Nom :
Prénom :
No étudiant :

Université Paris Cité
UFR Mathématiques et Informatique
2023-2024

M1 MMA - Optimisation sous Contraintes - Interro n°3 / TP évalué

Contexte général.

Vous allez implémenter l'algorithme Frank-Wolfe (FW) permettant de résoudre les problèmes d'optimisation de la forme

$$(\mathcal{P}) \qquad \inf_{x\in C} f(x),$$

où

- C est un convexe compact non vide de $\mathbb{R}^n$,
- $f : \Omega \to \mathbb{R}$ une fonction $\mathcal{C}^1$ sur Ω ouvert de $\mathbb{R}^n$ contenant C, convexe sur le convexe C, et telle que ∇f est L-Lipschitz sur C, pour un certain $L \geqslant 0$.

Définition algorithme FW.

L'algorithme de FW produit une suite $(x_k)_{k\in\mathbb{N}}$ d'éléments de C telle que pour $x_0 \in C$ choisi par l'utilisateur, on a

$$\forall k \in \mathbb{N}, \quad \begin{cases} s_k \in \operatorname{argmin}_{s\in C} \langle \nabla f(x_k),\, s\rangle, \\ \lambda_k = \frac{2}{k+2}, \\ x_{k+1} = (1-\lambda_k)x_k + \lambda_k s_k. \end{cases}$$

Contexte particulier à l'évaluation.

A partir de maintenant on suppose que $f : x \in \mathbb{R}^n \mapsto \frac{1}{2}\left\|Ax - y\right\|_2^2$ avec $A \in \mathcal{M}_{m,n}(\mathbb{R})$, $m, n \in \mathbb{N}^*$, $y \in \mathbb{R}^m$ et que C est la boule unité fermée pour la norme ℓ_1 i.e.

$$C = \{x \in \mathbb{R}^n \ / \ \left\|x\right\|_1 \leqslant 1\}.$$

Questions.

1. Montrer que l'on peut appliquer l'algorithme Frank-Wolfe à notre problème $(\mathcal{P})$.

2. Montrer que nos problèmes $(\mathcal{P})$ et

$$(\mathcal{L}) \qquad \inf_{s\in C} \langle \nabla f(\tilde{x}),\, s\rangle,$$

pour tout $\tilde{x} \in \mathbb{R}^n$ fixé, sont bien posés.

> On rappelle pour la suite que le problème $(\mathcal{L})$ admet au moins une solution parmi les points extrêmes de C, c'est-à-dire les éléments
> $$\operatorname{Ext}(C) = \{e_1, -e_1, e_2, -e_2, \ldots, e_n, -e_n\},$$
> où $\mathcal{B} = (e_1, \ldots, e_n)$ est la base canonique de $\mathbb{R}^n$. Plus précisément,
> $$s = -\operatorname{sign}\left((\nabla f(\tilde{x}))_{k_0}\right) e_{k_0} \quad \text{avec} \quad k_0 \in \operatorname{argmax}_{1\leqslant k\leqslant n}|(\nabla f(\tilde{x}))_k|,$$
> est solution de $(\mathcal{L})$. Voir la correction de l'interro no2 au besoin.

3. Implémenter l'algorithme Frank-Wolfe et l'utiliser pour la résolution de notre problème $(\mathcal{P})$ lorsque

$$n = m = 3, \quad A = I_3, \quad y = (3.4, 3.3, 3.3).$$

Indication : on pourra utiliser la commande Python v.argmin() *qui permet, lorsque v est un array, de renvoyer l'indice* k_0 *tel que* $v_{k_0} = \min_k v_k$.

4. Illustrer alors la convergence de l'algorithme Frank-Wolfe. <u>Bonus :</u> commenter la vitesse de convergence.

Correction.

1. L'ensemble C est une boule unité fermé d'une norme en dimension finie, donc C est non vide, compact en tant que fermé borné, et également convexe grâce à l'inégalité triangulaire puisque pour tout $x, x' \in C$, $t \in [0,1]$

$$\left\|(1-t)x + tx'\right\|_1 \leqslant \|(1-t)x\|_1 + \left\|tx'\right\|_1 = (1-t)\|x\|_1 + t\left\|x'\right\|_1 \leqslant (1-t) + t = 1,$$

d'où $(1-t)x + tx' \in C$.

La fonction f est $\mathcal{C}^2$ sur $\mathbb{R}^n$ donc en particulier $\mathcal{C}^1$. De plus pour tout $x \in \mathbb{R}^n$, on a $\nabla f(x) = A^T(Ax - y)$, puis $\nabla^2 f(x) = A^T A$. Pour tout $x \in \mathbb{R}^n$ et tout $h \in \mathbb{R}^n$, on a alors

$$\langle \nabla f(x)h,\, h\rangle = \left\langle A^T A h,\, h\right\rangle = \|Ah\|_2^2 \geqslant 0,$$

i.e. $\nabla^2 f(x) \succeq 0$. Comme c'est vrai pour tout $x \in \mathbb{R}^n$, on déduit que f est convexe sur $\mathbb{R}^n$ donc également sur l'ensemble convexe C. Ensuite pour tout $x, x' \in \mathbb{R}^n$, on a

$$\left\|\nabla f(x) - \nabla f(x')\right\|_2 = \left\|A^T(Ax-b) - A^T(Ax'-b)\right\|_2 = \left\|A^T A(x-x')\right\|_2 \leqslant \underbrace{\left\|A^T A\right\|_2}_{=L} \left\|x - x'\right\|_2,$$

ainsi ∇f est L-Lipschitzienne sur $\mathbb{R}^n$ donc en particulier sur C.

Les hypothèses d'application de l'algorithme FW sont bien satisfaites.

2. Les fonctions objectifs de ces problèmes d'optimisation sont continues et l'ensemble des contraintes C est compact, donc elles admettent un minimum global sur C et ainsi les problèmes sont bien posés.

3. Voir le notebook associé à ce document.

Quelques commentaires sur la condition d'arrêt utilisée dans l'algorithme Frank-Wolfe. Comme f est convexe sur le convexe C et différentiable, on sait que $x_k \in C$ est un minimum global de f sur C si et seulement si

(1) $$\forall s \in C, \quad \langle \nabla f(x_k),\, s - x_k\rangle \geqslant 0.$$

La condition (1) représente la condition d'optimalité d'ordre 1 associée au problème $(\mathcal{P})$ puisque C est convexe et comme déjà affirmé, c'est une CNS d'optimalité car f est en plus convexe sur le convexe C.

> Si x_k satisfait la condition (1) et $x_k \in \overset{\circ}{C}$ alors elle devient simplement $\nabla f(x_k) = 0$. Cependant il n'y a aucune raison, a priori, que $x_k \in \overset{\circ}{C}$ soit vraie, donc on ne peut utiliser ici la condition d'arrêt classique de l'optimisation sans contraintes : $\|\nabla f(x_k)\|_2 \leqslant \varepsilon$.

Ainsi on peut être sûr que tant que cette condition (1) n'est pas satisfaite, c'est-à-dire s'il existe $\tilde{s} \in C$ satisfaisant $\langle \nabla f(x_k),\, \tilde{s} - x_k\rangle < 0$, alors x_k n'est pas un minimum global de f sur C et donc l'algorithme doit continuer.

Comment trouver un tel $\tilde{s}$? On peut en prendre un au hasard, ou alors dans notre exemple on a $0 \in C$ donc si jamais $\langle \nabla f(x_k),\, -x_k\rangle < 0$, l'algorithme doit nécessairement continuer. Mais si le test $\langle \nabla f(x_k),\, -x_k\rangle < 0$ est FAUX, rien ne permet d'affirmer que $\langle \nabla f(x_k),\, s - x_k\rangle < 0$ est FAUX pour tout $s \in C$. Cependant il existe au moins un $\tilde{s} \in C$ bien particulier tel que si $\langle \nabla f(x_k),\, \tilde{s} - x_k\rangle \geqslant 0$ est VRAI, alors cette condition est également vrai pour tout $s \in C$. En effet, prenons

$$s_k \in \operatorname{argmin}_{s \in C} \langle \nabla f(x_k),\, s\rangle.$$

Un tel élément $s_k \in C$ existe bien, car on a vu que le problème $(\mathcal{L})$ admet au moins une solution. On a donc par définition

$$\forall s \in C, \quad \langle \nabla f(x_k),\, s\rangle \geqslant \langle \nabla f(x_k),\, s_k\rangle,$$

donc si $\langle \nabla f(x_k),\, s_k - x_k\rangle \geqslant 0$ est VRAI, alors

$$\forall s \in C, \quad \langle \nabla f(x_k),\, s - x_k\rangle \geqslant \langle \nabla f(x_k),\, s_k - x_k\rangle \geqslant 0.$$

et donc x_k est un minimum global de f sur C et l'algorithme peut-être arrêté.

Ce qui est formidable, c'est qu'un tel $s_k \in C$ est calculé à chaque itération par définition de l'algorithme Frank-Wolfe !

Il suffit donc de maintenir la boucle principale de l'algorithme TANT QUE $\langle \nabla f(x_k),\, s_k - x_k\rangle < 0$. Plus précisément, on ne s'attend pas à ce que l'on trouve un x_k tel que $\langle \nabla f(x_k),\, s_k - x_k\rangle \geqslant 0$ (puisque cela signifierait que x_k est exactement un minimum global de f sur C). On va plutôt maintenir la boucle principale TANT QUE $\langle \nabla f(x_k),\, s_k - x_k\rangle \leqslant -\varepsilon$ avec $\varepsilon > 0$ qui représente donc une tolérance sur l'arrêt de l'algorithme. Plus ε est choisi petit, plus l'algorithme renverra un x_k tel que $\langle \nabla f(x_k),\, s_k - x_k\rangle$ est proche de 0 (tout en étant négatif) et donc plus on s'attend à ce que x_k soit proche d'un minimum global de f sur C.

4. Voir le notebook associé à ce document. On note que l'algorithme converge à une vitesse en $\frac{1}{k}$. Voir le code pour plus de détails.