IASD/MASH project
Convergence rates for the Frank-Wolfe algorithm

Vincent Duval
vincent.duval@inria.fr

October 20, 2023

This project aims at solving the constrained optimization problem:

$$\min_{x \in C} f(x) \tag{1}$$

where $C \subseteq \mathbb{R}^d$ is a nonempty compact convex set, and $f : \mathbb{R}^d \to \mathbb{R}$ is a C^1 function, using the Frank-Wolfe algorithm.

Our goal is to investigate the convergence rates provided in [WKP23], that is, we want to find k such that $h_t = O\left(\frac{1}{t^k}\right)$ (or even better $h_t = O(e^{-t})$), where

$$h_t \stackrel{\text{def.}}{=} f(x_t) - f(x^*),$$

$(x_t)_{t \in \mathbb{N}}$ are the iterates, and x^* is a solution to (1).

To complete this project, it is not necessary to understand all the proofs nor all the results of [WKP23]. But getting a quick glimpse at the paper[1] might help. We will only focus on a few results and definitions which are referenced below.

Note: *For each experiment it is requested to draw two kinds of plots, corresponding to different dimensions:*

- *A plot of the energy decay (that is, $f(x_t)$ or $\min_{0 \le i \le t} f(x_t)$ versus t), with the scale of your choice. For this you may consider medium-size problems (e.g. $d = 100$)*
- *A plot of the optimization path (that is, the successive iterates x_t inside the convex set C) so as to understand the effect of the curvature of the set. If possible, please superimpose the shape of the constraint set and the level lines of f. For this, you may use $d = 2$ (or $d = 3$).*

[1] The article may be downloaded at https://arxiv.org/abs/2205.12838

Generalities. Throughout the project, we will consider the following convex sets:

- $C_2 = \left\{ x \in \mathbb{R}^d \mid \sum_{i=1}^d |x_i|^2 \leq 1 \right\}$, the ℓ^2 unit ball,
- $C_\infty = \left\{ x \in \mathbb{R}^d \mid \max_{1 \leq i \leq d} |x_i| \leq 1 \right\}$, the ℓ^∞-unit ball,
- $C_\mathcal{P} = \left\{ x \in \mathbb{R}^d \mid \sum_{j=1}^d x_j = 1, \text{ and } \forall i, \ x_i \geq 0 \right\}$ the probability simplex.

1. Given a vector $a \in \mathbb{R}^d$, we want to find a solution to the linear minimization problem

$$\min_{p \in C} \langle a, p \rangle, \tag{2}$$

where C is C_2, C_∞, or $C_\mathcal{P}$.

(a) What are the extreme points of those convex sets? Which convex set is a polytope?

(b) Check that some solutions are respectively given by

$$p_2 \overset{\text{def.}}{=} -\frac{a}{\|a\|_2} \tag{3}$$

$$p_\infty \overset{\text{def.}}{=} -\operatorname{sign}(a), \tag{4}$$

where $(\operatorname{sign}(a))_i = 1$ if $a_i > 0$, -1 if $a_i < 0$ and $a_i \in [-1, 1]$ if $a_i = 0$,

$$p_\mathcal{P} = e_i, \quad \text{where } i = \operatorname{argmin}_j a_j \tag{5}$$

and $e_j = (0, \ldots, 0, 1, 0, \ldots)$ is the jth canonical vector.

(c) Implement the linear minimization oracles, that is, functions `lmo2`, `lmoInf`, `lmoProb`, which take as input some vector $a \in \mathbb{R}^d$ and return a solution to (2) in these cases.

2. Implement the Frank-Wolfe algorithm with open-loop step-size (see Algorithm 1 in [WKP23]). The function(s) should rely on the linear minimization oracles defined above, and on a stepsize rule given by

$$\forall t \in \mathbb{N}, \ \eta_t = \frac{\ell}{t + \ell} \tag{6}$$

where $\ell \in \mathbb{N}$ is a parameter given to the function.

Note: *In this study, in order to avoid too specific behaviors, please do not initialize the algorithm with $x_0 = 0$, but pick a random point in each convex set. In order to make reproducible experiments, you may create a random number generator with a specific seed. For instance:*

```
import numpy as np
rng = np.random.default_rng(12345) #sets the seed
d= 100 # ambient dimension
myrandomvec = rng.random(d) # random vector using the seed
xzero = ...#do some stuff to build xzero from myrandomvec
          #so that xzero belongs to the convex set C
```

3. Assume that f is m-strongly convex, with (unique) minimizer $x^* \in \mathbb{R}^d$. For which values of μ and θ does f satisfy the Hölder Error bound (Def. 2.3),

$$\forall x \in \mathbb{R}^d, \quad \mu \left(f(x) - f(x^*)\right)^\theta \geq \|x - x^*\|_2 \tag{7}$$

?

Probability simplex. We set $C = C_\mathcal{P}$ and we consider the function f given by

$$f(x) = \frac{1}{2} \|x\|_2^2 . \tag{8}$$

4. Check that the point $x^* = \frac{1}{d}(1, 1, \ldots, 1)$ is the unique minimizer of (1) *(you may use the KKT conditions but a clever use of the Cauchy-Schwartz inequality is quicker).* Is it in the relative interior of $C_\mathcal{P}$?
5. What order of convergence rate does Theorem 3.6 predict?
6. Run the algorithm and compare with the theory.

Now, we define the vector $\bar{\mathbb{1}} = (0, \ldots, 0, 1, \ldots, 1)$ with $d/2$ zeros and $d/2$ ones (we assume d is even). We admit that the unique minimizer to (1) with

$$f(x) = \frac{1}{2} \|x - \rho\bar{\mathbb{1}}\|_2^2 \tag{9}$$

is given by $x^* = \frac{2}{d}\bar{\mathbb{1}}$.

7. Is it possible to apply Theorem 3.6? and Theorem 3.10? Theorem 3.12?
8. What does Theorem 4.5 tell us about the order of convergence? Run the algorithm and compare the practical performance and the theoretical prediction. You may also try different values of ℓ.

ℓ^∞ unit ball.

9. Try to observe similar behaviors with $C = C_\infty$ as with the probability simplex, by carefully choosing some (e.g. quadratic) functions f.

ℓ^2 unit ball. Now, we take $C = C_2$, and we admit without proof that C is strongly convex (or $(\alpha, 2)$ uniformly convex, in the sense of Definition 2.1).

We consider the function f defined by

$$f(x) = \frac{1}{2} \|Ax - b\|_2^2, \tag{10}$$

where $b = Ax^\dagger$ with $x^\dagger \in \mathrm{Im} A^\top$ and $\|x^\dagger\|_2 = 2$. In the experiments, you may typically draw $A \in \mathbb{R}^{N\times d}$ at random with $N < d$, or take A from your favorite dataset.

10. Using Pythagora's theorem, prove that for every $x \in \mathrm{argmin}_{\mathbb{R}^d} f$,

$$\|x\|_2 \geq 2.$$

Deduce that there exists some some constant $\lambda > 0$ such that

$$\forall x \in C_2, \quad \|\nabla f(x)\|_2 \geq \lambda > 0. \tag{11}$$

11. What does Theorem 3.10 tell us about the order of convergence (see also Remark 3.11)?

12. Try this numerically, and compare the practical performance with the theory. You may try the step-sizes corresponding to $\ell \in \{1, 2, 4, 6\}$.

Comparison with exact linesearch. Other update rules are possible in Algorithm 1. Instead of the open-loop step-size $\eta_t = \frac{\ell}{t+\ell}$, one might use a linesearch:

$$\eta_t = \mathrm{argmin}_{\eta\in[0,1]} f\left((1-\eta)x_t + \eta p_t\right) \tag{12}$$

If the objective is quadratic, e.g. $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, for $A \in \mathbb{R}^{N\times d}$, it is possible to have closed-form expressions for the linesearch by using the following observation:

Lemma 1 *Let $P(\eta) = a\eta^2 + b\eta + c$, with $a \neq 0$, be a polynomial of degree 2, such that $P'(0) \leq 0$. Then*

$$\mathrm{argmin}_{\eta\in[0,1]} P(\eta) = \begin{cases} -\frac{b}{2a} & \text{if } P'(1) > 0, \\ 1 & \text{if } P'(1) \leq 0. \end{cases} \tag{13}$$

13. Implement the Frank-Wolfe algorithm with exact line-search, and compare its performance with the open-loop step-sizes $\eta_t = \frac{\ell}{t+\ell}$, $\ell \in \mathbb{N}^*$.

    Explore

    - different values of ℓ,
    - different quadratic functions,
    - the different convex sets discussed above.

    You may discuss the result in light of Table 1 (do you observe the same rates?).

Acceleration strategies (optional)

14. Implement the Primal Averaging Frank-Wolfe algorithm (Algorithm 2) and the Momentum-guided Frank-Wolfe (Algorithm 3) strategies. Compare their performance with the regular Frank-Wolfe algorithm.

References

[WKP23] Elias Wirth, Thomas Kerdreux, and Sebastian Pokutta. Acceleration of frank-wolfe algorithms with open-loop step-sizes. arXiv preprint, `https://arxiv.org/abs/2205.12838` 2023.