A smoothing approach for composite conditional gradient with nonsmooth loss

Federico Pierucci, Zaid Harchaoui, Jérôme Malick

A smoothing approach for composite conditional gradient with nonsmooth loss

Federico Pierucci , Zaid Harchaoui , Jérôme Malick

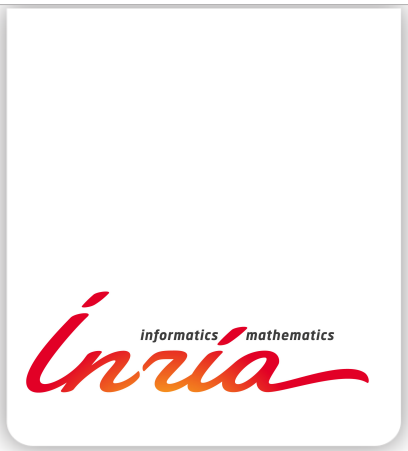
RESEARCH REPORT

N° 8662

June 2014

Project-Teams LEAR and BiPoP

ISSN 0249-6399 ISRN INRIA/RR--8662--FR+ENG



A smoothing approach for composite conditional gradient with nonsmooth loss

Federico Pierucci* †, Zaid Harchaoui *, Jérôme Malick ‡

Project-Teams LEAR and BiPoP

Research Report n° 8662 — June 2014 — 21 pages

Abstract: We consider learning problems where the non-smoothness lies both in the convex empirical risk and in the regularization penalty. Examples of such problems include learning with nonsmooth loss functions and atomic decomposition regularization penalty. Such doubly nonsmooth learning problems prevent the use of recently proposed composite conditional gradient algorithms for training, which are particularly attractive for large-scale applications. Indeed, they rely on the assumption that the empirical risk part of the objective is smooth.
We propose a composite conditional gradient algorithm with smoothing to tackle such learning problems. We set up a framework allowing to systematically design parametrized smooth surrogates of nonsmooth loss functions. We then propose a smoothed composite conditional gradient algorithm, for which we prove theoretical guarantees on the accuracy. We present promising experimental results on collaborative filtering tasks.

Key-words: conditional gradient, Frank-Wolfe, smoothing, large-scale convex optimization

* Inria, LEAR team, Inria Grenoble Rhône-Alpes, Laboratoire Jean Kuntzmann, CNRS, Univ. Grenoble Alpes, France.
† Univ. Grenoble Alpes, Laboratoire Jean Kuntzmann, CNRS, Inria Grenoble Rhône-Alpes, France.
‡ CNRS, BIPOP team, Laboratoire Jean Kuntzmann, CNRS, Inria Grenoble Rhône-Alpes, Univ. Grenoble Alpes, France.

RESEARCH CENTRE GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Contents

1 Introduction

The conditional gradient algorithm, *a.k.a.* Frank-Wolfe from the authors of the original paper in 1956, performs smooth optimization over a compact convex set and only requires i) a first-order oracle and ii) a linear minimization oracle over that compact convex set. This historical algorithm and its recent extensions to different optimization formulations [JS10, HJN13, HK12, ZYS12, LJJSP13] are increasingly popular due to their relevance for large-scale applications. Applications include collaborative filtering on the Netflix dataset [JS10, SSGS11]. Related works also include greedy or forward selection algorithms [SSGS11], which can be considered as cousins to conditional gradient algorithms.

Indeed, conditional gradient algorithms stand in contrast to proximal algorithms for first-order optimization. For composite smooth optimization, proximal algorithms [BJMO12] require a first-order oracle that returns objective and gradient evaluations (for the smooth part), and a proximal operator oracle associated with the nonsmooth part of the objective. Such algorithms are particularly attractive when the proximal operator is cheap to compute, as *e.g.* for the vector ℓ_1-norm, and they enjoy an $O(1/t^2)$ convergence rate for their accelerated versions [JN10]. However, they could turn out to be prohibitive when the proximal operator is expensive if not impossible to compute, *e.g.* for the nuclear-norm of matrices when these matrices are high-dimensional, as it arises in the large-scale applications mentioned above. On the other hand, in place of the proximal operator oracle, conditional gradient algorithms (CGAs) require instead a linear minimization oracle (LMO), which is much cheaper to compute, *e.g.* for the nuclear-norm of matrices (maximal pair of singular vectors, in place of full SVD for the proximal operator).

Composite conditional gradient algorithms, that is first-order optimization algorithms for composite objectives that decompose into a smooth part and a nonsmooth part for which a LMO is available, have been proposed [DHM12, HJN13, ZYS12]. Composite objectives correspond to learning problems with smooth loss functions and nonsmooth *regularization penalty*. Convergence rates with rate $O(1/t)$ were recently proven for such algorithms [HJN13]. However, in a machine

learning context, these algorithms assume smooth loss functions, whereas for several applications nonsmooth loss functions would be preferable [WKLS07, AFSU07]. Smoothing strategies were recently proposed for nonsmooth counterparts of the "historical" conditional gradient algorithm, that is for nonsmooth objectives (instead of smooth in the original [Jag13]) with a compact convex constraint [Lan, GH13].

We propose here a smoothed version of the composite conditional gradient algorithm, using the smoothing technique from [Nes05]. We give a detailed study of smoothing of nonsmooth loss functions in a machine learning context, and give theoretical grounding for several popular smoothed counterpart of nonsmooth loss functions. We prove a theoretical guarantee on the accuracy of the solution given by our algorithm and present promising experimental results on collaborative filtering.

2 Smooth optimization with atomic-decomposition regularization

In this section, we recall the main properties of atomic-decomposition norms, and then describe composite conditional gradient algorithms [DHM12, HJN13, ZYS12], which are tailored for learning problems with these norms, as regularizers.

Learning with atomic-decomposition norms Consider a sequence of *i.i.d.* examples $u_1, \dots, u_N$, and a loss function $\ell(W, u)$. Denote $R_{\text{emp}}(W) = 1/N \sum_{i=1}^{N} \ell(W, u_i)$ the corresponding empirical risk. In this paper, we consider regularized learning problems that write as

$$\min_{W} \; g(W) := \lambda \|W\|_{\mathcal{A}} + R_{\text{emp}}(W) \tag{1}$$

where $\|\cdot\|_{\mathcal{A}}$ is a so-called atomic-decomposition norm [CRPW12, DHM12]. Atomic-decomposition norms (or atomic norm, in short) can be defined by the following simple variational description with respect to a compact set $\mathcal{A}$ (the "atoms"). Assume that the elements of $\mathcal{A}$ are the extreme points of $\text{conv}(\mathcal{A})$ (the convex hull of $\mathcal{A}$), we have

$$\|W\|_{\mathcal{A}} = \inf \left\{ \sum_{i \in I} \theta_i \; : \; \theta_i > 0, \, W = \sum_{i \in I} \theta_i a_i \right\}$$

where I is an index set spanning the elements of $\mathcal{A}$, and where $(a_i)_{i \in I} \in \mathcal{A}$. Such characterization leverages the property that norms belong to the larger family of "gauges", that are convex and positively homogeneous functions, centered in the origin. The support function of the collection of atoms $\mathcal{A}$ writes as

$$\|W\|_{\mathcal{A}}^{*} = \sup_{a \in \mathcal{A}} \langle W, a \rangle. \tag{2}$$

We can recognize that $\|\cdot\|_{\mathcal{A}}^{*}$ is the dual (or polar) norm associated with $\|\cdot\|_{\mathcal{A}}$.

Many useful atomic norms enjoy collections of atoms $\mathcal{A}$ that are simple to describe, and whose support functions are *computationally easy to compute*. Examples include the ℓ_1-norm in $\mathbb{R}^d$, where $\mathcal{A}$ is the canonical basis of $\mathbb{R}^d$, and the trace-norm (or nuclear-norm) in the space of rectangular matrices $\mathbb{R}^{d \times k}$, where $\mathcal{A} = \{uv^T, \, \|u\|_2 = \|v\|_2 = 1\}$. We refer to [Jag13] for a review of popular atomic norms.

Conditional gradient algorithms, which we shall describe in the next paragraph, take advantage of this attractive feature: they make progress using an (approximated) optimal solution of (2).

Composite conditional gradient for smooth risk Assume that the empirical risk $R_{\text{emp}}(\cdot)$ is a convex function with Lipschitz continuous gradient with Lipschitz constant L. Under suitable assumptions [HJN13], the composite conditional gradient algorithm with infinite memory enjoys the following theoretical guarantee

$$g(W_t) - \min_W g(W) \leq O\left(\frac{1}{t}\right).$$

The composite conditional gradient algorithm works by making calls to a *first-order oracle*, that returns $R_{\text{emp}}(W)$ and $\nabla R_{\text{emp}}(W)$ for any W, and to a *linear minimization oracle*, that is a subroutine that returns for any W

$$\mathbf{LMO}(W) := \operatorname*{argmin}_{a \in \mathcal{A}} \langle a, \nabla R_{\text{emp}}(W) \rangle . \tag{3}$$

This is in contrast to proximal algorithms, which make progress by making calls to a *proximal operator oracle*. Proximal operators are computationally expensive to compute in several large-scale learning problems. Typical examples are matrix completion with noise, or multi-class classification with nuclear-norm penalty, where the proximal operator associated with the nuclear-norm corresponds to a full singular value decomposition of the current iterate, which is prohibitive in large-scale applications. Moreover, recent results from [GN13] show that the conditional gradient algorithm, which runs in $O(1/t)$, as opposed to the accelerated proximal algorithms which run in $O(1/t^2)$, is almost optimal (up to a log factor) for large-scale optimization problems.

The composite conditional gradient algorithm is summarized below (see Algo. 1). An ϵ-solution is a W that satisfies

(i) $\|\nabla R_{\text{emp}}(W)\|_{\mathcal{A}} \leq \lambda + \epsilon$, and

(ii) $|\langle \nabla R_{\text{emp}}(W), W \rangle + \lambda \|W\|_{\mathcal{A}}| \leq \epsilon \|W\|_{\mathcal{A}}$.

Algorithm 1 Composite Conditional Gradient

Inputs: λ, ϵ
Initialize $W = \mathbf{0}$, $t = 1$
while W is not an ϵ-solution **do**
  Call the linear minimization oracle: $\mathbf{LMO}(W_t)$
  Compute
$$\min_{\theta_1, \ldots, \theta_t \geq 0} \quad \lambda \sum_{i=1}^{t} \theta_i + R_{\text{emp}}\Big(\sum_{i=1}^{t} \theta_i a_i \Big)$$
  Increment $t \leftarrow t + 1$
end while
Return $W = \sum_i \theta_i a_i$

3 Motivating example

We present here collaborative filtering as motivating example for designing a composite conditional gradient algorithm for matrix learning problems with nonsmooth loss functions. The (nonsmooth) regularization is the nuclear-norm, the sum of singular values of the matrix, which has an atomic-decomposition form $\|\cdot\|_{\mathcal{A}}$ with

$$\mathcal{A} = \{uv^T, \|u\|_2 = \|v\|_2 = 1\}.$$

Collaborative filtering, or matrix completion, consists in the generation of a low-rank matrix from few known approximate entries. The loss $\ell(w, x) = |w - x|$, based on ℓ_1 norm [Hub81] ensures robustness to outliers. We have (1)

$$\min_{W \in \mathbb{R}^{d \times k}} \quad \frac{1}{N} \sum_{(i,j) \in \Omega} |W_{ij} - X_{ij}| + \lambda \|W\|_{\mathcal{A}} \tag{4}$$

where Ω is the subset of $\{1, \dots, d\} \times \{1, \dots, k\}$ denoting pairs of observations (N is the size of Ω and $\{x_{ij}\}_{(i,j)\in\Omega}$ are the known entries).

4 Smoothing non-smooth loss functions

The smoothing technique that we consider in this paper was formalized, for the accelerated gradient method, by Nesterov [Nes05, Nes07]; see also [Ber04] for a review of earlier works. We study in greater detail this smoothing for the specific purpose of smoothing loss functions considered in learning problems. In this section, we give a short and comprehensive presentation of this smoothing technique applied to support functions, revealing a convex transformation generalizing the standard convex conjugation [HUL01].

Ball-conjugate Let $\gamma > 0$, a set $\mathcal{B} \subset \mathbb{R}^n$ and a function $f : \mathcal{B} \to \mathbb{R} \cup \{+\infty\}$. Note that, for the study of this section, the variable of function f is x.

We introduce the $\mathcal{B}$-conjugate of f of parameter γ to be the function defined for all s in $\mathbb{R}^n$ by

$$(\gamma f)^{\mathcal{B}}(s) := \max_{\substack{x \in \mathcal{B} \\ x \in \operatorname{dom} f}} \langle x, s \rangle - \gamma f(x) \,. \tag{5}$$

As a maximum of affine function, $(\gamma f)^{\mathcal{B}}$ is convex function. When $\gamma = 1$ and $\mathcal{B} = \mathbb{R}^n$, the $\mathcal{B}$-conjugate is nothing but the traditional convex conjugate: $f^{\mathcal{B}} = f^*$ (see Chap. E of [HUL01]). In general, we see on definitions that the $\mathcal{B}$-conjugate of f is the convex conjugate of γf restricted to $\mathcal{B}$,

$$(\gamma f)^{\mathcal{B}} = \left(\gamma f_{|_{\mathcal{B}}}\right)^* \tag{6}$$

where $f_{|_{\mathcal{B}}}$ is the restriction of f on $\mathcal{B}$.

The ball-conjugate inherits some (but not all) properties of the convex conjugate. Outstandingly, the Fenchel inequality still holds : for all $x \in \mathcal{B}$

$$s \in \partial f(x) \quad \Leftrightarrow \quad f^{\mathcal{B}}(s) + f(x) = \langle s, x \rangle. \tag{7}$$

Observe also that, by definition, the $\mathcal{B}$-conjugate of the constant zero function (denoted $\mathbf{0} : s \mapsto 0$) is the called support function of $\mathcal{B}$

$$\forall s \in \mathbb{R}^n \quad \sigma_{\mathcal{B}}(s) = \mathbf{0}^{\mathcal{B}}(s).$$

In the next section, we state the two properties of the ball-conjugate with respect to smooth approximations. Other properties, including many useful calculus rules, are presented in supplementary material.

Smooth approximation of support functions We show here that ball-conjugation gives an easy, constructive and controllable way to approximate, by smooth functions, the support function of $\mathcal{B}$

$$\sigma(s) = \max_{x \in \mathcal{B}} \langle s, x \rangle.$$

The next two theorems show the main results of the approximation. Let f be a convex function and $\mathcal{B}$ a convex compact set.

Theorem 4.1 (Approximation)**.** *Consider the lower and upper bounds on $\mathcal{B}$: $m \leq f(x) \leq M$ for all $x \in \mathcal{B}$. Then, for $s \in \mathbb{R}^n$*

$$\gamma m \leq \sigma(s) - (\gamma f)^{\mathcal{B}}(s) \leq \gamma M.$$

Theorem 4.2 (Smoothing)**.** *Assume f to be strongly convex with constant c on $\mathcal{B}$. Then $(\gamma f)^{\mathcal{B}}$ is differentiable on $\mathbb{R}^n$ and its gradient*

$$\nabla(\gamma f)^{\mathcal{B}}(s) = \operatorname*{argmax}_{x \in \mathcal{B}} \langle s, x \rangle - \gamma f(x)$$

is Lipschitz continuous on $\mathbb{R}^n$ with constant $L = 1/\gamma c$.

In words, any strongly convex function f generates, through its $\mathcal{B}$-conjugate, a smooth approximation of the support function of $\mathcal{B}$. In general, the choice of f would depend on the capability on computing easily its $\mathcal{B}$-conjugate and on the geometry of $\mathcal{B}$ to control the constants m and M. In the next section, we explicit computation with the squared euclidean norm. Other examples are given in appendix.

Ball-conjugate of ℓ_2^2-norm in $[-1, 1]$ Smooth functions generated by the squared euclidean norm have explicit form involving the projection operator onto $\mathcal{B}$

$$\pi_{\mathcal{B}}(y) = \operatorname*{argmin}_{x \in \mathcal{B}} \|x - y\|_2^2 \ .$$

Proposition 4.3. *Let $\gamma > 0$ and $\mathcal{B}$ convex compact set. The $\mathcal{B}$-conjugate of $f(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ can be expressed*

$$(\gamma f)^{\mathcal{B}}(s) = \left\langle \pi_{\mathcal{B}}\left(\tfrac{1}{\gamma}s\right), s \right\rangle - \frac{\gamma}{2}\left\|\pi_{\mathcal{B}}\left(\tfrac{1}{\gamma}s\right)\right\|_2^2.$$

Its gradient is

$$\nabla(\gamma f)^{\mathcal{B}}(s) = \pi_{\mathcal{B}}\left(\frac{s}{\gamma}\right) \ .$$

In addition $(\gamma f)^{\mathcal{B}}$ has Lipschitz constant $L = 1/\gamma$.

Smooth function based on the ℓ_2^2-norm would be interested only if the projection $\pi_{\mathcal{B}}$ is fast to compute. The next example provides an illustration that will be used in the next section. Consider in $\mathbb{R}$ the set $\mathcal{B} = [-1, 1]$, whose support function is just the absolute value $\sigma = |\cdot|$. By the previous proposition, the conjugate of the squared norm is for s in $\mathbb{R}$

$$(\gamma f)^{\mathcal{B}}(s) = s \cdot \pi_{[-1,1]}\left(\frac{s}{\gamma}\right) - \frac{\gamma}{2}\left(\pi_{[-1,1]}\left(\frac{s}{\gamma}\right)\right)^2. \tag{8}$$

In this simple example, the projection is explicit and we have

$$(\gamma f)^{\mathcal{B}}(s) = \begin{cases} \frac{1}{2\gamma}s^2 & \text{if } |s| \leq \gamma \\ |s| - \frac{\gamma}{2} & \text{if } |s| > \gamma \end{cases} \tag{9}$$

We observe that we have the global approximation $0 \leq |s| - (\gamma f)^{\mathcal{B}}(s) \leq \frac{\gamma}{2}$, as shown in Thm. 4.1. and that the function is smooth, as predicted by Thm 4.2. Note finally that

$$\nabla(\gamma f)^{\mathcal{B}}(s) = \begin{cases} 1 & \text{if} \quad s > \gamma \\ \frac{1}{\gamma} s & \text{if} \ |s| \leq \gamma \\ -1 & \text{if} \quad s < -\gamma \quad . \end{cases} \tag{10}$$

Application to the motivating example We show how the smoothing technique can be applied to the nonsmooth empirical loss of collaborative filtering with noise. We approximate the absolute value in the empirical risk of problem (4) by the smooth conjugate of (9) that we denote

$$\ell^{\gamma}(W_{ij}, X_{ij}) = (\gamma f)^{\mathcal{B}}(W_{ij} - X_{ij}).$$

For given smoothing parameter γ, we thus consider the smooth surrogate learning problem

$$\min_{W} \frac{1}{N} \sum_{(i,j)\in\Omega} \ell^{\gamma}(W_{ij}, X_{ij}) + \lambda \left\|W\right\|_{\mathcal{A}}. \tag{11}$$

The empirical risk of this problem is now smooth and we have a explicit expression of its gradient by Eqn. (10). For any $(i, j) \in \Omega$

$$(\nabla R^{\gamma}_{emp}(W))_{ij} = \frac{1}{N} \nabla_{W_{ij}} \ell^{\gamma}(W_{ij}, X_{ij}). \tag{12}$$

5 Smoothed Composite Conditional Gradient

We present here the proposed algorithm, termed *Smoothed Composite Conditional Gradient* and abbreviated SCCG in the remainder of the paper.

Smoothing the empirical risk In the motivating example, the empirical risk $R_{\text{emp}}(W)$ (abbreviated $R(W)$ from now on) is an empirical average over all the examples of some *nonsmooth* loss function

$$R(W) = \frac{1}{N} \sum_{i=1}^{N} \ell(W, u_i)$$

where ℓ is a support function precomposed with an affine operator:

$$\ell(W, u_i) := \max_{x\in\mathcal{B}} \langle A(W, u_i), x\rangle.$$

We wrote the loss function in a compact form, to be understood as an abstract form that can be represent also (among others) our motivating example. Thanks to the smoothing technique, we can now design a *smoothed* version $R^{\gamma}(W)$ of the empirical risk, parameterized by a smoothing parameter γ that controls the amount of smoothing.

$$R^{\gamma}(W) = \frac{1}{N} \sum_{i=1}^{N} \ell^{\gamma}(W, u_i)$$

where

$$\ell^{\gamma}(W, u_i) := \max_{x\in\mathcal{B}} \langle A(W, u_i), x\rangle - \gamma f(x).$$

By Proposition 4.2, $R^\gamma(\cdot)$ is differentiable with Lipschitz continuous gradient. Therefore, one can use the composite conditional gradient algorithm presented earlier to solve the smooth optimization problem

$$\min_W \; g^\gamma(W) := \lambda\|W\|_{\mathcal{A}} + R^\gamma(W). \tag{13}$$

However, assuming that a solution to this problem is found by the composite conditional gradient algorithm

$$W^\gamma := \operatorname*{argmin}_W \; g^\gamma(W)$$

it is yet to be determined how this solution deviates from the solution of the original problem (1).

The next proposition gives an insight on this issue, with notation of the previous section. m, M and c are coming for the smoothing (see Theorems 4.1 and 4.2).

Proposition 5.1. *Assume that $\gamma \in [0, \gamma_{max}]$. In addition, assume that, for all $\gamma \in [0, \gamma_{max}]$, there exists D such that $\lambda r + R^\gamma(W) \leq R^\gamma(\boldsymbol{0})$ together with $\|W\|_{\mathcal{A}} \leq r$ imply that $r \leq D$. Then, setting*

$$\gamma(\epsilon) = \frac{\epsilon}{2(M-m)} \;,$$

we have that, after

$$T(\epsilon) \geq \left\lfloor \frac{16D^2}{\gamma c \epsilon} - 13 \right\rfloor$$

iterations, for a sufficiently small accuracy ϵ, the SCCG algorithm returns an ϵ-optimal minimum of g.

Proof. We decompose the difference into three parts

$$\begin{aligned} g(W_t) - \min_W g(W) \leq\; & g(W_t) - g^\gamma(W_t) \\ & + g^\gamma(W_t) - \min_W g^\gamma(W) \\ & + \min_W g^\gamma(W) - \min_W g(W) \end{aligned}$$

With the assumption, we can use the convergence rate of the composite conditional gradient with smooth loss (see Thm. 3 in [HJN13]): we get that

$$g^\gamma(W_t) - \min_W g^\gamma(W) \leq \frac{8L_\gamma D^2}{t+14}$$

where L_γ is the Lipschitz constant of the gradient. Here $L_\gamma = 1/\gamma c$ by Theorem 4.2. On the other hand, we have from Theorem 4.1

$$g(W_t) - g^\gamma(W_t) \leq \gamma M$$

and

$$\min_W g^\gamma(W) - \min_W g(W) \leq -\gamma m$$

Therefore, to get an ϵ-optimal minimum of g, it suffices to set $\gamma = \gamma(\epsilon)$ and to run $T(\epsilon)$ iterations of the SCCG algorithm. □

The above proposition can be interpreted as follows. Given a target accuracy ϵ, the optimal amount of smoothing $\gamma(\epsilon)$ can be computed so that after some number of iterations $T(\epsilon)$ an ϵ-optimal minimum of the objective function of interest g is reached.

The *smoothed composite conditional gradient algorithm* (SCCG) is summarized in Algo. 2. The SCCG algorithm works by making calls to a first-order oracle, that returns $R^{\gamma}_{\text{emp}}(W)$ and $\nabla R^{\gamma}_{\text{emp}}(W)$ for any W, and to a linear minimization oracle, that is a subroutine that returns for any W

$$\mathbf{LMO}^{\gamma}(W) := \operatorname*{argmin}_{a \in \mathcal{A}} \langle a, \nabla R^{\gamma}_{\text{emp}}(W) \rangle \; . \tag{14}$$

Algorithm 2 Smoothed Composite Conditional Gradient

Inputs: λ, γ, ϵ
Initialize $W = \mathbf{0}$, $t = 1$
for $t = 1, \ldots, T(\epsilon)$ **do**
  Call the linear minimization oracle: $\mathbf{LMO}^{\gamma}(W_t)$
  Compute

$$\min_{\theta_1, \ldots, \theta_t \geq 0} \quad \lambda \sum_{i=1}^{t} \theta_i + R^{\gamma}_{\text{emp}} \left(\sum_{i=1}^{t} \theta_i a_i \right)$$

end for
Return $W = \sum_i \theta_i a_i$

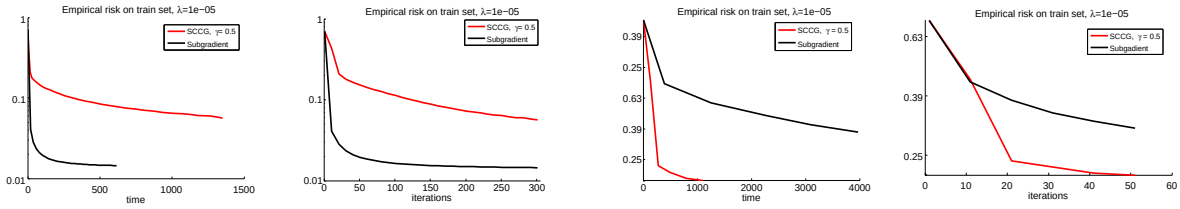


Figure 1: Comparison of empirical risk vs time (left of each box) and vs iterations number (right); $\gamma = 0.5\, \lambda = 10^{-5}$, on MovieLens datasets: the small dataset for the two figures on the left-hand side, the medium dataset for the two on the right-hand side.

Learning the smoothing parameter The optimal smoothing parameter $\gamma(\epsilon)$ depends on data-dependent quantities and requires some prior knowledge. Furthermore, the above result only gives insights on the optimal amount of smoothing in terms *optimization of the empirical risk*, whereas in real-world applications one is mainly interested *in fine* in the *risk on the test set*. In the experiments section, we see that an effective strategy would be to *learn the smoothing parameter* γ *from data* on a validation set. One would run the proposed algorithm SCCG(γ) for all the values of γ ranging on a discretized set, and measure the validation error for each value of γ on a held-out validation set. Then, one would pick the best γ in terms of error on the validation set. With that learned value $\gamma^{\star}$, one finally computes the test error obtained by the W returned by SCCG($\gamma^{\star}$) on the training set.

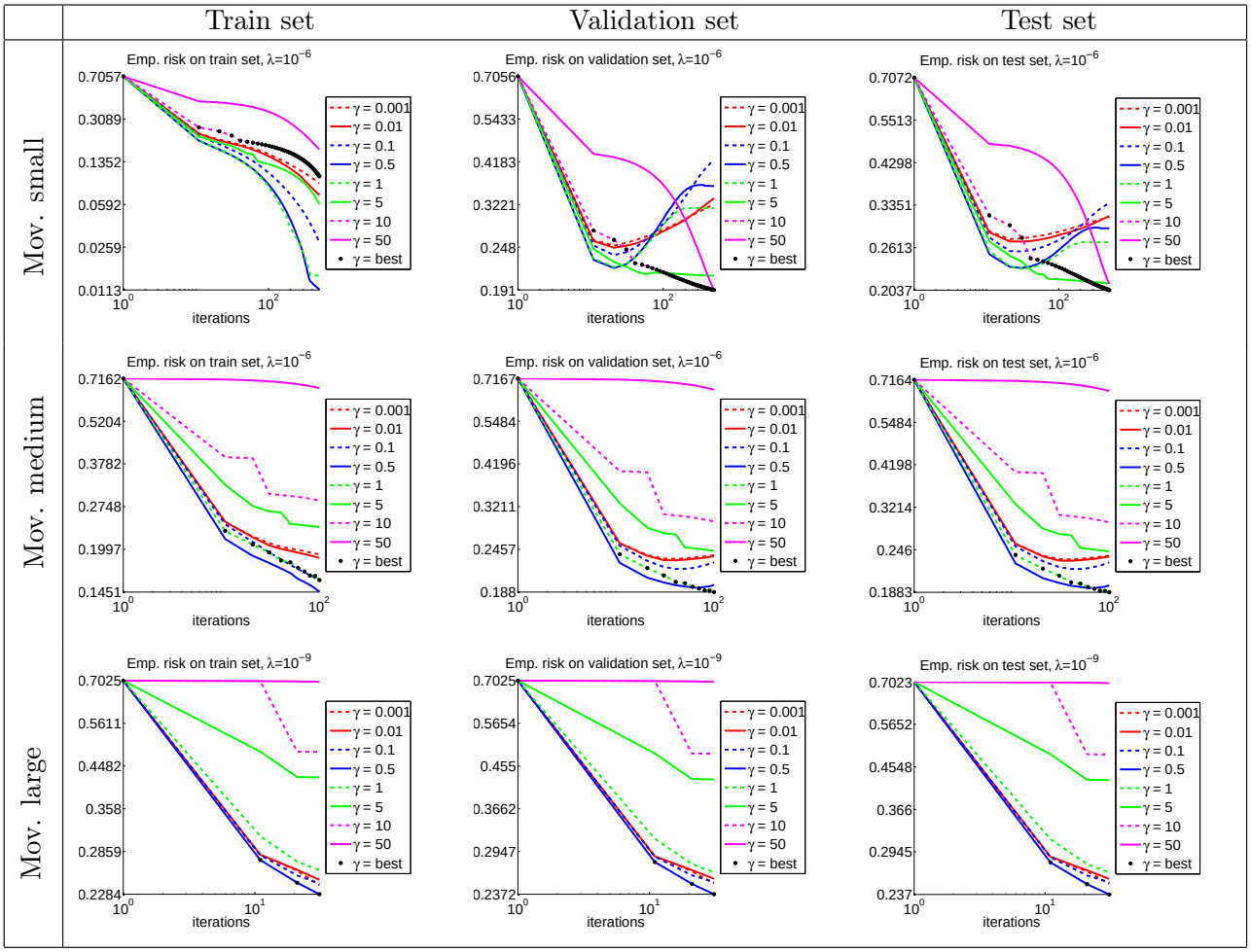


Figure 2: Movielens data - Empirical risk versus iterations.

6 Experiments

We now present the experimental results of the proposed composite conditional gradient algorithm for the learning problem of collaborative filtering with noise, on the MovieLens datasets, with nuclear norm penality and nonsmooth loss function. The experiences are launched on 3 disjoint sets for train, validation, test.

We chose $\gamma \in \{0.001; 0.01; 0.1; 0.5; 1; 5; 10; 50\}$; $\lambda \in \{10^{-2}; 10^{-4}; 10^{-6}; 10^{-8}; 10^{-10}; 10^{-12}\}$. For each λ the best γ_λ is chosen as the one that minimizes the empirical risk on validation set at last iteration. The pair $(\lambda_{\text{best}}, \gamma_{\text{best}})$ minimizes the empirical risk on validation set at last iteration. We chose as stop criterion a fixed number of iterations.

6.1 Implementation details

We implement our algorithm SCCG in Matlab. We use the quasi-Newton solver L-BFGS-B [BLNZ95] (via a Matlab interface) to perform, at iteration t of our algorithm, the minimization over the fixed set of t atoms.

To deal with large scale data, we pay attention to the memory to store the W_t generated by the algorithm. Each W_t is represented as a set containing the vectors u_t, v_t of each atoms a_t, of length N, and coefficients θ. An object of the form $\{(u_j, v_j, \theta_j)\}_{j=1\dots t}$ is stored at iteration t. With T the maximum number of iterations, the memory to store all the iterations is then proportional to $\frac{T(T+1)}{2}(2N+1)$.

Let us add a remark about the memory used for computations for movielens. Even though each W_t is a dense matrix of dimension $d \times k$, we are interested only in its observed entries for the optimization. So, W_t is never created as matrix object, but we keep only a representation of it with a vector of entries and a vector of indices of length n. So we use only n doubles instead of dk. The only time we need a matrix of size $d \times k$ is to compute the descent direction, but this matrix corresponds to the gradient of the loss and is sparse.

6.2 Competing approaches

A direct approach to solve out problem (1) would be to use standard nonsmooth optimization algorithms, namely bundle-like methods (see [HUL01]) or subgradient-like methods (see [Nes04], including proximal methods interpreted as implicit subgradient methods). Each iteration of these methods requires the knowledge of a subgradient of the entire objective function g (or at least an approximation of a subgradient). For many standard empirical losses, as the one used in this paper, a subgradient is readily available. There also exists an explicit expression of the subdifferential of the trace-norm: we get a subgradient of the trace-norm at W from an SVD decomposition of W, see [Lew99]. This is a bottleneck in scaling such approach to large dimension: for the large-scale learning problem we consider, even computing a single SVD (then a single iteration of a nonsmooth optimization algorithm) is out-of-reach in a reasonable amount of time.

To illustrate this fact on collaborative filtering problems, we compare the SCCG algorithm with fixed γ with a tailored basic nonsmooth optimization: truncated subgradient descent. An iteration of this algorithm iteration writes $W_{k+1} = W_k + t_k G_k$ with G_k approximates a subgradient in $\partial g(W_k)$. We compute only the 100 largest singular values to construct G_k to save computing time. The comparaison of the decrease of the nonsmooth empirical risk is plotted in Figure 1. We see that for the small dataset the decrease of the subgradient method is better with respect of iterations and time, but that the situation is reversed for the medium-scale problems (and become even non-comparable for large-scale problems, not shown here). This confirms the discussion above about the prohibitive cost of computing a (even a poorly approximate of a) subgradient.

Again, more efficient algorithm as bundle methods would suffer from the same drawback: even though the algorithms are performant, they use information given by an oracle which over-costly for the problems we consider.

6.3 Collaborative filtering

Dataset We test our approach on the MovieLens dataset for collaborative filtering, described in [MAL+03]. This dataset contains evaluations of movies made by customers, represented by the sparse matrix $X \in \mathbb{R}^{d \times k}$. As every customer evaluated only a small number the movies, X is sparse. Here completing X means predict how a customer would evaluate a movie which he hasn't seen. Entries of X are normalized dividing by the max entry of X. We split the dataset into a training set, validation and test sets with respectively 60%, 20%, 20% of the entries. The number of users and movies in the MovieLens datasets are resp. $(943;\ 11,682)$ for the small, $(3,952;\ 16,040)$ for the medium, and $(71,564;\ 165,133)$ for the large one.

Table 1: Data used for Collaborative Filtering. Sparsity is observations divided by total number of entries.

MovieLens	users	movies	observ.	spars.
Small	943	1 682	100 000	6.3%
Medium	3 952	6 040	1 000 209	4.2%
Large	71 564	65 133	10 000 054	0.21%

Results The plots with the nonsmooth empirical risk show better performance for medium γ values. When γ gets smaller we have a better approximation of the empirical risk, but the larger Lipschitz constant $L = 1/\gamma$ slows down the convergence of the algorithm. When γ gets larger the approximation of the empirical risk gets worst. We recall that we obtain iterates with SCCG optimizing the smooth surrogate, but we plot the values of nonsmooth loss for those iterates. In fig 3 and 2 we see the performance for all γ with the best choice of λ.

7 Conclusion

We proposed a composite conditional gradient algorithm that is suitable for regularized learning problems with nonsmooth loss functions, and showed promising experimental results. The framework we used allows to build smoothed counterparts of nonsmooth loss functions in a principled manner, with theoretical guarantees on the accuracy with respect to the orignal *doubly non-smooth* objective.

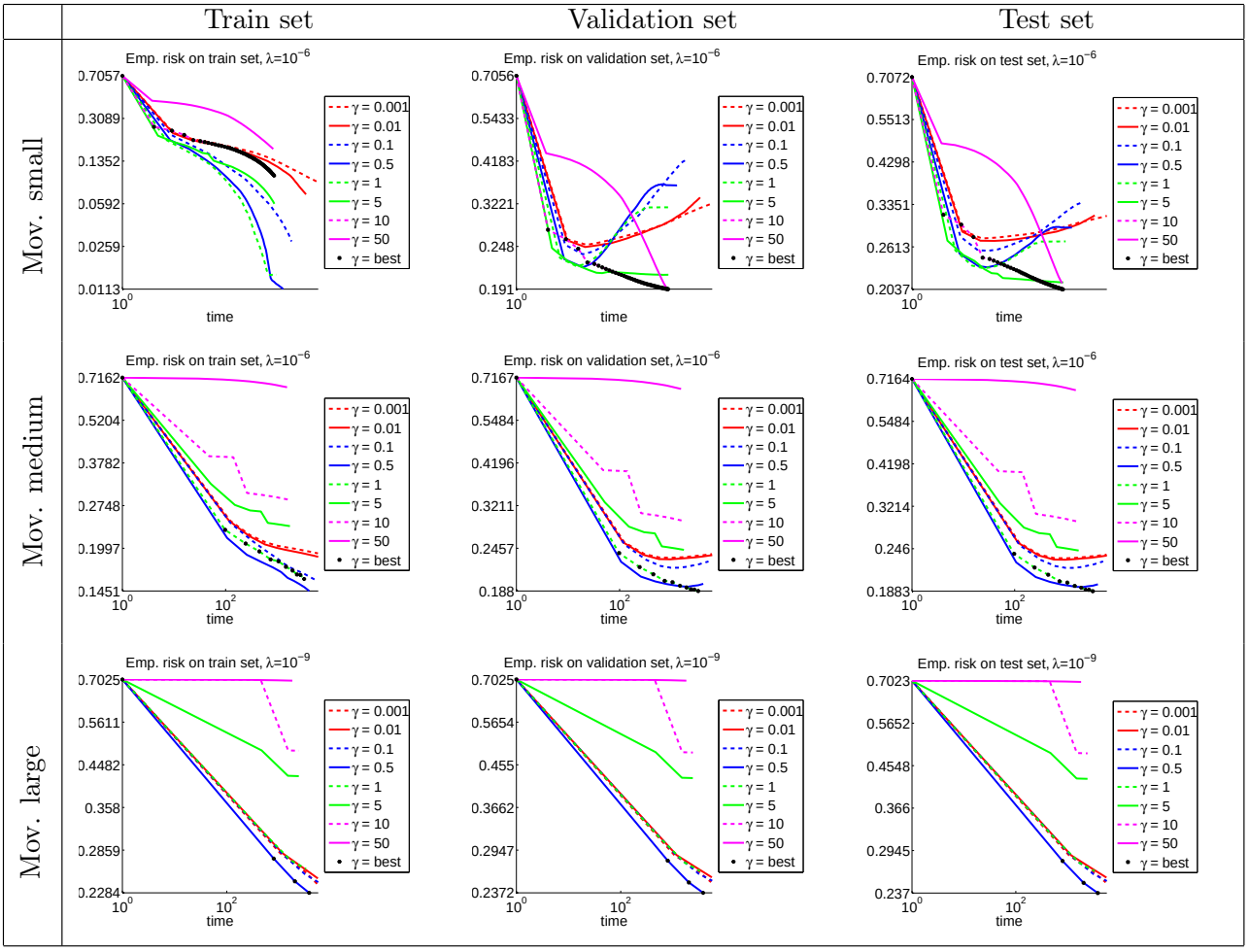


Figure 3: Movielens data - Empirical risk versus time. Related to all γ for the best choice of λ.

References

[AFSU07] Y. Amit, M. Fink, N. Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In *ICML*, 2007.

[Ber04] D. Bertsekas. *Nonlinear Programming (2nd ed.)*. Athena Scientific, 2004.

[BJMO12] F. R. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[BLNZ95] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[CRPW12] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *FOCM*, 12(6):805–849, 2012.

[DHM12] M. Dudik, Z. Harchaoui, and J. Malick. Lifted coordinate descent for learning with trace-norm regularization. *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

[GH13] D. Garber and E. Hazan. A Linearly Convergent Conditional Gradient Algorithm with Applications to Online and Stochastic Optimization. *ArXiv e-prints*, 1301.4666, 2013.

[GN13] C. Guzman and A. Nemirovski. On Lower Complexity Bounds for Large-Scale Smooth Convex Optimization. *ArXiv e-prints*, 1307.5001, 2013.

[HJN13] Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional Gradient Algorithms for Norm-Regularized Smooth Convex Optimization. *ArXiv e-prints*, 1302.2325, 2013.

[HK12] E. Hazan and S. Kale. Projection-free online learning. In *ICML*, 2012.

[Hub81] P. J. Huber. *Robust statistics*. Wiley Series in Probability and Mathematical Statistics. J. Wiley, 1981.

[HUL01] J.B. Hiriart-Urruty and C. Lemarechal. *Fundamentals of Convex Analysis*. 2001.

[Jag13] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, pages 427–435, 2013.

[JN10] A. Juditsky and A. Nemirovski. First order methods for nonsmooth convex large-scale optimization. *Optimization for Machine Learning,(Sra, Nowozin, Wright, Eds), MIT Press, 2012*, 2010.

[JS10] M. Jaggi and M. Sulovský. A Simple Algorithm for Nuclear Norm Regularized Problems. *ICML 2010: Proceedings of the 27th international conference on Machine learning*, 2010.

[Lan] G. Lan. The Complexity of Large-scale Convex Programming under a Linear Optimization Oracle. *ArXiv e-prints*, 1302.2325.

[Lew99] A.S. Lewis. Nonsmooth analysis of eigenvalues. *Mathematical Programming*, 84(1):1–24, 1999.

[LJJSP13] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *ICML 2013*, 2013.

[MAL+03] B. Miller, I. Albert, S. K. Lam, J. Konstan, and J. Riedl. Movielens unplugged: Experiences with a recommender system on four mobile devices. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, 2003.

[Nes04] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer, 2004.

[Nes05] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1), 2005.

[Nes07] Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Math. Program.*, 110(2):245–259, 2007.

[SSGS11] S. Shalev-Shwartz, A. Gonen, and O. Shamir. Large-Scale Convex Minimization with a Low-Rank Constraint. In *ICML*, 2011.

[WKLS07] M. Weimer, A. Karatzoglou, Q. V. Le, and A. J. Smola. Cofi rank - maximum margin matrix factorization for collaborative ranking. In *NIPS*, 2007.

[ZYS12] X. Zhang, Y. Yu, and D. Schuurmans. Accelerated training for matrix-norm regularization: A boosting approach. In *NIPS*, 2012.

Appendix

Federico Pierucci, Zaid Harchaoui, Jérôme Malick

June 15, 2014

A Properties of the B-conjugate

In this section, we give the proofs of the results about the $\mathcal{B}$-conjugate, stated in the paper. We also add a couple of useful lemmas. Our developments rely on basic convex analysis; for the reader convenience, the main results we need are recalled in Section B.

We start with proving that the approximation of the support function of $\mathcal{B}$ by the $\mathcal{B}$-conjugate comes directly from its construction.

Proof. (of Thm. 4.1) The bounds $m \leq f(\cdot) \leq M$ yield that, for all $x \in \mathcal{B}$ and $s \in \mathbb{R}^n$

$$\gamma m + \langle x, s\rangle - \gamma f(x) \leq \langle x, s\rangle \leq \gamma M + \langle x, s\rangle - \gamma f(x)$$

Taking the max over $x \in \mathcal{B}$ gives for all $s \in \mathbb{R}^n$

$$\gamma m + (\gamma f)^{\mathcal{B}}(s) \leq \sigma_{\mathcal{B}}(s) \leq \gamma M + (\gamma f)^{\mathcal{B}}(s)$$

which permits to conclude. □

It is important for our approach to have some calculus rules to construct functions $(\gamma f)^{\mathcal{B}}$. Suppose we know the explicit formula for $f^{\mathcal{B}}$, we derive expression for the $\mathcal{B}$-conjugate of γf and of the sum of f and an affine function.

Proposition A.1. *For any $\gamma > 0$, $b \in \mathbb{R}$ and $k \in \mathbb{R}^n$, we have:*

$$(\gamma f)^{\mathcal{B}}(s) = \gamma f^{\mathcal{B}}\left(\tfrac{1}{\gamma} s\right) \tag{15}$$

$$(f + \langle \cdot, k\rangle + b)^{\mathcal{B}}(s) = f^{\mathcal{B}}(s - k) - b \tag{16}$$

Proof. (of Prop. A.1) We just develop from the definitions:

$$\begin{aligned}(\gamma f)^{\mathcal{B}}(s) &= \max_{x \in \mathcal{B}} \langle x, s\rangle - \gamma f(x) \\ &= \gamma \max_{x \in \mathcal{B}} \tfrac{1}{\gamma}\langle x, s\rangle - f(x) \\ &= \gamma f^{\mathcal{B}}\left(\tfrac{s}{\gamma}\right).\end{aligned}$$

In a similar way:

$$\begin{aligned}(f + \langle \cdot, k\rangle + b)^{\mathcal{B}}(s) &= \max_{x \in \mathcal{B}} \langle x, s\rangle - \langle x, k\rangle - f(x) - b \\ &= \max_{x \in \mathcal{B}} \langle x, s - k\rangle - f(x) - b \\ &= f^{\mathcal{B}}(s - k) - b.\end{aligned}$$

□

We now turn to the smoothness of the $\mathcal{B}$-conjugate. In what follows, $\mathcal{B}$ is a convex compact set, and f is a closed convex function on $\mathcal{B}$. We also introduce

$$x^*(s) := \operatorname*{argmax}_{x \in \mathcal{B}} \langle s, x \rangle - \gamma f(x),$$

whose dependence on $\mathcal{B}$, f and γ is implicit when obvious.

Lemma A.2. *Let f be strictly convex on $\mathcal{B}$; then $f^{\mathcal{B}}$ is differentiable on $\mathbb{R}^n$ and*

$$\nabla f^{\mathcal{B}}(s) = x^*(s) \in \mathcal{B}.$$

Proof. (of Lemma A.2) By Theorem B.1 and the compactness of $\mathcal{B}$

$$\partial f^{\mathcal{B}}(s) = \left\{ x \;\middle|\; x \in \operatorname*{argmax}_{x \in \mathcal{B}} \langle x, s \rangle - f(x) \right\}.$$

We observe that the argmax is unique as f is strictly convex, then $\partial f^{\mathcal{B}}$ has only one element. By Lemma B.2 we conclude that $\nabla f^{\mathcal{B}}(s) = x^*(s)$. □

Proposition A.3. *Let f be strictly convex on $\mathcal{B}$; the following propositions are equivalent*

(i) $f^*(s) + f(x) = \langle x, s \rangle$

(ii) $s \in \partial f(x)$

(iii) $x = \nabla f^{\mathcal{B}}(s)$

Proof. (of Proposition A.3) From (6), the property comes from Theorem B.4 applied to $f_{|_{\mathcal{B}}}$: we just have to recall that $f^{\mathcal{B}}$ is differentiable and $\nabla f^{\mathcal{B}}(s) \in \mathcal{B}$ by Lemma A.2. □

We are now in position to prove that the strong convexity of f yields smoothness of its $\mathcal{B}$-conjugate.

Proof. (of Theorem 4.2) Let us prove the result for $\gamma = 1$; the result with general $\gamma > 0$ will follow by applying (15). From Lemma A.2, we know that $f^{\mathcal{B}}$ is differentiable and that its gradient is in $\mathcal{B}$. For any s_1, $s_2 \in \mathbb{R}^n$ we take

$$x_1 := \nabla f^{\mathcal{B}}(s_1), \quad x_2 := \nabla f^{\mathcal{B}}(s_2). \tag{17}$$

Then $s_1 \in \partial f(x_1)$, $s_1 \in \partial f(x_2)$, by Proposition A.3. Now recall from Theorem 6.1.2 of [HUL01] that strong convexity of f implies that

$$\langle s_1 - s_2, x_1 - x_2 \rangle \geq c \left\| x_1 - x_2 \right\|^2.$$

By substitution, we obtain for all $s_1, s_1 \in \mathbb{R}^n$

$$\langle s_1 - s_2, \nabla f^{\mathcal{B}}(s_1) - \nabla f^{\mathcal{B}}(s_2) \rangle \geq c \left\| \nabla f^{\mathcal{B}}(s_1) - \nabla f^{\mathcal{B}}(s_2) \right\|^2.$$

We apply Cauchy-Schwarz inequality and we simplify by $\left\| \nabla f^{\mathcal{B}}(s_1) - \nabla f^{\mathcal{B}}(s_2) \right\|$ to get

$$\left\| \nabla f^{\mathcal{B}}(s_1) - \nabla f^{\mathcal{B}}(s_2) \right\| \leq \frac{1}{c} \left\| s_1 - s_2 \right\|.$$

We conclude that the gradient of $f^{\mathcal{B}}$ is Lipschitzian on $\mathbb{R}^n$ with $L = \frac{1}{c}$. The result for $(\gamma f)^{\mathcal{B}}$ comes by using (15). □

Let us explicit the expressions and properties in the case of the squared norm.

Proof. (Proposition 4.3) Let us first prove the results with $\gamma = 1$. We have

$$\begin{aligned} x^*(s) &= \operatorname*{argmax}_{x\in\mathcal{B}} \langle x, s\rangle - f(x) \\ &= \operatorname*{argmin}_{x\in\mathcal{B}} \frac{1}{2}\|x\|^2 - \langle x, s\rangle \\ &= \operatorname*{argmin}_{x\in\mathcal{B}} \|x - s\|^2 - \|s\|^2 \\ &= \operatorname*{argmin}_{x\in\mathcal{B}} \|x - s\|^2 = \pi_{\mathcal{B}}(s), \end{aligned}$$

from which we get

$$\begin{aligned} f^{\mathcal{B}}(s) &= \langle x^*(s), s\rangle - \frac{1}{2}\|x^*(s)\|^2 \\ &= \langle \pi_{\mathcal{B}}(s), s\rangle - \frac{1}{2}\|\pi_{\mathcal{B}}(s)\|^2 . \end{aligned}$$

Obviously $f = \frac{1}{2}\|\cdot\|_2^2$ is strongly convex with modulus 1. By Lemma A.2, the gradient of $f^{\mathcal{B}}$ coincides with x^*, and by Theorem 4.2 the gradient $f^{\mathcal{B}}$ is Lipschitzian with Lipschitz constant is $L = 1$. We can also this property directly from the above expression of the gradient and the properties of the projection (which is 1-Lipschitz). The result for $(\gamma f)^{\mathcal{B}}$ comes by combining this with (15), as follows

$$\begin{aligned} (\gamma f)^{\mathcal{B}}(s) &= \gamma f^{\mathcal{B}}\left(\tfrac{1}{\gamma}s\right) \\ &= \gamma\left(\langle \pi_{\mathcal{B}}\left(\tfrac{1}{\gamma}s\right), \frac{s}{\gamma}\rangle - \tfrac{1}{2}\left\|\pi_{\mathcal{B}}\left(\tfrac{1}{\gamma}s\right)\right\|_2^2\right) \\ &= \langle \pi_{\mathcal{B}}\left(\tfrac{1}{\gamma}s\right), s\rangle - \tfrac{\gamma}{2}\left\|\pi_{\mathcal{B}}\left(\tfrac{1}{\gamma}s\right)\right\|_2^2, \end{aligned}$$

and

$$\nabla(\gamma f)^{\mathcal{B}}(s) = \nabla f^{\mathcal{B}}\left(\frac{s}{\gamma}\right) = \pi_{\mathcal{B}}\left(\frac{s}{\gamma}\right).$$

of Lipschitz constant is $L = \frac{1}{\gamma}$. □

Strong convexity of f on $\mathcal{B}$ is the key property to get smoothness of the $\mathcal{B}$-conjugate of f. We state and illustrate a simple lemma which permits to get easily strong convexity of f on $\mathcal{B}$.

Proposition A.4. *Let f be continuous on $\mathcal{B} \cap \operatorname{dom} f$ and twice differentiable. If there exists $c > 0$ such that for all $x \in \operatorname{int}(\mathcal{B}) \cap \operatorname{dom} f$, we have that the smallest eigenvalue of the Hessian $\nabla^2 f(x)$ is greater c. Then f is strongly convex over $\mathcal{B} \cap \operatorname{dom} f$ with modulus c.*

Proof. (of Proposition A.4) For any $x, y \in \operatorname{boundary}(\mathcal{B})$ we take two sequences $\{x_n\}, \{y_n\}$ lying in $\operatorname{int}(\mathcal{B})$ and converging to x, y. By Lemma B.3, we have that f is strongly convex on the interior of $\mathcal{B}$. By definition, this means that $f(\lambda x_n + (1-\lambda)y_n) \leq \lambda f(x_n) + (1-\lambda)f(y_n) - \frac{1}{2}c\lambda(1-\lambda)\|x_n - y_n\|^2$. Since f is continuous, we can pass to the limit in the above inequality to get the inequality for x and y too with the same modulus of strong convexity c. □

We finish with an notable remark on $(\cdot)^{\mathcal{B}}$ viewed on an operator on functions.

Proposition A.5. *The $\mathcal{B}$-conjugation $(\cdot)^{\mathcal{B}}$ is a convex operator, i.e.*

$$(\alpha f + (1-\alpha)g)^{\mathcal{B}} \leq \alpha f^{\mathcal{B}} + (1-\alpha)g^{\mathcal{B}}$$

and it is an anti-monotone operator respect to the relation of partial order '$>$', i.e.

$$f > g \text{ on } \mathcal{B} \quad \Longrightarrow \quad f^{\mathcal{B}} < g^{\mathcal{B}} \text{on } \mathbb{R}^n.$$

Proof. (of Proposition A.5) The results come easily from the definitions. We have:

$$\begin{aligned}
&(\alpha f + (1-\alpha)g)^{\mathcal{B}}(s) \\
&= \max_{x\in\mathcal{B}} \langle x, s\rangle - \alpha f(x) - (1-\alpha)g(x) \\
&= \max_{x,y\in\mathcal{B}, x=y} \alpha(\langle x, s\rangle - f(x)) + (1-\alpha)(\langle y, s\rangle - g(y)) \\
&\leq \max_{x,y\in\mathcal{B}} \alpha(\langle x, s\rangle - f(x)) + (1-\alpha)(\langle y, s\rangle - g(y)) \\
&= \alpha \max_{x,y\in\mathcal{B}} (\langle x, s\rangle - f(x)) + (1-\alpha) \max_{x,y\in\mathcal{B}} (\langle y, s\rangle - g(y)) \\
&= \alpha f^{\mathcal{B}}(s) + (1-\alpha)g^{\mathcal{B}}(s).
\end{aligned}$$

Moreover,

$$\begin{aligned}
f(x) > g(x) \Rightarrow \quad & -\langle s, x\rangle + f(x) > -\langle s, x\rangle + g(x) \\
\Rightarrow \quad & \langle s, x\rangle - f(x) < \langle s, x\rangle - g(x) \\
\Rightarrow \quad & \max_{x\in\mathcal{B}} \langle s, x\rangle - f(x) < \max_{x\in\mathcal{B}} \langle s, x\rangle - g(x) \\
\Rightarrow \quad & f^{\mathcal{B}}(s) < g^{\mathcal{B}}(s).
\end{aligned}$$

□

B Cited theorems

Our developments rely heavily on basic convex analysis properties. For sake of completeness, we recall some of them, as extracted from the textbook [HUL01].

Theorem B.1 (Thm. D 4.4.2)**.** *Let I be compact set, $f(x) = \sup\{f_i(x) \,|\, i \in I\}$. i is the active set of f. Assume the functions $i \to f_i(x)$ are upper semicontinuous. Then*

$$\partial f(x) = \operatorname{co}\{\cup \partial f_i(x) \,|\, i \in I(x)\}.$$

Lemma B.2 (Thm. D 2.1.4)**.** *Let F be convex. If $\partial F(x)$ contains only one element p (i.e. $\partial F(x) = \{p\}$), then F is (Fréchet) differentiable at x and $\nabla F(x) = p$.*

Lemma B.3 (Thm. B 4.3.1)**.** *Let F be twice differentiable over a convex set $\Omega \subset \mathbb{R}^n$. Then*

(i) F is strongly convex with modulus c on Ω if and only if the smallest eigenvalue of $\nabla^2 F$ is minorized by c on Ω, i.e.

$$\forall x \in \Omega \, \forall d \in \mathbb{R}^n \quad \langle \nabla^2 f(x) d, d\rangle \geq c \|d\|^2.$$

(ii) if $\nabla^2 f(x)$ is positive definite for all $x \in \Omega$, then f is strictly convex on Ω.

Theorem B.4. *The convex conjugate of a function f is defined by*

$$f^*(s) := \max_{x \in \operatorname{dom} f} \langle x, s \rangle - \gamma f(x), \tag{18}$$

If f is a closed convex function, the following propositions are equivalent

(i) $f^*(s) + f(x) = \langle x, s \rangle$

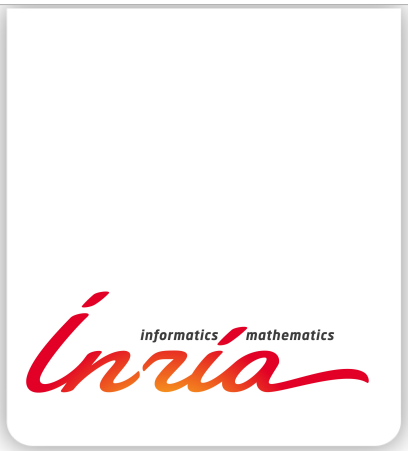
(ii) $s \in \partial f(x)$

(iii) $x \in \partial f^*(s)$

Lemma B.5. *Let F be differentiable and S affine, then*

$$\nabla(F \circ S)(W) = S^*(\nabla F(S(W))$$

where S^* is the adjoint of S.



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399