_y and ψ is a Lipschitz continuous convex function on Y given by a First Order oracle.

In the sequel we set

$$f(y) = \max_{x\in X} \left[\langle x, Ay + a\rangle + \psi(y)\right],$$

and consider two optimization problems

$$\begin{array}{rcll} \mathrm{Opt}(P) & = & \max\limits_{x\in X} f_*(x) & (P) \\ \mathrm{Opt}(D) & = & \min\limits_{y\in Y} f(y) & (D) \end{array}$$

By the standard saddle point argument, we have $\mathrm{Opt}(P) = \mathrm{Opt}(D)$.

2.2 Main observation

Observe that the First Order oracle for ψ along with the LO oracle for X provide a First Order oracle for (D); specifically, the vector field

$$f'(y) = A^T x_X(Ay + a) + \psi'(y) : \ Y \to E_y,$$

where $\psi'(y) \in \partial\psi(y)$ is a subgradient field of f.

Consider a collection $y^t = \{y_\tau \in Y, f'(y_\tau)\}_{\tau=1}^t$ along with a collection $\lambda^t = \{\lambda_\tau \geq 0\}_{\tau=1}^t$ such that $\sum_{\tau=1}^t \lambda_\tau = 1$, and let us set

$$\begin{array}{rcl} y(y^t, \lambda^t) & = & \sum_{\tau=1}^t \lambda_\tau y_\tau, \\ x(y^t, \lambda^t) & = & \sum_{\tau=1}^t \lambda_\tau x_X(Ay_\tau + a), \\ \epsilon(y^t, \lambda^t) & = & \max\limits_{y\in Y} \sum_{\tau=1}^t \lambda_\tau \langle f'(y_\tau), y_\tau - y\rangle. \end{array}$$

In the sequel, the components y_τ of y^t will be the search points generated by a First Order minimization method as applied to (D) at the steps $1, ..., t$, We call y^t the associated *execution protocol*, call a collection λ^t of t nonnegative weights summing up to 1 an *accuracy certificate* for this protocol, and refer to the quantity $\epsilon(y^t, \lambda^t)$ as to the *resolution of the certificate* λ^t *at the protocol* y^t.

Our main observation is as follows:

Proposition 2.1 *Let y^t, λ^t be as above. Then $\widehat{x} := x(y^t, \lambda^t)$, $\widehat{y} := y(y^t, \lambda^t)$ are feasible solutions to problems (P), (D), respectively, and*

$$f(\widehat{y}) - f_*(\widehat{x}) = [f(\widehat{y}) - \mathrm{Opt}(D)] + [\mathrm{Opt}(P) - f_*(\widehat{x})] \leq \epsilon(y^t, \lambda^t). \tag{7}$$

Proof. Let $F(x,y) = \langle x, Ay + a\rangle + \psi(x)$ and $x(y) = x_X(Ay+a)$, so that $f(y) = F(x(y), y)$. Observe that $f'(y) = F'_y(x(y), y)$, where $F'_y(x,y)$ is a selection of the subdifferential of F w.r.t. y, that is, $F'_y(x,y) \in \partial_y F(x,y)$ for all $x \in X$, $y \in Y$. Setting $x_\tau = x(y_\tau)$, we have for all $y \in Y$:

$$\begin{aligned}
\epsilon(y^t, \lambda^t) &\geq \sum_{\tau=1}^t \lambda_\tau \langle f'(y_\tau), y_\tau - y\rangle = \sum_{\tau=1}^t \lambda_\tau \langle F'_y(x_\tau, y_\tau), y_\tau - y\rangle \\
&\geq \sum_{\tau=1}^t \lambda_\tau \left[F(x_\tau, y_\tau) - F(x_\tau, y)\right] \ [\text{by convexity of } F \text{ in } y] \\
&= \sum_{\tau=1}^t \lambda_\tau \left[f(y_\tau) - F(x_\tau, y)\right] \ [\text{since } x_\tau = x(y_\tau), \text{ so that } F(x_\tau, y_\tau) = f(y_\tau)] \qquad (8) \\
&\geq f(\widehat{y}) - F(\widehat{x}, y) \ [\text{by convexity of } f \text{ and concavity of } F(x,y) \text{ in } x].
\end{aligned}$$

We conclude that

$$\epsilon(y^t, \lambda^t) \geq \max_{y \in y} \left[f(\widehat{y}) - F(\widehat{x}, y)\right] = f(\widehat{y}) - f_*(\widehat{x}).$$

The inclusions $\widehat{x} \in X$, $\widehat{y} \in Y$ are evident. □

Remark 2.1 In the proof of Proposition 2.1, the linearity of F w.r.t. x was never used, so that in fact we have proved a more general statement:
Given a concave in $x \in X$ and convex in $y \in Y$ Lipschitz continuous function $F(x,y)$, let us associate with it a convex function $f(y) = \max_{x \in X} F(x,y)$, a concave function $f_*(x) = \min_{y \in Y} F(x,y)$ and problems (P) and (D). Let $F'_y(x,y)$ be a vector field with $F'_y(x,y) \in \partial_y F(x,y)$, so that with $x(y) \in \mathrm{Argmax}_{x \in X} F(x,y)$, the vector $f'(y) = F'_y(x(y), y)$ is a subgradient of f at y. Assume that problem (D) associated with F is solved by a FO method using $f'(y) = F'_y(x(y), y)$ which produced execution protocol y^t and accuracy certificate λ^t. Then setting

$$\widehat{x} = \sum_\tau \lambda_\tau x(y_\tau), \ \ \textit{and} \ \ \widehat{y} = \sum_\tau \lambda_\tau y_\tau,$$

we ensure (7).

Moreover, let $\delta \geq 0$, and let $x_\delta(y)$ be a δ-maximizer of $F(x,y)$ in $x \in X$: for all $y \in Y$,

$$F(x_\delta(y), y) \geq \max_{x \in X} F(x,y) - \delta.$$

Suppose that (D) is solved by a FO method using approximate subgradients $\tilde{f}'(y) = F'_y(x_\delta(y), y)$, and producing execution protocol $y^t = \{y_\tau, \tilde{f}'(y_\tau)\}_{\tau=1}^t$ and accuracy certificate λ^t. Then setting $\widehat{x} = \sum_\tau \lambda_\tau x_\delta(y_\tau)$ and $\widehat{y} = \sum_\tau \lambda_\tau y_\tau$, we ensure the δ-relaxed version of (7) *– the relation*

$$f(\widehat{y}) - f_*(\widehat{x}) \leq \epsilon(y^t, \lambda^t) + \delta, \ \epsilon(y^t, \lambda^t) = \max_{y \in Y} \sum_{\tau=1}^t \lambda_\tau^t \langle \tilde{f}'(y_\tau), y_\tau - y\rangle.$$

All we need to extract the "Moreover" part of this statement from the proof of Proposition 2.1 is to set $x_\tau = x_\delta(y_\tau)$, to replace $f'(y_\tau)$ with $\tilde{f}'(y_\tau)$ and to replace the equality in (8) with the inequality

$$\sum_{\tau=1}^t \lambda_\tau \left[F(x_\tau, y_\tau) - F(x_\tau, y)\right] \geq \sum_{\tau=1}^t \lambda_\tau \left[f(y_\tau) - \delta - F(x_\tau, y)\right].$$

Discussion. Proposition 2.1 says that whenever we can equip the subsequent execution protocols generated by a FO method, as applied to the dual problem (D), with accuracy certificates, we can generate solutions to the primal problem (P) of inaccuracy going to 0 at the same rate as the certificate resolution. In the sequel, we shall point out some "good" accuracy certificates for several most attractive FO algorithms for nonsmooth convex minimization.

3 Accuracy certificates in oracle-oriented methods for large-scale nonsmooth convex optimization

3.1 Convex minimization with certificates, I: Mirror Descent

3.1.1 Proximal setup.

As it was mentioned in the introduction, the Mirror Descent (MD) algorithm solving (D) is given by a norm $\|\cdot\|$ on E_y and a distance-generating function (d.-g.f.) $\omega(y): Y \to \mathbf{R}$ which should be continuous and convex on Y, should admit a continuous in $y \in Y^o = \{y \in Y : \partial\omega(y) \neq \emptyset\}$ selection of subdifferentials $\omega'(y)$, and should be strongly convex, modulus 1, w.r.t. $\|\cdot\|$, that is,

$$\forall y, y' \in Y^o : \langle \omega'(y) - \omega'(y'), y - y' \rangle \geq \|y - y'\|^2.$$

A proximal setup $(\|\cdot\|, \omega(\cdot))$ for Y, E_y gives rise to several entities, namely,

- Bregman distance $V_y(z) = \omega(z) - \omega(y) - \langle \omega'(y), z - y \rangle$ $(y \in Y^o, z \in Y)$. Due to strong convexity of ω, we have

$$\forall (z \in Y, y \in Y^o) : V_y(z) \geq \frac{1}{2}\|z - y\|^2; \tag{9}$$

- ω-center $y_\omega = \operatorname{argmin}_{y \in Y} \omega(y)$ of Y and ω-diameter

$$\Omega = \Omega[Y, \omega(\cdot)] := \sqrt{2\left[\max_{y \in Y} \omega(y) - \min_{y \in Y} \omega(y)\right]}.$$

  Observe that

$$\langle \omega'(y_\omega), y - y_\omega \rangle \geq 0, \tag{10}$$

  (see Lemma A.1), so that

$$V_{y_\omega}(z) \leq \omega(z) - \omega(y_\omega) \leq \frac{1}{2}\Omega^2, \, \forall z \in Y, \tag{11}$$

  which combines with the inequality $V_y(z) \geq \frac{1}{2}\|z - y\|^2$ to yield the relation

$$\forall y \in Y : \|y - y_\omega\| \leq \Omega; \tag{12}$$

- prox-mapping

$$\operatorname{Prox}_y(\xi) = \operatorname*{argmin}_{z \in Y} \left[\langle \xi, z \rangle + V_y(z)\right],$$

  where $\xi \in E_y$ and $y \in Y^o$. This mapping takes its values in Y^o and satisfies the identity

$$\forall (y \in Y^o, \xi \in E_y, y_+ = \operatorname{Prox}_y(\xi)) : \langle \xi, y_+ - z \rangle \leq V_y(z) - V_{y_+}(z) - V_y(y_+) \; \forall z \in Y, \tag{13}$$

  see Lemma A.1.

3.1.2 Mirror Descent algorithm

MD algorithm works with a vector field

$$y \mapsto g(y) : Y \to E_y, \tag{14}$$

which is *oracle represented*, meaning that we have access to an oracle which, given on input $y \in Y$, returns $g(y)$. From now on we assume that this field is bounded:

$$\|g(y)\|_* \le L[g] < \infty, \forall y \in Y, \tag{15}$$

where $\|\cdot\|_*$ is the norm conjugate to $\|\cdot\|$. The algorithm is the recurrence

$$y_1 = y_\omega; y_\tau \mapsto g_\tau := g(y_\tau) \mapsto y_{\tau+1} := \mathrm{Prox}_{y_\tau}(\gamma_\tau g_\tau), \tag{MD}$$

where $\gamma_\tau > 0$ are stepsizes. Let us equip this recurrence with accuracy certificates, setting

$$\lambda^t = \left(\sum_{\tau=1}^t \gamma_\tau\right)^{-1} [\gamma_1; ...; \gamma_t]. \tag{16}$$

Proposition 3.1 *For every t, the resolution*

$$\epsilon(y^t, \lambda^t) := \max_{y \in Y} \sum_{\tau=1}^t \lambda_\tau \langle g(y_\tau), y_\tau - y\rangle$$

of λ^t on the execution protocol $y^t = \{y_\tau, g(y_\tau)\}_{\tau=1}^t$ satisfies the standard MD efficiency estimate

$$\epsilon(y^t, \lambda^t) \le \frac{\Omega^2 + \sum_{\tau=1}^t \gamma_\tau^2 L^2[g]}{2\sum_{\tau=1}^t \gamma_\tau}. \tag{17}$$

In particular, if $\gamma_\tau = \frac{\gamma(t)}{\|g(y_\tau)\|_*}$, $\gamma(t) := \frac{\Omega}{\sqrt{t}}$ for $1 \le \tau \le t$ [2]*,*

$$\epsilon(y^t, \lambda^t) \le \frac{\Omega L[g]}{\sqrt{t}}. \tag{18}$$

Proof is given by the standard MD rate-of-convergence derivation:

$$\begin{array}{rl}
\forall z \in Y: & \langle \gamma_\tau g_\tau, y_{\tau+1} - z\rangle \le V_{y_\tau}(z) - V_{y_{\tau+1}}(z) - V_{y_\tau}(y_{\tau+1}) \text{ [see (13)]} \\
\Rightarrow \forall z \in Y: & \langle \gamma_\tau g_\tau, y_\tau - z\rangle \le V_{y_\tau}(z) - V_{y_{\tau+1}}(z) + [\underbrace{\langle g_\tau, y_\tau - y_{\tau+1}\rangle}_{\le \gamma_\tau \|g_\tau\|_* \|y_\tau - y_{\tau+1}\|} - \underbrace{V_{y_\tau}(y_{\tau+1})}_{\ge \frac{1}{2}\|y_{\tau+1} - y_\tau\|^2}] \\
\Rightarrow \forall z \in Y: & \langle \gamma_\tau g_\tau, y_\tau - z\rangle \le V_{y_\tau}(z) - V_{y_{\tau+1}}(z) + \frac{1}{2}\gamma_\tau^2 \|g_\tau\|_*^2 \\
\Rightarrow \forall z \in Y: & \sum_{\tau=1}^t \gamma_t \langle g_\tau, y_\tau - z\rangle \le \underbrace{V_{y_w}(z)}_{\le \frac{1}{2}\Omega^2} - \underbrace{V_{y_{t+1}}(z)}_{\ge 0} + \frac{1}{2}\sum_{\tau=1}^t \gamma_\tau^2 \|g_\tau\|_*^2 \\
\Rightarrow \forall z \in Y: & \sum_{\tau=1}^t \lambda_\tau^t \langle g_\tau, y_\tau - z\rangle \le \frac{\Omega^2 + \sum_{\tau=1}^t \gamma_\tau^2 L^2[g]}{2\sum_{\tau=1}^t \gamma_\tau} \Rightarrow (17),
\end{array}$$

where the concluding $\Rightarrow$ is given by (15) □

[2] We assume here that $g_\tau \ne 0$ for all $\tau \le t$. In the opposite case, the situation is trivial: when $g(y_{\tau_*}) = 0$, for some $\tau* \le t$, setting $\lambda_\tau^t = 0$ for $\tau \ne \tau_*$ and $\lambda_{\tau_*}^t = 1$, we ensure that $\epsilon(y^t, \lambda^t) = 0$.

Solving (P) **and** (D) **using MD** In order to solve (D), we apply MD to the vector field $g = f'$. Assuming that

$$L_f = \sup_{y \in Y} \{\|f'(y)\|_* \equiv \|A^* x(y) + \psi'(y)\|_*\} < \infty \quad [x(y) = x_X(Ay + a)], \tag{19}$$

we can set $L[g] = L_f$. For this setup, Proposition 2.1 implies that the MD accuracy certificate λ^t, as defined in (16), taken together with the MD execution protocol $y^t = \{y_\tau, g(y_\tau) = f'(y_\tau) := A^* \underbrace{x(y_\tau)}_{x_\tau} + a + \psi'(y_\tau)\}_{\tau=1}^t$, yield the primal-dual feasible approximate solutions

$$\widehat{x}^t = \sum_{\tau=1}^t \lambda_\tau^t x_\tau, \ \widehat{y}^t = \sum_{\tau=1}^t \lambda_\tau^t y_\tau \tag{20}$$

to (P) and to (D) such that

$$f(\widehat{y}^t) - f_*(\widehat{x}^t) \le \epsilon(y^t, \lambda^t).$$

Combining this conclusion with Proposition 3.1, we arrive at the following result:

Corollary 3.1 *In the case of* (19)*, for every* $t = 1, 2, ...$ *the* t*-step MD with the stepsize policy*[3]

$$\gamma_\tau = \frac{\Omega}{\sqrt{t}\|f'(x_\tau)\|_*}, \ 1 \le \tau \le t$$

as applied to (D) *yields feasible approximate solutions* $\widehat{x}^t$*,* $\widehat{y}^t$ *to* (P)*,* (D) *such that*

$$\begin{gathered}[f(\widehat{y}^t) - \mathrm{Opt}(P)] + [\mathrm{Opt}(D) - f_*(\widehat{x}^t] \le \frac{\Omega L_f}{\sqrt{t}} \\ [\Omega = \Omega[Y, \omega(\cdot)]]\end{gathered} \tag{21}$$

In particular, given $\epsilon > 0$*, it takes at most*

$$t(\epsilon) = \mathrm{Ceil}\left(\frac{\Omega^2 L_f^2}{\epsilon^2}\right) \tag{22}$$

steps of the algorithm to ensure that

$$[f(\widehat{y}^t) - \mathrm{Opt}(P)] + [\mathrm{Opt}(D) - f_*(\widehat{x}^t)] \le \epsilon. \tag{23}$$

3.2 Convex minimization with certificates, II: Mirror Descent with full memory

Algorithm MDL – Mirror Descent Level method – is a non-Euclidean version of (a variant of) the Bundle-Level method [9]; to the best of our knowledge, this extension was not presented in the literature. Another novelty in what follows is equipping the method with accuracy certificates.

MDL is a version of MD with "full memory", meaning that the first order information on the objective being minimized is preserved and utilized at subsequent steps, rather than being "summarized" in the current iterate, as it is the case for MD. While the guaranteed accuracy bounds for MDL are similar to those for MD, the typical practical behavior of the algorithm is better than that of the "memoryless" MD.

[3] We assume that $f'(y_\tau) \ne 0$, for $\tau \le t$; otherwise, as we remember, the situation is trivial.

3.2.1 Preliminaries

MDL with certificates which we are about to describe is aimed at processing an oracle-represented vector field (14) satisfying (15), with the same assumptions on Y and the same proximal setup as in the case of MD.

We associate with $y \in Y$ the affine function

$$h_y(z) = \langle g(y), y - z \rangle \geq f(y) - f(z),$$

and with a finite set $S \subset Y$ the family $\mathcal{F}_S$ of affine functions on E_y which are convex combinations of the functions $h_y(z)$, $y \in S$. In the sequel, the words "we have at our disposal a function $h(\cdot) \in \mathcal{F}_S$" mean that we know the functions $h_y(\cdot)$, $y \in S$, and nonnegative weights λ_y, $y \in S$, summing up to 1, such that $h(z) = \sum_{y \in S} \lambda_y h_y(z)$.

The goal of the algorithm is, given a tolerance $\epsilon > 0$, to find a finite set $S \subseteq Y$ and $h \in \mathcal{F}_S$ such that

$$\max_{y \in Y} h(y) \leq \epsilon. \tag{24}$$

Note that our target S, h are of the form $S = \{y_1, ..., y_t\}$, $h(y) = \sum_{\tau=1}^t \lambda_\tau \langle g(y_\tau), y_\tau - y \rangle$ with nonnegative λ_τ summing up to 1. In other words, our target is to build an execution protocol $y^t = \{y_\tau, g(y_\tau)\}_{\tau=1}^t$ *and* an associated accuracy certificate λ^t such that $\epsilon(y^t, \lambda^t) \leq \epsilon$.

3.2.2 Construction

As applied to (14), MDL at a step $t = 1, 2, ...$ generates search point $y_t \in Y$ where the value $g(y_t)$ of g is computed; it provides us with the affine function $h_t(z) = \langle g(y_t), y_t - z \rangle$. Steps of the method are split into subsequent *phases* numbered $s = 1, 2, ...$, and every phase is associated with *optimality gap* $\Delta_s \geq 0$.

To initialize the method, we set $y_1 = y_\omega$, $S_0 = \emptyset$, $\Delta_0 = +\infty$.

At a step t we act as follows:

- given y_t, we compute $g(y_t)$, thus getting $h_t(\cdot)$, and set $S_t^+ = S_{t-1} \cup \{t\}$;
- we solve the auxiliary problem

$$\epsilon_t = \max_{y \in Y} \min_{\tau \in S_t^+} h_\tau(y) = \max_{y \in Y} \min_{\lambda} \left\{ \sum_{\tau \in S_t^+} \lambda_\tau h_\tau(y) : \lambda_\tau \geq 0, \sum_{\tau \in S_t^+} \lambda_\tau = 1 \right\} \tag{25}$$

  By the von Neumann lemma, an optimal solution to this (auxiliary) problem is associated with nonnegative and summing up to 1 weights λ_τ^t, $\tau \in S_t^+$ such that

$$\epsilon_t = \max_{y \in Y} \sum_{\tau \in S_t^+} \lambda_\tau^t h_\tau(y),$$

  and we assume that as a result of solving (25), both ϵ_t and λ_τ^t become known. We set $\lambda_\tau^t = 0$ for all $\tau \leq t$ which are not in S_t^+, thus getting an accuracy certificate $\lambda^t = [\lambda_1^t; ...; \lambda_t^t]$ for the execution protocol $y^t = \{y_\tau, g(y_\tau)\}_{\tau=1}^t$ along with $h^t(\cdot) = \sum_{\tau=1}^t \lambda_\tau^t h_\tau(\cdot)$. Note that by construction

$$\epsilon(y^t, \lambda^t) = \max_{y \in Y} h^t(y) = \epsilon_t.$$

  If $\epsilon_t \leq \epsilon$, we terminate – $h(\cdot) = h^t(\cdot)$ satisfies (24). Otherwise we proceed as follows:

- If (case A) $\epsilon_t \le \gamma\Delta_{s-1}$, $\gamma \in (0,1)$ being method's control parameter, we say that step t starts phase s (e.g., step $t=1$ starts phase 1), set

$$\Delta_s = \epsilon_t,\ S_t = \{\tau : 1 \le \tau \le t : \lambda_\tau^t > 0\} \cup \{1\},\ \widehat{y}_t = y_\omega$$

otherwise (case B) we set

$$S_t = S_t^+,\ \widehat{y}_t = y_t.$$

Note that in both cases we have

$$\epsilon_t = \max_{y\in Y} \min_{\tau\in S_t} h_\tau(y) \tag{26}$$

- Finally, we define *t-th level* as $\ell_t = \gamma\epsilon_t$ and associate with this quantity the *level set* $U_t = \{y \in Y : h_\tau(y) \ge \ell_t\ \forall \tau \in S_t\}$, specify y_{t+1} as the ω-projection of $\widehat{y}_t$ on U_t:

$$y_{t+1} = \operatorname*{argmin}_{y\in U_t} \left[\omega(y) - \langle \omega'(\widehat{y}_t), y\rangle\right] \tag{27}$$

and loop to step $t+1$.

3.2.3 Efficiency estimate

Proposition 3.2 *Given on input a target tolerance $\epsilon > 0$, the MDL algorithm terminates after finitely many steps, with the output $y^t = \{Y \ni y_\tau, g(y_\tau)\}_{\tau=1}^t$, $\lambda^t = \{\lambda_\tau^t \ge 0\}_{\tau=1}^t$, $\sum_\tau \lambda_\tau^t = 1$ such that*

$$\epsilon(y^t, \lambda^t) \le \epsilon. \tag{28}$$

The number of steps of the algorithm does not exceed

$$N = \frac{\Omega^2 L^2[g]}{\gamma^4(1-\gamma^2)\epsilon^2} + 1 \qquad [\Omega = \Omega[Y, \omega(\cdot)]]. \tag{29}$$

For proof, see Section A.2.

Remark 3.1 Assume that $\omega(\cdot)$ is continuously differentiable on the entire Y, so that the quantity

$$\Omega^+ = \Omega^+[Y,\omega] := \max_{y,z\in Y} V_y(z)$$

is finite. From the proof of Proposition 3.2 it follows immediately that one can substitute the rule "$\widehat{y}_t = y_\omega$ when t starts a phase and $\widehat{y}_t = y_t$ otherwise" with a simpler one "$\widehat{y}_t = y_t$ for all t," at the price of replacing Ω in (29) with Ω^+.

Solving (P) and (D) via MDL is completely similar to the case of MD: given a desired tolerance $\epsilon > 0$, one applies MDL to the vector field $g(y) = f'(y)$ until the target (24) is satisfied. Assuming (19), we can set $L[g] = L_f$, so that by Proposition 3.3 our target will be achieved in

$$t(\epsilon) \le \mathrm{Ceil}\left(\frac{\Omega^2 L_f^2}{\gamma^4(1-\gamma^2)\epsilon^2} + 1\right) \quad [\Omega = \Omega[Y,\omega(\cdot)]] \tag{30}$$

steps, with L_f given by (19). Assuming that the target is attained at a step t, we have at our disposal the execution protocol $y^t = \{y_\tau, f'(y_\tau)\}_{\tau=1}^t$ along with the accuracy certificate $\lambda^t = \{\lambda_\tau^t\}$ such that $\epsilon(y^t, \lambda^t) \le \epsilon$ (by the same Proposition 3.2). Therefore, specifying $\widehat{x}^t$, $\widehat{y}^t$ according to (20) and invoking Proposition 2.1, we ensure (23). Note that the complexity $t = t(\epsilon)$ of finding these solutions, as given by (30), is completely similar to the complexity bound (22) of MD.

3.3 Convex minimization with certificates, III: restricted memory Mirror Descent

The fact that the number of linear functions $h_\tau(\cdot)$ involved into the auxiliary problems (25), (27) (and thus computational complexity of these problems) grows as the algorithm proceeds is a serious shortcoming of MDL from the computational viewpoint. NERML (Non-Euclidean Restricted Memory Level Method) algorithm, which originates from [2] is a version of MD "with restricted memory". In this algorithm the number of pieces in the models never exceeds a given number m, a control parameter which can be set to any desired integer value. The original NERML algorithm, however, was not equipped with accuracy certificates, and our goal here is to correct this omission.

3.3.1 Construction

Same as MDL, NERML processes an oracle-represented vector field (14) satisfying the boundedness condition (15), with the ultimate goal to ensure (24). The setup for the algorithm is identical to that for MDL.

The algorithm builds search sequence $y_1 \in Y, y_2 \in Y, ...$ along with the sets $S_\tau = \{y_1, ..., y_\tau\}$, according to the following rules:

A. Initialization. We set $y_1 = y_\omega := \mathrm{argmin}_{y\in Y}\, \omega(y)$, compute $g(y_1)$ and set $f_1 = \max\limits_{y\in Y} h_{y_1}(y)$. We clearly have $f_1 \geq 0$.

- In the case of $f_1 = 0$, we terminate and output $h(\cdot) = h_{y_1}(\cdot) \in \mathcal{F}_{S_1}$, thus ensuring (24) with $\epsilon = 0$.
- When $f_1 > 0$, we proceed. Our subsequent actions are split into *phases* indexed with $s = 1, 2, ...$

B. Phase $s = 1, 2, ...$ At the beginning of phase s, we have at our disposal

- the set $S^s = \{y_1, ..., y_{t_s}\} \subset Y$ of already built search points, and
- an affine function $h^s(\cdot) \in \mathcal{F}_{S^s}$ along with the real $f_s := \max\limits_{y\in Y} h^s(y) \in (0, f_1]$.

We define the *level* ℓ_s of phase s as

$$\ell_s = \gamma f_s,$$

where $\gamma \in (0, 1)$ is a control parameter of the method. Note that $\ell_s > 0$ due to $f_s > 0$.

To save notation, we denote the search points generated at phase s as $u_1, u_2, ...$, so that $y_{t_s+\tau} = u_\tau$, $\tau = 1, 2, ...$

B.1. Initializing phase s**.** We somehow choose collection of m functions $h^s_{0,j}(\cdot) \in \mathcal{F}_{S^s}$, $1 \leq j \leq m$, such that the set

$$Y^s_0 = \mathrm{cl}\,\{y \in Y : h^s_{0,j}(y) > \ell_s, 1 \leq j \leq m\}$$

is nonempty (here a positive integer m is a control parameter of the method).[4] We set

$$u_1 = y_\omega.$$

B.2. Step $\tau = 1, 2, ...$ **of phase** s**:**

B.2.1. At the beginning of step τ, we have at our disposal

[4]Note that to ensure the nonemptiness of Y^s_0, it suffices to set $h^s_{0,j}(\cdot) = h^s(\cdot)$, so that $h_{0,j}(y) > \ell_s$ for $y \in \mathrm{Argmax}_Y\, h^s(\cdot)$; recall that $f_s = \max_{y\in Y} h^s(y) > 0$.

1. the set $S^s_{\tau-1}$ of all previous search points;

2. a collection of functions $\{h^s_{\tau-1,j}(\cdot) \in \mathcal{F}_{S^s_{\tau-1}}\}_{j=1}^m$ such that the set

$$Y^s_{\tau-1} = \mathrm{cl}\,\{x \in Y : h^s_{\tau-1,j}(x) > \ell_s, 1 \le j \le m\}$$

is nonempty,

3. current search point $u_\tau \in Y^s_{\tau-1}$ such that

$$u_\tau = \operatorname*{argmin}_{y \in Y^s_{\tau-1}} \omega(y) \qquad (\Pi^s_\tau)$$

Note that this relation is trivially true when $\tau = 1$.

B.2.2. Our actions at step τ are as follows.
B.2.2.1. We compute $g(u_\tau)$ and set

$$h_{\tau-1,m+1}(y) = \langle g(u_\tau), u_\tau - y\rangle.$$

B.2.2.2. We solve the auxiliary problem

$$\mathrm{Opt} = \max_{y \in Y} \min_{1 \le j \le m+1} h_{\tau-1,j}(y) \qquad (31)$$

Note that

$$\begin{aligned}\mathrm{Opt} &= \max_{y\in Y} \min_{\lambda_j \ge 0, \sum_j \lambda_j = 1} \sum_{j=1}^{m+1} \lambda_j h^s_{\tau-1,j}(y) = \min_{\lambda_j \ge 0, \sum_j \lambda_j = 1} \max_{y \in Y} \sum_{j=1}^{m+1} \lambda_j h^s_{\tau-1,j}(y) \\ &= \max_{y\in Y} \textstyle\sum_{j=1}^{m+1} \lambda^\tau_j h^s_{\tau-1,j}(y),\end{aligned}$$

where $\lambda^\tau_j \ge 0$ and $\sum_{j=1}^{m+1} \lambda^\tau_j = 1$. We assume that when solving the auxiliary problem, we compute the above weights λ^τ_j, and thus have at our disposal the function

$$h^{s,\tau}(\cdot) = \sum_{j=1}^{m+1} \lambda^\tau_j h^s_{\tau-1,j}(\cdot) \in \mathcal{F}_{S^s_\tau}$$

such that

$$\mathrm{Opt} = \max_{y \in Y} h^{s,\tau}(y).$$

B.2.2.3. Case A: If $\mathrm{Opt} \le \epsilon$ we terminate and output $h^{s,\tau}(\cdot) \in \mathcal{F}_{S^s_\tau}$; this function satisfies (24).
Case B: In case of $\mathrm{Opt} < \ell_s + \theta(f_s - \ell_s)$, where $\theta \in (0,1)$ is method's control parameter, we terminate phase s and start phase $s+1$ by setting $h^{s+1} = h^{s,\tau}$, $f_{s+1} = \mathrm{Opt}$. Note that by construction $0 < f_{s+1} \le [\gamma + \theta(1-\gamma)]f_s < f_1$, so that we have at our disposal all we need to start phase $s+1$.
Case C: When neither A nor B takes place, we proceed with phase s, specifically, as follows:
B.2.2.4. Note that there exists a point $u \in Y$ such that $h^s_{\tau-1,j}(u) \ge \mathrm{Opt} > \ell_s$, so that the set $Y_\tau = \{y \in Y : h^s_{\tau-1,j}(y) \ge \ell_s,\ 1 \le j \le m+1\}$, intersects with the relative interior of Y. We specify $u_{\tau+1}$ as

$$u_{\tau+1} = \operatorname*{argmin}_{y \in Y_\tau} \omega(y). \qquad (32)$$

Observe that

$$u_{\tau+1} \in Y^s_{\tau-1} \tag{33}$$

due to $Y_\tau \subset Y^s_{\tau-1}$.

B.2.2.5. By optimality conditions for (32) (see Lemma A.1), for certain nonnegative μ_j, $1 \le j \le m+1$, such that

$$\mu_j[h^s_{\tau-1,j}(u_{\tau+1}) - \ell_s] = 0,\ 1 \le j \le m+1,$$

the vector

$$e := \omega'(u_{\tau+1}) - \sum_{j=1}^{m+1} \mu_j \nabla h^s_{\tau-1,j}(\cdot) \tag{34}$$

is such that

$$\langle e, y - u_{\tau+1}\rangle \ge 0\ \forall y \in Y. \tag{35}$$

• In the case of $\mu = \sum_j \mu_j > 0$, we set

$$h^s_{\tau,1} = \frac{1}{\mu}\sum_{j=1}^{m+1} \mu_j h^s_{\tau-1,j},$$

so that

$$(a)\quad h^s_{\tau,1} \in \mathcal{F}_{S^s_\tau},\quad (b)\quad h^s_{\tau,1}(u_{\tau+1}) = \ell_s,\quad (c)\quad \langle \omega'(u_{\tau+1}) - \mu\nabla h^s_{\tau,1}, y - u_{\tau+1}\rangle \ge 0\,\forall y \in Y \tag{36}$$

We then discard from the collection $\{h^s_{\tau-1,j}(\cdot)\}_{j=1}^{m+1}$ two (arbitrarily chosen) elements and add to $h^s_{\tau,1}$ the remaining $m-1$ elements of the collection, thus getting an m-element collection $\{h^s_{\tau,j}\}_{j=1}^m$ of elements of $\mathcal{F}_{S^s_\tau}$.

> **Remark 3.2** We have ensured that the set $Y^s_\tau = \mathrm{cl}\,\{y \in Y : h^s_{\tau,j}(y) > \ell_s, 1 \le j \le m\}$ is nonempty (indeed, we clearly have $h^s_{\tau,j}(\widehat{u}) > \ell_s$, $1 \le j \le m$, where $\widehat{u}$ is an optimal solution to (31)). Besides this, we also have $(\Pi^s_{\tau+1})$. Indeed, by construction $u_{\tau+1} \in Y_\tau$, meaning that $h^s_{\tau-1,j}(u_{\tau+1}) \ge \ell_s$, $1 \le j \le m+1$. Since $h^s_{\tau,j}$ are convex combinations of the functions $h^s_{\tau-1,j}$, $1 \le j \le m+1$, it follows that $u_{\tau+1} \in Y^s_\tau$. Further, (36.b) and (36.c) imply that $u_{\tau+1} = \mathrm{argmin}_y \left\{\omega(y) : y \in Y,\, h^s_{\tau,1}(y) \ge \ell_s\right\}$, and the right hand side set clearly contains Y^s_τ. We conclude that $u_{\tau+1}$ indeed is the minimizer of $\omega(\cdot)$ on Y^s_τ.

• In the case of $\mu = 0$, (34) – (35) say that $u_{\tau+1}$ is a minimizer of $\omega(\cdot)$ on Y. In this case, we discard from the collection $\{h^s_{\tau-1,j}\}_{j=1}^{m+1}$ one (arbitrarily chosen) element, thus getting the m-element collection $\{h^s_{\tau,j}\}_{j=1}^m$. Here, for exactly the same reasons as above, the set $Y^s_\tau := \mathrm{cl}\,\{y \in Y : h^s_{\tau,j}(y) > \ell_s\}$ is nonempty and contains $u_{\tau+1}$, and, of course, (Π^s_τ) holds true (since $u_{\tau+1}$ minimizes $\omega(\cdot)$ on the entire Y).

In both cases (those of $\mu > 0$ and of $\mu = 0$), we have built the data required to start step $\tau+1$ of phase s, and we proceed to this step.

The description of the algorithm is completed.

Remark 3.3 Same as MDL, the outlined algorithm requires solving at every step two nontrivial auxiliary optimization problems – (31) and (32). It is explained in [2] that these problems are relatively easy, provided that m is moderate (note that this parameter is under our full control) and Y and ω are "simple and fit each other," meaning that we can easily solve problems of the form

$$\min_{x \in Y}\left[\omega(x) + \langle a, x\rangle\right] \tag{$*$}$$

(that is, our proximal setup for Y results in easy-to-compute prox-mapping).

Remark 3.4 By construction, the presented algorithm produces upon termination (if any)

- an execution protocol $y^t = \{y_\tau, g(y_\tau)\}_{\tau=1}^t$, where t is the step where the algorithm terminates, and y_τ, $1 \le \tau \le t$, are the search points generated in course of the run; by construction, all these search points belong to Y;
- an accuracy certificate λ^t – a collection of nonnegative weights $\lambda_1, ..., \lambda_t$ summing up to 1 – such that the affine function $h(y) = \sum_{\tau=1}^t \lambda_\tau \langle g(y_\tau), y_\tau - y\rangle$ satisfies the relation $\epsilon(y^t, \lambda^t) := \max\limits_{x \in Y} h(x) \le \epsilon$, where ϵ is the target tolerance, exactly as required in (24).

3.3.2 Efficiency estimate

Proposition 3.3 *Given on input a target tolerance $\epsilon > 0$, the NERML algorithm terminates after finitely many steps, with execution protocol y^t and accuracy certificate λ^t, described in Remark 3.4. The number of steps of the algorithm does not exceed*

$$N = C(\gamma, \theta)\frac{\Omega^2 L^2[g]}{\epsilon^2}, \text{ where } C(\gamma, \theta) = \frac{(1+\gamma^2)}{\gamma^2[1-[\gamma+(1-\gamma)\theta]^2]}. \tag{37}$$

For proof, see Section A.3.

Remark 3.5 Inspecting the proof of Proposition 3.3, it is immediately seen that when $\omega(\cdot)$ is continuously differentiable on the entire Y, one can replace the rule (32) with

$$u_{\tau+1} = \operatorname*{argmin}_{y \in Y_\tau}[\omega(y) - \langle \omega'(y^s), y - y^s\rangle],$$

where y^s is an arbitrary point of $Y^o = Y$. The cost of this modification is that of replacing Ω in the efficiency estimate with Ω^+, see Remark 3.1. Computational experience shows that a good choice of y^s is the best, in terms of the objective, search point generated before the beginning of phase s.

Solving (P) and (D) by NERML is completely similar to the case of MDL, with the bound

$$t(\epsilon) \le \mathrm{Ceil}\left(\frac{\Omega^2 L_f^2(1+\gamma^2)}{\gamma^2[1-[\gamma+(1-\gamma)\theta]^2\epsilon^2}\right) \quad [\Omega = \Omega[Y, \omega(\cdot)]] \tag{38}$$

in the role of (30).

Remark 3.6 Observe that Propositions 3.1-3.3 do not impose restrictions of the vector field $g(\cdot)$ processed by the respective algorithms aside from the boundedness assumption (15). Invoking Remark 2.1, we arrive at the following conclusion:

in the situation of section 2.1 and given $\delta \ge 0$, let instead of exact maximizers $x(y) \in \mathrm{Argmax}_{x \in X}\langle x, Ay+a\rangle$, approximate maximizers $x_\delta(y) \in X$ such that $\langle x_\delta(y), Ay+a\rangle \ge \langle x(y), Ay+a\rangle - \delta$ for all $y \in Y$ be available. Let also

$$f'_\delta(y) = A^* x_\delta(y) + \psi'(y)$$

be the associated approximate subgradients of the objective f of (D). Assuming

$$L_{f,\delta} = \sup_{y \in Y} \|f'_\delta(y)\| < \infty,$$

let MD/MDL/NERML be applied to the vector field $g(\cdot) = f'_\delta(\cdot)$. Then the number of steps of each method before termination remains bounded by the respective bound (18), (29) or (37), *with $L_{f,\delta}$ in the role of L_f. Beside this, defining the approximate solutions to* (P), (D) *according to* (20), *with $x_\tau = x_\delta(y_\tau)$, we ensure the validity of δ-relaxed version of the accuracy guarantee* (23), *specifically, the relation*

$$[f(\widehat{y}^t) - \mathrm{Opt}(P)] + [\mathrm{Opt}(D) = f_*(\widehat{x}^t] \le \epsilon + \delta.$$

4 An alternative: Smoothing

An alternative to the approach we have presented so far is based on the use of the proximal setup for Y to *smoothen* f_* and then to maximize the resulting smooth approximation of f_* by the Conditional Gradient (CG) algorithm. This approach is completely similar to the one used by Nesterov in his breakthrough paper [11], with the only difference that since in our situation domain X admits LO oracle rather than a good proximal setup, we are bounded to replace the $O(1/t^2)$-converging Nesterov's method for smooth convex minimization with $O(1/t)$-converging CG.

Let us describe the CG implementation in our setting. Suppose that we are given a norm $\|\cdot\|_x$ on E_x, a representation (2) of f_*, a proximal point setup $(\|\cdot\|_y,\ \omega(\cdot))$ for Y and a desired tolerance $\epsilon > 0$. We assume w.l.o.g. that $\min\limits_{y\in Y}\omega(y) = 0$ and set, following Nesterov [11],

$$\begin{array}{c} f_*^\beta(x) = \min\limits_{y\in Y}\left[\langle x, Ay + a\rangle + \psi(y) + \beta\omega(y)\right] \\ \beta = \beta(\epsilon) := \frac{\epsilon}{\Omega^2},\ \Omega = \Omega[Y, \omega(\cdot)] \end{array} \tag{39}$$

From (2), the definition of Ω and the relation $\min_Y \omega = 0$ it immediately follows that

$$\forall x \in X : f_*(x) \le f_*^\beta(x) \le f_*(x) + \frac{\epsilon}{2}, \tag{40}$$

and f_*^β clearly is concave. It is well known[5] that strong convexity, modulus 1 w.r.t. $\|\cdot\|_y$, of $\omega(y)$ implies smoothness of f_*^β, specifically,

$$\forall (x, x' \in X) :\ \|\nabla f_*^\beta(x) - \nabla f_*^\beta(x')\|_{x,*} \le \frac{1}{\beta}\|A\|^2_{y;x,*}\|x - x'\|_x, \tag{41}$$

where

$$\|A\|_{y;x,*} = \max\{\|A^*u\|_{y,*} : u \in E_x, \|u\|_x \le 1\} = \max\{\|Ay\|_{x,*} : y \in E_y, \|y\|_y \le 1\}.$$

Observe also that under the assumption that an optimal solution $y(x)$ of the right hand side minimization problem in (39) is available at a moderate computational cost, [6] we have at our disposal a FO oracle for f_*^β:

$$f_*^\beta(x) = \langle x, Ay(x) + a\rangle + \psi(y(x)) + \beta\omega(y(x)),\quad \nabla f_*^\beta(x) = Ay(x) + a.$$

We can now use this oracle, along with the LO oracle for X, to solve (P) by CG. In the sequel, we refer to the outlined algorithm as to SCGS (Smoothed Conditional Gradient).

[5] To make the paper self-contained, we provide verification in Appendix.

[6] In typical applications, ψ is just linear, so that computing $y(x)$ is as easy as computing the value of the prox-mapping associated with Y, $\omega(\cdot)$,

Efficiency estimate for SCG is readily given by Proposition 2.1. Indeed, assume from now on that X is contained in $\|\cdot\|_x$-ball of radius R of E_x. It is immediately seen that under this assumption, (41) implies the validity of the condition (cf. (3) with $q = 2$)

$$\forall x, x' \in X : f_*^\beta(x') \geq f_*^\beta(x) + \langle \nabla f_*^\beta(x), x' - x\rangle - \tfrac{1}{2}\mathcal{L}\|x' - x\|_x^2, \quad \mathcal{L} = \tfrac{4R^2}{\beta} = \tfrac{4R^2\Omega^2}{\epsilon}. \tag{42}$$

In order to find an ϵ-maximizer of f_*, it suffices, by (40), to find an $\epsilon/2$-maximizer of f_*^β; by (5) (where one should set $q = 2$), what takes

$$t^{\rm SCG}(\epsilon) = O(1)\epsilon^{-1}D = O(1)\frac{\Omega^2\|A\|_{y;x,*}^2}{\epsilon^2} \tag{43}$$

steps.

Discussion. Let us assume, as above, that X is contained in the centered at the origin ball of radius R, and let us compare the essentially identical to each other[7] complexity bounds (22), (30), (38), with the bound (43). Under the natural assumption that the subgradients of ψ we use satisfy the bounds $\|\psi'(y)\|_{y,*} \leq L_\psi$, where L_ψ is the Lipschitz constant of ψ w.r.t. the norm $\|\cdot\|_y$, (19) implies that

$$L_f \leq \|A\|_{y;x,*}R + L_\psi. \tag{44}$$

Thus, the first three complexity bounds reduce to

$$\mathcal{C}_{\rm MD}(\epsilon) = {\rm Ceil}\left(O(1)\frac{[R\|A\|_{y;x,*} + L_\psi]^2\Omega^2[Y,\omega(\cdot)]}{\epsilon^2}\right), \tag{45}$$

while the conditional gradients based complexity bound is

$$\mathcal{C}_{\rm SCG}(\epsilon) = {\rm Ceil}\left(O(1)\frac{[R\|A\|_{y;x,*}]^2\Omega^2[Y,\omega(\cdot)]}{\epsilon^2}\right). \tag{46}$$

We see that *assuming* $L_\psi \leq O(1)R\|A\|_{y;x,*}$ (which indeed is the case in many applications, in particular, in the examples we are about to consider), *the complexity bounds in question are essentially identical.* This being said, we believe that the two approaches in question seem to have their own advantages and disadvantages. Let us name just a few:

- Formally, the SCG has a more restricted area of applications than MD/MDL/NERML, since relative simplicity of the optimization problem in (39) is a more restrictive requirement than relative simplicity of computing prox-mapping associated with $Y, \omega(\cdot)$. At the same time, in most important applications known to us ψ is just linear, and in this case the just outlined phenomenon disappears.

- An argument in favor of SCG, is its insensitivity to the Lipschitz constant of ψ. Note, however, that in the case of linear ψ (which, as we have mentioned, is the case of primary interest) the nonsmooth techniques admit simple modifications (not to be considered here) which make them equally insensitive to L_ψ.

- Our experience shows that the convergence pattern of nonsmooth methods utilizing memory (MDL and NERML) is, at least at the beginning of the solution process, much better than is predicted by their worst-case efficiency estimates. It should be added that in theory

[7] provided the parameters $\gamma \in (0,1)$, $\theta \in (0,1)$ in (30), (38) are treated as absolute constants.

there exist situations where the nonsmooth approach "most probably," or even provably, significantly outperforms the smooth one. This is the case when E_y is of moderate dimension. A well-established *experimental* fact is that when solving (D) by MDL, every $\dim E_y$ iterations of the method reduce the inaccuracy by an absolute constant factor, something like 3. It follows that if n is in the range of few hundreds, a couple of thousands of MDL steps can yield a solution of accuracy which is incomparably better than the one predicted by the theoretical worst-case oriented $O(1/\epsilon^2)$ complexity bound of the algorithm. Moreover, in principle one can solve (D) by the Ellipsoid method with certificates [10], building accuracy certificate of resolution ϵ in *polynomial time* $O(1)n^2\ln(L_f\Omega[Y,\omega(\cdot)/\epsilon)$. It follows that when $\dim E_y$ is in the range of few tens, the nonsmooth approach allows to solve, in moderate time, problems (P) and (D) to high accuracy. Note that low dimensionality of E_y by itself does not prevent X to be high-dimensional and "difficult;" how frequent are these situations in actual application, this is another story.

We believe that the choice of one, if any, of the outlined approaches to use, is the issue which should be resolved, on the case-by-case basis, by computational practice. We believe, however, that it makes sense to keep them both in mind.

5 Application examples

In this section we work out some application examples, with the goal to demonstrate that the approach we are proposing possesses certain application potential.

5.1 Uniform norm matrix completion

Our first example (for its statistical motivation, see [6]) is as follows: given a symmetric $p\times p$ matrix b and a positive real R, we want to find the best entrywise approximation of B by a positive semidefinite matrix x of given trace R, that is, to solve the problem

$$\begin{array}{c}\min_{x\in X}\left[-f_*(x)=\|x-b\|_\infty\right]\\ X=\{x\in\mathbf{S}^p:\ x\succeq 0,\ \mathrm{Tr}(x)=R\},\ \|x\|_\infty=\max\limits_{1,j\le p}|x_{ij}|\end{array}\qquad(47)$$

where $\mathbf{S}^p$ is the space of $p\times p$ symmetric matrices. Note that with our X, computing prox-mappings associated with all known proximal setups needs eigenvalue decomposition of an $p\times p$ symmetric matrix and thus becomes computationally demanding in the large scale case. On the other way, to maximize a linear form $\langle\xi,x\rangle=\mathrm{Tr}(\xi x)$ over $x\in X$ requires computing the maximal eigenvalue of ξ along with corresponding eigenvector. In the large scale case this task is by orders of magnitude less demanding than computing full eigenvalue decomposition. Note that our f_* admits a simple Fenchel-type representation:

$$f_*(x)=-\|x-b\|_\infty=\min_{y\in Y}\left[f(y)=\langle -x,y\rangle+\langle b,y\rangle\right],\ Y=\{y\in\mathbf{S}^p:\|y\|_1:=\sum_{i,j}|y_{ij}|\le 1\}.$$

Equipping $E_y=\mathbf{S}^p$ with the norm $\|\cdot\|_y=\|\cdot\|_1$, and Y with the d.-g.f.

$$\omega(y)=\alpha\ln(p)\sum_{i,j=1}^p|y_{ij}|^{1+r(p)},\ r(p)=\frac{1}{\ln(p)},$$

where α is an appropriately chosen constant of order of 1 (induced by the necessity to make $\omega(\cdot)$ strongly convex, modulus 1, w.r.t. $\|\cdot\|_1$), we get a proximal setup for Y such that

$$\Omega[Y,\omega(\cdot)]\le O(1)\sqrt{\ln p}.$$

We see that our problem of interest fits well the setup of methods developed in this paper. Invoking the bounds (45), (46), we conclude that (47) can be solved within accuracy ϵ in at most

$$t(\epsilon) = O(1)\frac{[R + \|b\|_\infty]^2 \ln(p)}{\epsilon^2} \tag{48}$$

steps by any of methods MD, MDL or NERML, and in at most $O(1)\frac{R^2 \ln(p)}{\epsilon^2}$ steps by SCG.

It is worth to mention that in the case in question, the algorithms yielded by the nonsmooth approach admit a "sparsification" as follows. We are in the case of $\langle x, Ay + a\rangle \equiv \mathrm{Tr}(xy)$, and $X = \{x : x \succeq 0, \mathrm{Tr}(x) = R\}$, so that $x(y) = Re_y e_y^T$, where e_y is the leading eigenvector of a matrix y normalized to have $\|e_y\|_2 = 1$. Given a desired accuracy $\epsilon > 0$ and a unit vector $e_{y,\epsilon}$ such that $\mathrm{Tr}(y[e_{y,\epsilon}e_{y,\epsilon}^T]) \geq \mathrm{Tr}(y[e_y e_y^T]) - R^{-1}\epsilon$, and setting $x_\epsilon(y) = Re_{y,\epsilon}e_{y,\epsilon}^T$, we ensure that $x_\epsilon(y) \in X$ and that $x_\epsilon(y)$ is an ϵ-maximizer of $\langle x, Ay + a\rangle$ over $x \in X$. Invoking Remark 3.6, we conclude that when utilizing $x_\epsilon(\cdot)$ in the role of $x(\cdot)$, we get 2ϵ-accurate solutions to (P), (D) in no more than $t(\epsilon)$ steps. Now, we can take as $e_{y,\epsilon}$ the normalized leading eigenvector of an arbitrary matrix $\widehat{y}(y)$ such that $\|\sigma(\widehat{y} - y)\|_\infty \leq \epsilon$. Assuming $R/\epsilon > 1$ and given $y \in Y$, let us sort the magnitudes of entries in y and build y_ϵ by "thresholding" – by zeroing out as many smallest in magnitude entries as is possible under the restriction that the remaining part of the matrix y is symmetric, and the sum of squares of the entries we have replaced with zeros does not exceed $R^{-2}\epsilon^2$. Since $\|y\|_1 \leq 1$, the number N_ϵ of nonzero entries in y_ϵ is at most $O(1)R^2/\epsilon^2$. On the other hand, by construction, the Frobenius norm $\|\sigma(y - y_\epsilon)\|_2$ of $y - y_\epsilon$ is $\leq R^{-1}\epsilon$, thus $\|\sigma(y - y_\epsilon)\|_\infty \leq R^{-1}\epsilon$, and we can take as $e_{y,\epsilon}$ the normalized leading eigenvector of y_ϵ. When the size p of y is $\gg R^2/\epsilon^2$ (otherwise the outlined sparsification does not make sense), this approach reduces the problem of computing the leading eigenvector to the case when the matrix is question is relatively sparse, thus reducing its computational cost.

5.2 Nuclear norm SVM

Our next example is as follows: we are given an N-element sample of $p \times q$ matrices z_j ("images") equipped with labels $\epsilon_j \in \{-1, 1\}$. We assume the images to be normalized by the restriction

$$\|\sigma(z_j)\|_\infty \leq 1. \tag{49}$$

We want to find a linear classifier of the form

$$\widehat{\epsilon}_j = \mathrm{sign}(\langle x, z_j\rangle + b). \qquad [\langle z, x\rangle = \mathrm{Tr}(zx^T)]$$

which predicts well labels of images. The "low-rank-oriented" SVM-based setting of this problem is

$$\min_{x:\|\sigma(x)\|_1 \leq R} \left[h(x) := \min_{b \in \mathbf{R}} \left[N^{-1}\sum_{j=1}^N [1 - \epsilon_j [\langle x, z_j\rangle + b]]_+\right]\right], \tag{50}$$

where $\|\sigma(\cdot)\|_1$ is the nuclear norm, $[a]_+ = \max[a, 0]$, and $R \geq 1$ is a parameter.[8]

In this case the domain X of problem (P) is the ball of the nuclear norm in the space $\mathbf{R}^{p\times q}$ of $p \times q$ matrices and p, q are large. As we have explained in the introduction, same as in the example of the previous section, in this case the computational complexity of LO oracle is much

[8] The restriction $R \geq 1$ is quite natural. Indeed, with the optimal choice of x, we want most of the terms $[1 - \epsilon_j [\langle x, z_j\rangle + b]]_+$ to be $\ll 1$; assuming that the number of examples with $\epsilon_j = -1$ and $\epsilon_j = 1$ are of order of N, this condition can be met only when $|\langle x, z_j\rangle|$ are at least of order of 1 for most of j's. The latter, in view of (49), implies that $\|\sigma(x)\|_1$ should be at least $O(1)$.

smaller than the complexity of computing prox-mapping. Thus, from practical viewpoint, in a meaningful range of values of p, q the LO oracle is "affordable," while the prox-mapping is not.

Observing that $[a]_+ = \max_{0\le y\le 1} ya$, and denoting $\mathbf{1} = [1; ...; 1] \in \mathbf{R}^q$, we get

$$N^{-1}\sum_{j=1}^{N}\left[1-\epsilon_j\left[\langle x, z_j\rangle + b\right]\right]_+ = \max_{y:0\le y\le \mathbf{1}}\left\{N^{-1}\sum_{j=1}^{N} y_j\left[1-\epsilon_j\left[\langle x, z_j\rangle + b\right]\right]\right\},$$

whence

$$h(x) = \max_{y\in Y} N^{-1}\sum_{j=1}^{N} y_j\left[1-\epsilon_j\langle x, z_j\rangle\right], \tag{51}$$

where

$$Y = \{y\in\mathbf{R}^N : 0\le y\le \mathbf{1}, \sum_j \epsilon_j y_j = 0\}; \tag{52}$$

from now on we assume that $Y\neq\emptyset$. When setting

$$\mathcal{A}y = N^{-1}\sum_{j=1}^{N} y_j\epsilon_j z_j : \mathbf{R}^N\to\mathbf{R}^{p\times q},\ X = \{x\in\mathbf{R}^{p\times q} : \|\sigma(x)\|_1\le R\},\ \psi(y) = -N^{-1}\mathbf{1}^T y \tag{53}$$

and passing from minimizing $h(x)$ to maximizing $f_*(x)\equiv -h(x)$, problem (50) becomes

$$\max_{x\in X}\left[f_*(x) := \min_{y\in Y}\left[\langle x, \mathcal{A}y\rangle + \psi(y)\right]\right]. \tag{54}$$

Let us equip $E_y = \mathbf{R}^N$ with the standard Euclidean norm $\|\cdot\|_2$, and Y - with the Euclidean d.-g.f. $\omega(y) = \frac{1}{2}y^T y$. Observe that

$$\left[f'(y)\right]_j = N^{-1}\left[\langle x(y), \epsilon_j z_j\rangle - 1\right],\ \ x(y)\in \operatorname*{Argmax}_{x\in X}\operatorname{Tr}(xy^T),$$

meaning that

$$\|f'(y)\|_\infty \le \max_j\left[N^{-1}\left[1+\|\sigma(x(y))\|_1\|\sigma(z_j)\|_\infty\right]\right] \le N^{-1}[R+1]\le 2N^{-1}R.$$

Using our notation of section 3 we have

$$L_f := \sup_{y\in Y}\|f'(y)\|_* \le 2N^{-1/2}R$$

(we are in the case of $\|\cdot\|_* = \|\cdot\|_2$), and, besides,

$$\Omega_Y \le \sqrt{N/2}.$$

We conclude that for every $\epsilon > 0$, the number t of MD steps needed to ensure (23) does not exceed

$$t_{\mathrm{MD}}(\epsilon) = \mathrm{Ceil}\left(\frac{2R^2}{\epsilon^2}\right)$$

(see (22)), and similarly for MDL, NERML, and SCG.

5.3 Multi-class classification under $\infty|2$ norm constraint

Our last example illustrates the potential of the proposed approach in the case when the domain X of (P) does not admit a proximal setup with "moderate" Ω_X. Namely, let (P) be the problem

$$\min_{x\in X}[-f_*(x) := \|Bx-b\|_{y,*}] \qquad [x\mapsto Bx : E_x \to E_y] \tag{55}$$

where $\|\cdot\|_{y,*}$ is the norm conjugate to a norm $\|\cdot\|_y$ on E_y. We are interested in the case of box-type X, specifically,

$$X = \{[x^1;...;x^M] \in \mathbf{R}^{n_1}\times ...\times \mathbf{R}^{n_M} : \|x^i\|_2 \le R,\, 1\le i\le M\} \tag{56}$$

As it was mentioned in Introduction, for every proximal setup $(\|\cdot\|, \omega_x(\cdot))$ for X which is normalized by the requirement that simple "well behaved" on X convex functions should have moderate Lipschitz constants w.r.t. $\|\cdot\|$ (specifically, the coordinates of $x\in X$ should have Lipschitz constants ≤ 1), one has $\Omega[X,\omega_x(\cdot)] \ge O(1)\sqrt{M}R$. As a result, the theoretical complexity of the FO methods as applied to (55) grows with M at the rate at least $O(\sqrt{M})$, thus becoming prohibitively high for large M. We are about to show that the approaches developed in this paper are free of this shortcoming. Specifically, we can easily build a Fenchel-type representation of f_*:

$$f_*(x) = -\|Bx-b\|_{y,*} = \min_{y\in Y}\left[\langle B^*y, x\rangle - \langle b, y\rangle\right],\ Y = \{y\in E_y : \|y\|_y \le 1\}.$$

Assume that Y admits a good proximal setup. We can augment $\|\cdot\|_y$ with a d.-g.f. $\omega_y(\cdot)$ for Y such that $\|\cdot\|_y, \omega(y)$ form a proximal setup, and applying any of the methods we have developed in sections 3 and 4, the complexity of finding ϵ-solution to (55) by any of these methods becomes

$$O(1)\left(\frac{R\|B\|_{x;y,*}+\|b\|_{y,*}}{\epsilon}\right)^2,\ \|B\|_{x;y,*} = \max_{x=[x^1;...;x^M]}\left\{\|Bx\|_{y,*} : \|x\|_{\infty|2} := \max_i \|x^i\|_2 \le 1\right\}.$$

Note that in this bound M does not appear, at least explicitly.

Multi-class classification problem we consider is as follows: we observe N "feature vectors" $z_j\in\mathbf{R}^q$, each belonging to one of M non-overlapping classes, along with labels $\chi_j\in\mathbf{R}^M$ which are basic orths in $\mathbf{R}^M$; the index of the (only) nonzero entry in χ_j is the number of class to which z_j belongs. We want to build a multi-class analogy of the standard linear classifier as follows: a multi-class classifier is specified by a matrix $x\in\mathbf{R}^{M\times q}$ and a vector $b\in\mathbf{R}^M$. Given a feature vector z, we compute the M-dimensional vector $xz+b$, identify its maximal component, and treat the index of this component as our guess for the serial number of the class to which z belongs.

The multi-class analogy of the usual approach to building binary classifiers by minimizing the empirical hinge loss is as follows [3, 1]. Let $\bar\chi_j = \mathbf{1}-\chi_j$ be the "complement" of χ_j.Given a feature vector z and the corresponding label χ, let us set

$$h = h(x,b;z,\chi) = [xz+b] - [\chi^T[xz+b]]\mathbf{1} + \bar\chi \in \mathbf{R}^M \qquad [\mathbf{1} = [1;...;1]\in\mathbf{R}^M].$$

Note that if i_* is the index of the only nonzero entry in χ, then the i_*-th entry in h is zero (since $\chi_{i_*}=1$). Further, h is nonpositive if and only if the classifier, given by x,b and evaluated at z, "recovers the class i_* of z with margin 1", i.e., we have $[xz+b]_j \le [xz+b]_{i_*}-1$ for $j\ne i_*$. On the other hand, if the classifier fails to classify z correctly (that is, $[xz+b]_j \ge [xz+b]_{i_*}$ for some $j\ne i_*$), then the maximal entry in h is ≥ 1. Altogether, when setting

$$\eta(x,b;z,\chi) = \max_{1\le j\le M}[h(x,b;z,\chi)]_j,$$

we get a nonnegative function which vanishes for the pairs (z,χ) which are "quite reliably" – with margin ≥ 1 – classified by (x,b), and is ≥ 1 for the pairs (z,χ) with z not classified correctly. Thus the function

$$H(x,b) = \mathbf{E}\{\eta(x,b;z,\chi)\},$$

the expectation being taken over the distribution of examples (z,χ), is an upper bound on the probability for classifier (x,b) to misclassify a feature vector. What we would like to do now is to minimize $H(x,b)$ over x,b. To do this, since $H(\cdot)$ is not observable, we replace the expectation by its empirical counterpart

$$H_N(x,b) = N^{-1}\sum_{j=1}^{N}\eta(x,b;z_j,\chi_j).$$

For the sake of simplicity (and, upon a close inspection, without much harm), we assume from now on that $b=0$.[9] Imposing, as it is always the case in hinge loss optimization, an upper bound on some norm $\|x\|_x$ of x, we arrive at the optimization problem

$$\min_{x\in X}\left[-f_*(x) = N^{-1}\sum_{j=1}^{N}\max_{i\leq M}[xz_j - [\chi_j^T xz_j]\mathbf{1} + \bar{\chi}_j]_i\right], \quad X = \{x : \|x\|_x \leq R\}. \tag{57}$$

From now on we assume that z_j's are normalized:

$$\|z_j\|_2 \leq 1,\ 1\leq j\leq N. \tag{58}$$

Under this constraint, a natural (although not the only meaningful) choice of the norm $\|\cdot\|_x$ is the maximum of the $\|\cdot\|_2$-norms of the rows $[x^i]^T$ of x. If we identify x with the vector $[x^1;...;x^M]$, X becomes the set (56) with $n_1=n_2=...=n_M=q$, and the norm $\|\cdot\|_x$ becomes $\|\cdot\|_{\infty|2}$. The same argument as in the previous section allows us to assume that $R\geq 1$.

Noting that $\max\limits_i h_i = \max_u\{u^T h : u\geq 0, \sum_i u_i = 1\}$, (57) can be rewritten as

$$\max_{x\in X}\left[f_*(x) = \min_{y\in Y}[\langle y, Bx\rangle + \psi(y)]\right] \tag{59}$$

where

$$\begin{array}{l} Y = \{y = [y^1;...;y^N] : y^j\in\mathbf{R}^M_+,\ \sum_i [y^j]_i = N^{-1},\ 1\leq j\leq N\} \in E_y = \mathbf{R}^{MN} \\ Bx = [B^1x;...;B^Nx],\ B^jx = \left[z_j^T[x^{i(j)}-x^1];...;z_j^T[x^{i(j)}-x^M]\right],\ j=1,...,N \\ \psi(y) = \psi(y^1,...,y^N) = -\sum_{j=1}^N [y^j]^T\bar{\chi}_j; \end{array}$$

(here $i(j)$ is the class of z_j, i.e., the index of the only nonzero entry in χ_j). Note that Y is a part of the standard simplex $\Delta_{MN} = \{y\in\mathbf{R}^{MN}_+ : \sum_{j=1}^N\sum_{i=1}^M [y^j]_i = 1\}\subset E_y = \mathbf{R}^{MN}$. Equipping E_y with the norm $\|\cdot\|_y = \|\cdot\|_1$ (so that $\|\cdot\|_{y,*} = \|\cdot\|_\infty$), and Y – with the entropy d.-g.f.

$$\omega_y(y) = \sum_{j=1}^N\sum_{i=1}^M [y^j]_i\ln([y^j]_i)$$

[9] To arrive at this situation, one can augment z_j by additional entry, equal to 1, and to redefine x: the new x is the old $[x,b]$.

(known to complete $\|\cdot\|_1$ to a proximal setup for Δ_{MN}), we get a proximal setup for Y with $\Omega = \Omega[Y, \omega(\cdot)] \le \sqrt{2\ln(M)}$. Next, assuming $\|x\|_x \equiv \|x\|_{\infty|2} \le 1$, we have

$$\|Bx\|_{y,*} = \|Bx\|_\infty = \max_{\substack{1\le i\le m\\ 1\le j\le N}} |z_j^T[x^{i(j)} - x^i]| \le \|z_j\|_2 \|x^{i(j)} - x^i\|_2 \le 2,$$

so that $\|B\|_{x;y,*} \le 2$. Furthermore, ψ clearly is Lipschitz continuous with constant 1 w.r.t. $\|\cdot\|_y = \|\cdot\|_1$. It follows that the complexity of finding an ϵ-solution to (55) by MD, MDL, NERML or SCG is bounded by $O(1)\frac{R^2\ln(M)}{\epsilon^2}$ (see (44), (45), (46) and take into account that $R \ge 1$ and that what is now called B, was called A^* in the notation used in those bounds, so that $\|B\|_{x;y,*} = \|A\|_{y;x,*}$). Note that the resulting complexity bound is independent of N and is "nearly independent" of M. Finally, prox-mapping for Y is given by a closed form expression and can be computed in linear time:

$$\begin{array}{l}\operatorname*{argmin}\limits_{\{y^j\in\mathbf{R}^M\}_{j=1}^N} \left\{\sum_{j=1}^N\sum_{i=1}^M [y^j]_i \ln([y^j]_i) + \sum_{j=1}^N \langle \xi^j, y^j\rangle : y^j \ge 0, \sum_{i=1}^M [y^j]_i = 1,\ 1\le j\le N\right\}\\ = \left\{\widehat{y}^j : [\widehat{y}^j]_i = \frac{\exp\{-[\xi^j]_i\}}{N\sum_{s=1}^M \exp\{-[\xi^j]_s\}},\ 1\le i\le M\right\}_{j=1}^N.\end{array}$$

References

[1] Y. Amit, M. Fink, N. Srebro, and S. Ullman. "Uncovering shared structures in multiclass classification" – In *ICML 2007* (2007), 17-24.

[2] Ben-Tal, A., Nemirovski, A. "Non-Euclidean restricted memory level method for large-scale convex optimization" – *Mathematical Programming* **102** (2005), 407–456.

[3] Crammer, K., Singer, Y. "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines" – *Journ. Mach. Learn. Res.* **2** (2001), 265-292.

[4] V. Demyanov and A. Rubinov. *Approximate Methods in Optimization Problems*. American Elsevier, 1970.

[5] J. C. Dunn and S. Harshbarger. "Conditional gradient algorithms with open loop step size rules" – *Journal of Mathematical Analysis and Applications*, 62(2):432–444, 1978.

[6] Fan, J., Liao, Y., and Mincheva, M. "High Dimensional Covariance Matrix Estimation in Approximate Factor Models" – *Annals of Stat.,* **39** (2011), 3320-3356.

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

[8] Juditsky, A., Nemirovski, A., "First Order Methods for Nonsmooth Large-Scale Convex Minimization, I: General Purpose Methods; II: Utilizing Problem's Structure" – S. Sra, S. Nowozin, S. Wright, Eds., *Optimization for Machine Learning*, The MIT Press, 2012, 121-254.

[9] Lemarechal, C., Nemirovski, A., and Nesterov, Yu. 'New variants of bundle methods" – *Mathematical Programming* **69:1** (1995), 111–148.

[10] Nemirovski, A., Onn, S., Rothblum, U., "Accuracy certificates for computational problems with convex structure" – *Mathematics of Operations Research* **35:1** (2010), 52–78.

[11] Nesterov. Yu., "Smooth minimization of non-smooth functions" – *Mathematical Programming*, **103** (2005), 127–152.

[12] Nesterov. Yu., "Primal-dual subgradient methods for convex problems" – *Math. Program., Ser. B* **120** (2009), 221259.

[13] B. Pshenichnyj and Y. Danilin. *Numerical Methods in Extremal Problems*. Mir, 1978.

A Appendix: Proofs

A.1 Lemma on Prox-mapping

Lemma A.1 . *Let Y be a nonempty closed and bounded subset of an Euclidean space E_y, and let $\|\cdot\|$, $\omega(\cdot)$ be the corresponding proximal setup. Let, further, U be a closed convex subset of Y intersecting the relative interior of Y, and let $p \in E_y$.*
(i) *The optimization problem*

$$\min_{y\in U} h_p(y) := [\langle p, y\rangle + \omega(y)]$$

has a unique solution y_*. This solution is fully characterized by the inclusion $y_* \in U \cap Y^o$, $Y^o = \{y \in Y : \partial\omega(y) \neq \emptyset\}$, coupled with the relation

$$\langle p + \omega'(y_*), u - y_*\rangle \geq 0 \ \ \forall u \in U. \tag{60}$$

(ii) *When U is cut off Y by a system of linear inequalities $e_i(y) \leq 0$, $i = 1, ..., m$, there exist Lagrange multipliers $\lambda_i \geq 0$ such that $\lambda_i e_i(y_*) = 0$, $1 \leq i \leq m$, and*

$$\forall u \in Y : \ \ \langle p + \omega'(y_*) + \sum_{i=1}^{m} \lambda_i e_i', u - y_*\rangle \geq 0. \tag{61}$$

(iii) *In the situation of* (ii)*, assuming $p = \xi - \omega'(y)$ for some $y \in Y^o$, we have*

$$\forall u \in Y : \langle \xi, y_* - u\rangle - \sum_i \lambda_i e_i(u) \leq V_y(u) - V_{y_*}(u) - V_y(y_*). \tag{62}$$

Proof. (i): In one direction the statement is evident: if $y_* \in U \cap X^o$ satisfies (60), then clearly y_* minimizes $h_p(\cdot)$ on U. Further, the minimizer of h_p on U clearly exists and is unique due to the strong convexity of $h_p(\cdot)$. Thus, all we need to prove is that the minimizer y_* of h_p over U belongs to Y^o and satisfies (60). We can assume w.l.o.g. that Y spans the entire E_y and thus its relative interior is the same as its interior. Let $y \in U \cap \operatorname{int} Y$, and let $y_t = y_* + t(y - y_*)$, so that $y_t \in U \cap \operatorname{int} Y \subset Y^o$ for $0 < t \leq 1$. Consequently, the function $\phi(t) = h_p(y_t)$ is convex, continuously differentiable on $(0, 1]$ with the derivative $\phi'(t) = \langle \omega'(y_t) + p, y - y_*\rangle$, and attains its minimum on $[0, 1]$ at $t = 0$, whence $\phi'(t) \geq 0$ for $0 < t \leq 1$. Now let $r > 0$ be such that the Euclidean ball B of radius r centered at y is contained in Y. Since h_p is continuous on Y, we have $h_p(z) - h_p(y_t) \leq V < \infty$ for all $z \in B$ and all $t \in [0, 1]$, whence $\langle p + \omega'(y_t), z - y_t\rangle \leq V$ for all $t \in (0, 1]$ and all $z \in B$. On the other hand,

$$V \geq \max_{z\in B} \langle p + \omega'(y_t), z - y_t\rangle = \langle p + \omega'(y_t), y - y_t\rangle + r\|p + \omega'(y_t)\|_2 = \phi'(t) + r\|p + \omega'(y_t)\|_2$$
$$\geq r\|p + \omega'(y_t)\|_2.$$

We see that $\omega'(y_t)$ remains bounded as $t \to 0$. Thus there exists a sequence $t_i \to +0$, $i \to \infty$, and $e \in E_y$ such that $\omega'(y_{t_i}) \to e$ as $i \to \infty$. Since ω is continuous on Y and $y_{t_i} \to y_*$ as

$i \to \infty$, $e \in \partial\omega(y_*)$, that is, $y_* \in Y^o$, whence in fact $\omega'(y_t) \to \omega'(y_*)$, $t \to +0$, and thus $\langle p+\omega'(y_*), y-y_*\rangle = \lim_{t\to+0}\phi'(t) \geq 0$. Thus, $y_* \in Y^o$ and (60) holds true for all $u \in U \cap \operatorname{int} Y$. The latter set is dense in U, so that (60) holds true for all $u \in U$, as claimed in (i).

(ii): Let $K = \operatorname{cl}\{d : y_* + d \in Y\}$ be the tangential cone of Y at y_*, and

$$L = \{d : e_i(y_* + d) \leq 0 \ \ \forall i \in I = \{i : e_i(y_*) = 0\}\}.$$

Since $U = \{y \in Y : e_i(y) \leq 0\}$ intersects int Y, the interior of the cone K intersects with L. This combines with (60) to ensure that the vector $p + \omega'(y_*)$ belongs to the cone dual to $K \cap L$. By the Dubovitski-Milutin lemma this implies that $p+\omega'(y_*)$ belongs to the arithmetics sum of the cones dual to K and to L, meaning that for some nonnegative λ_i, $i \in I$, $p + \sum_{i\in I}\lambda_i e_i' + \omega'(y_*)$ belongs to the cone dual to K, which is exactly what is stated in (ii).

(iii): With I and $\lambda_i, i \in I$ as described in the proof of (ii), we have

$$\begin{array}{ll} & \forall u \in Y: \ \ \langle \xi - \omega'(y) + \sum_{i\in I}\lambda_i e_i' + \omega'(y_*), u - y_*\rangle \geq 0 \ \text{[see (61)]} \\ \Rightarrow & \forall u \in Y: \ \ \langle \xi + \sum_{i\in I}\lambda_i e_i', y_* - u\rangle \leq \langle \omega'(y_*) - \omega'(y), y - y_*\rangle = V_y(u) - V_{y_*}(u) - V_y(y_*), \end{array}$$

where the concluding equality is readily given by the definition of $V_\cdot(\cdot)$ □

Applying item (i) of Lemma A.1 with $e_j \equiv 0$ (i.e., with $U = Y$), we arrive at $\langle \omega'(y_\omega), y - y_\omega\rangle \geq 0$ for all $y \in Y$, as required in (10). Applying item (iii) to the case of $e_i(u) \equiv 0$ for all u, we get (13).

A.2 Proof of Proposition 3.2

1^0. Observe that when t is a non-terminal step of the algorithm, the level set U_t is a closed convex subset of Y which intersects the relative interior of Y; indeed, by (26) and due to $\epsilon_t > 0$ for a non-terminal t, there exists $y \in Y$ such that $h_\tau(y) \geq \epsilon_t > \gamma\epsilon_t$, $\tau \in S_t$, that is, U_t is cut off Y by a system of constraints satisfying the Slater condition. Denoting $S_t = \{y_{\tau_1}, y_{\tau_2}, ..., y_{\tau_k}\}$ and invoking item (iii) of Lemma A.1 with $y = \widehat{y}_t$, $\xi = 0$ and $e_i(\cdot) = \gamma\epsilon_t - h_{\tau_i}(\cdot)$ (so that $U_t = \{u \in Y : e_i(u) \leq 0, 1 \leq i \leq k\}$, we get $y_* = y_{t+1}$ and

$$-\sum_i \lambda_i e_i(u) \leq V_{\widehat{y}_t}(u) - V_{y_{t+1}}(u) - V_{\widehat{y}_t}(y_{t+1}).$$

with some $\lambda_i \geq 0$. When $u \in U_t$, we have $e_i(u) \leq 0$, that is,

$$\forall u \in U_t : V_{y_{t+1}}(u) \leq V_{\widehat{y}_t}(u) - V_{\widehat{y}_t}(y_{t+1}). \tag{63}$$

2^0. When t starts a phase, we have $\widehat{y}_t = y_\omega = y_1$, and clearly $1 \in S_t$, whence $h_\tau(\widehat{y}_t) \leq 0$ for some $\tau \in S_t$ (specifically, for $\tau = 1$). When t does not start a phase, we have $\widehat{y}_t = y_t$ and $t \in S_t$, so that here again $h_\tau(\widehat{y}_t) \leq 0$ for some $\tau \in S_t$. On the other hand, $h_\tau(y_{t+1}) \geq \gamma\epsilon_t$ for all $\tau \in S_t$ due to $y_{t+1} \in U_t$. Thus, when passing from $\widehat{y}_t$ to y_{t+1}, at least one of $h_\tau(\cdot)$ grows by at least $\gamma\epsilon_t$. Taking into account that $h_\tau(z) = \langle g(y_\tau), y_\tau - z\rangle$ is Lipschitz continuous with constant $L[g]$ w.r.t. $\|\cdot\|$ (by (15)), we conclude that $\|\widehat{y}_t - y_{t+1}\| \geq \gamma\epsilon_t/L[g]$. With this in mind, (63) combines with (9) to imply that

$$\forall u \in U_t : V_{y_{t+1}}(u) \leq V_{\widehat{y}_t}(u) - \frac{1}{2}V_{\widehat{y}_t}(y_{t+1}) \leq V_{\widehat{y}_t}(u) - \frac{\gamma^2\epsilon_t^2}{2L^2[g]}. \tag{64}$$

3^0. Let the algorithm perform phase s, let t_s be the first step of this phase, and r be another step of the phase. We claim that all level sets U_t, $t_s \leq t \leq r$, have a point in common, specifically, (any) $u \in \operatorname{Argmax}_{y\in Y}\min_{\tau\in S_r} h_\tau(y)$. Indeed, since r belongs to phase s, we have

$$\gamma\Delta_s < \epsilon_r = \max_{y\in Y}\min_{\tau\in S_r} h_\tau(y) = \min_{\tau\in S_r} h_\tau(u)$$

and $\Delta_s = \epsilon_{t_s} = \max_{y\in Y}\min_{\tau\in S_{t_s}} h_\tau(y)$ (see (26) and the definition of Δ_s). Besides this, r belongs to phase s, and within a phase, sets S_t extend as t grows, so that $S_{t_s} \subset S_t \subset S_r$ when $t_s \le t \le r$, implying that $\epsilon_{t_s} \ge \epsilon_{t_s+1} \ge ... \ge \epsilon_r$. Thus, for $t \in \{t_s, s_{s+1}, ..., r\}$ we have

$$\min_{\tau\in S_t} h_\tau(u) \ge \min_{\tau\in S_r} h_\tau(u) \ge \gamma\Delta_s = \gamma\epsilon_{t_s} \ge \gamma\epsilon_t,$$

implying that $u \in U_t$.

With the just defined u, let us look at the quantities $v_t := V_{\widehat{y}_t}(u)$, $t_s \le t \le r$. We have $v_{t_s} \le \frac{1}{2}\Omega^2$ due to $\widehat{y}_{t_s} = y_\omega$ and (12), and

$$0 \le v_{t+1} \le v_t - \frac{\gamma^2\epsilon_t^2}{2L^2[g]} \le v_t - \frac{\gamma^4\Delta_s^2}{2L^2[g]}$$

when $t_s \le t < r$ (due to (64) combined with $\widehat{y}_t = y_t$ when $t_s < t \le r$). We conclude that $(r - t_s)\gamma^4\Delta_s^2 \le \Omega^2 L^2[g]$. Thus, the number T_s of steps of phase s admits the bound

$$T_s \le \frac{\Omega^2 L^2[g]}{\gamma^4\Delta_s^2}. \tag{65}$$

4^0. Assume that MDL does not terminate in course of first $T \ge 1$ steps, and let $\bar{s}$ be the index of the phase to which the step T belongs. Then $\Delta_{\bar{s}} > \epsilon$ (otherwise we would terminate not later than at the first step of phase $\bar{s}$); and besides this, by construction, $\Delta_{s+1} \le \gamma\Delta_s$ whenever phase $s+1$ takes place. Therefore

$$T \le \sum_{s=1}^{\bar{s}} T_s \le \frac{\Omega^2 L^2[g]}{\gamma^4}\sum_{r=0}^{\bar{s}-1}\Delta_{\bar{s}-r}^{-2} \le \frac{\Omega^2 L^2[g]}{\gamma^4\Delta_{\bar{s}}^2}\sum_{r=0}^{\bar{s}-1}\gamma^{2r} \le \frac{\Omega^2 L^2[g]}{\gamma^4(1-\gamma^2)\Delta_{\bar{s}}^2} \le \frac{\Omega^2 L^2[g]}{\gamma^4(1-\gamma^2)\epsilon^2}. \qquad \square$$

A.3 Proof of Proposition 3.3

Observe that the algorithm can terminate only in the case A of B.2.2.3, and in this case the output is indeed as claimed in Proposition. Thus, all we need to prove is the upper bound (37) on the number of steps before termination.

1^0. Let us bound from above the number of steps at an arbitrary phase s. Assume that phase s did not terminate in course of the first T steps, so that $u_1, ..., u_T$ are well defined. We claim that then

$$\|u_\tau - u_{\tau+1}\| \ge \ell_s/L[g],\ 1 \le \tau < T. \tag{66}$$

Indeed, by construction $h^s_{\tau-1,m+1}(y) := \langle g(u_\tau), u_\tau - y\rangle$ is $\ge \ell_s = \gamma f_s$ when $x = u_{\tau+1}$ (due to $u_{\tau+1} \in Y_\tau$). Since $\|g(u)\|_* \le L[g]$ for all $u \in Y$, (66) follows.

Now let us look at what happens with the quantities $\omega(u_\tau)$ as τ grows. By strong convexity of ω we have

$$\omega(u_{\tau+1}) - \omega(u_\tau) \ge \langle\omega'(u_\tau), u_{\tau+1} - u_\tau\rangle + \frac{1}{2}\|u_\tau - u_{\tau+1}\|^2$$

The first term in the right hand side is ≥ 0, since u_τ is the minimizer of $\omega(\cdot)$ over $Y^s_{\tau-1}$, while $u_{\tau+1} \in Y_\tau \subset Y^s_{\tau-1}$. The second term in the right hand side is $\ge \frac{L^2[g]}{2\ell_s^2}$ by (66). Since $\omega(u_{\tau+1}) - \omega(u_\tau) \ge \frac{L^2[g]}{2\ell_s^2}$, we get $\omega(u_T) - \omega(u_1) \ge (T-1)\frac{\ell_s^2}{2L^2[g]} = (T-1)\frac{\gamma^2 f_s^2}{2L^2[g]}$. Recalling the definition of Ω, the left hand side in this inequality is $\le \frac{1}{2}\Omega^2$. It follows that whenever phase

s does not terminate in course of the first T steps, one has $T \le \frac{\Omega^2 L^2[g]}{\gamma^2 f_s^2} + 1$, that is, the total number of steps at phase s, provided this phase exists, is at most $T_s = \frac{\Omega^2 L^2[g]}{\gamma^2 f_s^2} + 2$. Now, we have

$$f_s \le f_1 = \max_{y \in Y} \langle g(y_\omega), y - y_\omega \rangle \le L[g] \max_{x \in Y} \|y - y_\omega\| \le \Omega L[g]$$

(recall that $\|g(y)\|_* \le L[g]$ and see (12)). Thus

$$T_s = \frac{\Omega^2 L^2[g]}{\gamma^2 f_s^2} + 2 \le \frac{(1+2\gamma^2)}{\gamma^2} \frac{\Omega^2 L^2[g]}{f_s^2}$$

for all s such that s-th phase exists. By construction, we have $f_s \ge \epsilon$ and $f_s \le (\gamma + (1-\gamma)\theta) f_{s-1}$, whence the method eventually terminates (since $\gamma + (1-\gamma)\theta < 1$). Assuming that the termination happens at phase $\bar{s}$, the total number of steps is bounded by

$$\begin{aligned} \sum_{s=1}^{\bar{s}} \frac{(1+2\gamma^2)}{\gamma^2} \frac{\Omega^2 L^2[g]}{f_s^2} &\le \sum_{s=1}^{\bar{s}} \frac{(1+2\gamma^2)}{\gamma^2} \frac{\Omega^2 L^2[g](\gamma + (1-\gamma)\theta)^{2(\bar{s}-s)}}{f_{\bar{s}}^2} \\ &\le \sum_{s=1}^{\bar{s}} \frac{(1+2\gamma^2)}{\gamma^2} \frac{\Omega^2 L^2[g](\gamma + (1-\gamma)\theta)^{2(\bar{s}-s)}}{\epsilon^2} \le \frac{(1+2\gamma^2)}{\gamma^2[1 - (\gamma + (1-\gamma)\theta)^2]} \frac{\Omega^2 L^2[g]}{\epsilon^2}, \end{aligned}$$

as claimed. □

A.4 Verifying (41)

For a fixed $\beta > 0$, let $y(x) \in \operatorname{argmin}_{y \in Y} [\langle x, Ay + a \rangle + \psi(y) + \beta\omega(y)]$, so that $(f_*^\beta)'(x) = Ay(x) + a$. Let $x, x' \in X$. Taking into account ψ is Lipschitz continuous, by argument completely similar to the one in the proof of Lemma A.1, we have $y(x) \in Y^o$, $y(x') \in Y^o$ and

$$\begin{aligned} \langle A^* x, y(x') - y(x) \rangle + \langle \psi'(y(x)), y(x') - y(x) \rangle + \beta \langle \omega'(y(x)), y(x') - y(x) \rangle &\ge 0 \\ \langle A^* x', y(x) - y(x') \rangle + \langle \psi'(y(x')), y(x) - y(x') \rangle + \beta \langle \omega'(y(x')), y(x) - y(x') \rangle &\ge 0 \end{aligned}$$

with properly selected $\psi'(y(x)) \in \partial\psi(y(x))$, $\psi'(y(x')) \in \partial\psi(y(x'))$. It follows that

$$\begin{aligned} \langle x - x', A(y(x') - y(x)) \rangle &\ge \langle \psi'(y(x')) - \psi'(y(x)), y(x') - y(x) \rangle \\ &\quad + \beta \langle \omega'(y(x')) - \omega'(y(x)), y(x') - y(x) \rangle \\ &\ge \beta \|y(x') - y(x)\|_y^2, \end{aligned}$$

and therefore

$$\begin{aligned} \|x - x'\|_x \|A\|_{y;x,*} \|y(x) - y(x')\|_y &\ge \|x - x'\|_x \|Ay(x) - Ay(x')\|_{x,*} \ge \langle x - x', A(y(x') - y(x)) \rangle \\ &\ge \beta \|y(x') - y(x)\|_y^2. \end{aligned}$$

The latter implies that

$$\|y(x) - y(x')\|_y \le \beta^{-1} \|A\|_{y;x,*} \|x - x'\|_x$$

so that

$$\|(f_*^\beta)'(x) - (f_*^\beta)'(x')\|_{x,*} = \|A(y(x) - y(x'))\|_{x,*} \le \beta^{-1} \|A\|_{y;x,*}^2 \|x - x'\|_x,$$

which is (41). □