

Optimisation Convexe : Algorithmes et Applications en Apprentissage

Contrôle terminal. Durée : 2h. Tous les documents sont autorisés.

Dans tout le sujet ci-dessous, $\|x\|_p$ désigne la norme ℓ_p : $\|x\|_p := (\sum_i |x_i|^p)^{1/p}$ si $p < \infty$, $\|x\|_\infty := \max_i |x_i|$.

Partout dans les solutions on écrira $\|\cdot\|$ pour $\|\cdot\|_2$, par simplicité.

Exercice 1 (6 points). Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ la fonction définie par $f(x) := \sqrt{1 + \|x\|_2^2}$. Calculer f^* . En quels points le sous-différentiel de f^* est-il non-vidé ?

Solution : On a $f^*(y) = \sup_x x \cdot y - \sqrt{1 + \|x\|_2^2}$. On calcule le gradient du terme qu'on maximise, qui est concave. On obtient $y - x/\sqrt{1 + \|x\|_2^2}$. On distingue trois cas

— Si $\|y\| < 1$ alors le point

$$x = \frac{y}{\sqrt{1 - \|y\|^2}}$$

annule ce gradient, et donc maximise parce que la fonction à maximiser est concave. En remplaçant on obtient

$$f^*(y) = y \cdot \frac{y}{\sqrt{1 - \|y\|^2}} - \sqrt{1 + \frac{\|y\|^2}{1 - \|y\|^2}} = -\sqrt{1 - \|y\|^2}.$$

— Si $\|y\| > 1$ on prend $x_n = ny$ et on voit que l'on a $x_n \cdot y - \sqrt{1 + \|x_n\|^2} \geq n\|y\|^2 - 1 - n\|y\| \rightarrow \infty$ (en utilisant $\|y\|^2 > \|y\|$). Le sup vaut donc $+\infty$ et on trouve $f^*(y) = +\infty$.

— Si $\|y\| = 1$ on a $x \cdot y \leq \|x\| \leq \sqrt{1 + \|x\|^2}$, ce qui prouve que le sup vaut au plus 0. En prenant là aussi $x_n = ny$ on trouve que le sup vaut au moins

$$\lim_{n \rightarrow \infty} n\|y\|^2 - \sqrt{1 + n^2\|y\|^2} = \lim_{n \rightarrow \infty} \frac{n^2\|y\|^4 - 1 - n^2\|y\|^2}{n\|y\|^2 + \sqrt{1 + n^2\|y\|^2}} = \lim_{n \rightarrow \infty} \frac{-1}{n + \sqrt{1 + n^2}} = 0,$$

où dans la dernière égalité on a utilisé $\|y\| = 1$. Dans ce cas on a donc $f^*(y) = 0$.

En conclusion on trouve

$$f^*(y) = \begin{cases} -\sqrt{1 - \|y\|^2} & \text{si } \|y\| \leq 1; \\ +\infty & \text{sinon.} \end{cases}$$

Cette fonction est convexe et différentiable dans la boule ouverte $B(0,1)$. Son sous-différentiel y est donc non-vidé. Le sous-différentiel est vide pour $\|y\| > 1$ parce que $f^*(y) = +\infty$. Le sous-différentiel est vide également pour $\|y\| = 1$ parce que, si $v \in \partial f^*(y)$ et $\|y\| = 1$, on aurait $-\sqrt{1 - \|z\|^2} \geq v \cdot (z - y)$ et, en prenant $z = (1 - \varepsilon)y$ pour $\varepsilon < 1$ on trouverait $\sqrt{1 - (1 - \varepsilon)^2} \leq C\varepsilon$, ce qui est absurde parce que $\sqrt{1 - (1 - \varepsilon)^2} = O(\sqrt{\varepsilon})$.

Exercice 2 (6 points). Étant donné une matrice $A \in \mathbb{R}^{m \times n}$ on considère les deux problèmes suivants :

$$\min \left\{ \sum_{i=1}^n x_i \log(x_i) : x \in \mathbb{R}^n, x_i \geq 0 \text{ for all } i, \|Ax\|_\infty \leq 1 \right\} \quad \text{et} \quad \max \left\{ -\sum_{i=1}^n e^{-(A^t \xi)_i - 1} - \|\xi\|_1 : \xi \in \mathbb{R}^m \right\}.$$

1. Expliquer pourquoi les deux problèmes sont en dualité.
2. Suggérer quels algorithmes on pourrait utiliser pour résoudre l'un et l'autre et discuter leur faisabilité.

Solution :

— On sait que le dual de $\min f(x) + g(Ax)$ est $\max -f^*(-A^t\xi) - g^*(\xi)$. Ici on a

$$f(x) = \begin{cases} \sum_{i=1}^n x_i \log(x_i) & \text{si } x_i \geq 0 \text{ for all } i, \\ +\infty & \text{sinon,} \end{cases}$$

ainsi que

$$g(y) = \begin{cases} 0 & \text{si } \|y\|_\infty \leq 1, \\ +\infty & \text{sinon.} \end{cases}$$

En 1D, la transformée de $x \mapsto x \log x$ est $y \mapsto e^{y-1}$ et celle de la fonction valant 0 sur $[-1, 1]$ et $+\infty$ ailleurs est la fonction valeur absolue. Les transformées peuvent être calculées composante par composante. On obtient donc $f^*(z) = \sum_i e^{z_i-1}$. En ce qui concerne g^* , on a $g^*(\xi) = \pm \sum_j |\xi_j| = \|\xi\|_1$ et on retrouve donc le problème dual cherché.

— Le premier problème est la minimisation d'une fonction convexe sous un certain nombre de contraintes d'inégalités de la forme $(Ax)_j \leq 1$ ou $(Ax)_j \geq -1$. Il peut être traité à l'aide de l'algorithme d'Uzawa, qui demande à résoudre des problèmes d'optimisation de la forme $\min f(x) + \sum_j \lambda_j^+ ((Ax)_j - 1) + \sum_j \lambda_j^- (-(Ax)_j - 1)$. Comme la fonction f est séparable (elle est la somme de plusieurs fonctions chacune appliquée à une seule coordonnée x_i) et qu'on lui ajoute des fonctions linéaires, qui sont donc également séparables, pour trouver le minimiseur à chaque étape d'Uzawa il suffit de minimiser pour chaque i une fonction de la forme $x_i \log(x_i) + c_i x_i$ dont la solution est $x_i = e^{-c_i-1}$. On peut donc itérer un certain nombre d'opérations explicites pour approcher la solution.

Le deuxième problème, par contre, n'a pas de contraintes, mais contient un terme non-lisse $\|\xi\|_1$. Il peut être traité par l'algorithme de gradient proximal parce qu'on connaît explicitement l'opérateur proximal de la fonction g^* et que l'autre fonction est C^2 .

Exercice 3 (8 points). Étant donné une matrice $A \in \mathbb{R}^{m \times n}$ et deux vecteurs $y \in \mathbb{R}^n$ et $c \in \mathbb{R}^m$ on veut projeter y sur l'ensemble $C = \{x \in \mathbb{R}^n : Ax \leq c\}$, où l'inégalité entre vecteurs est à prendre composante par composante. On regarde donc

$$\min \left\{ \frac{\|x - y\|_2^2}{2} : Ax \leq c \right\}.$$

1. Un minimiseur x existe-t-il? est-il unique?
2. Suggérer un algorithme pour résoudre ce problème, en décrivant explicitement ses différentes étapes.
3. Peut-on utiliser l'algorithme de gradient projeté pour résoudre ce problème?

Solution :

- L'ensemble C est déterminé par un nombre fini d'inégalités linéaires larges. Il est donc un polyèdre convexe fermé, à condition d'être non-vide. La projection sur un convexe fermé existe toujours (la fonction qu'on minimise est la distance au carré, qui est coercive) et est unique (la fonction qu'on minimise est strictement convexe).
- Comme on a des contraintes d'inégalités, on peut utiliser l'algorithme d'Uzawa. On cherche une suite (λ_k, x_k) définie par

$$x_k = \operatorname{argmin} \frac{\|x - y\|_2^2}{2} + \lambda_k \cdot (Ax - c); \quad \lambda_{k+1} = [\lambda_k + \tau(Ax_k - c)]_+,$$

où la partie positive est prise composante par composante. On peut trouver explicitement x_k et on trouve

$$x_k = y - A^t \lambda_k.$$

Si $0 < \tau < 2\|A\|^{-2}$ la suite x_k converge vers la solution du problème (on utilise que la fonction $x \mapsto \frac{\|x-y\|_2^2}{2}$ est 1-elliptique).

- On ne peut pas utiliser l'algorithme de gradient projeté pour calculer une projection, parce qu'il demande à savoir déjà calculer la projection.

Exercice 4 (10 points). On s'intéresse à l'approximation d'un vecteur $y \in \mathbb{R}^n$ donné par un vecteur à composantes positives.

1. Trouver les coordonnées du point $\bar{x}(y)$ solution de

$$\min \left\{ \frac{\|x - y\|_2^2}{2} : x \geq 0 \right\}.$$

2. Si on accepte que certaines composantes soient négatives, on peut se donner $\lambda > 0$ et résoudre

$$\min \left\{ \frac{\|x - y\|_2^2}{2} + \lambda \sum_{i=1}^n [x_i]_- \right\},$$

où $[s]_- := \max\{-s, 0\}$ indique la partie négative du nombre $s \in \mathbb{R}$. Prouver que ce problème admet une unique solution et donner la formule pour la trouver, en termes de y et λ . On appelle cette solution $\bar{x}(\lambda, y)$.

3. Prouver que l'on a $\lim_{\lambda \rightarrow +\infty} \bar{x}(\lambda, y) = \bar{x}(y)$.
4. Étant donné une matrice $A \in \mathbb{R}^{m \times n}$ on veut maintenant approcher un vecteur $y \in \mathbb{R}^m$ par un vecteur de la forme Ax où x est un vecteur à composantes positives, ou dont peu des composantes sont négatives. On considère alors

$$\min \left\{ \frac{\|Ax - y\|_2^2}{2} + \lambda \sum_{i=1}^n [x_i]_- \right\}.$$

Suggérer un algorithme pour résoudre ce problème, en décrivant explicitement ses différentes étapes.

5. Pour la convergence de l'algorithme choisi il est sans doute nécessaire de supposer que le problème d'optimisation ci-dessus admet une solution. Prouver qu'il en admet une lorsque A est une matrice dont tous les coefficients sont strictement positifs.

Solution :

- La projection d'un nombre réel sur \mathbb{R}_+ est égale à sa partie positive. Les projections sur un rectangle peuvent être calculées composantes par composantes, on a donc $(\bar{x}(y))_i = [y_i]_+$ pour tout i .
- On minimise une fonction qui est la somme d'une fonction quadratique (qui est elliptique) et d'un certain nombre de parties négatives, la somme est donc une fonction elliptique et elle admet un unique point de minimum. En 1D on peut résoudre

$$\min h(x) := \frac{|x - y|^2}{2} + \lambda [x]_-$$

en regardant sa dérivée : on a $h'(x) = x - y - \lambda$ pour $x < 0$ et $h'(x) = x - y$ pour $x > 0$. La fonction h n'est pas différentiable en $x = 0$. On distingue trois cas : si $y < -\lambda$ alors le point $x = y + \lambda$ annule la dérivée et est donc la solution ; si $y > 0$ alors le point $x = y$ annule la dérivée et est donc la solution ; si $y \in [-\lambda, 0]$ la solution existe, mais ne peut être ni strictement négative ni strictement positive et est donc nulle. On considère la fonction T_λ définie par

$$T_\lambda(y) = \begin{cases} y + \lambda & \text{si } y < -\lambda, \\ 0 & \text{si } y \in [-\lambda, 0], \\ y & \text{si } y > 0. \end{cases}$$

La solution de $\min \frac{|x - y|^2}{2} + \lambda [x]_-$ est donc donnée par $x = T_\lambda(y)$. En dimension supérieure, la solution $(\bar{x}(\lambda, y))_i$ est donnée par $(\bar{x}(\lambda, y))_i = T_\lambda(y_i)$ puisque la fonction que l'on minimise est séparable variable par variable.

- On voit bien que lorsque $\lambda > \|y\|_\infty$ on a $T_\lambda(y_i) = [y_i]_+$ pour tout i , ce qui prouve $\lim_{\lambda \rightarrow +\infty} \bar{x}(\lambda, y) = \bar{x}(y)$ (en fait c'est mieux que ça, à partir d'un certain λ on a l'égalité $\bar{x}(\lambda, y) = \bar{x}(y)$).

- On peut appliquer l'algorithme de gradient proximal, puisque ce qu'on vient de calculer est de fait l'opérateur proximal de la fonction $g(x) = \lambda \sum_{i=1}^n [x_i]_-$. On l'a calculé pour $\tau = 1$, mais on peut facilement vérifier que l'on a $\text{Prox}_{\tau, g} = \text{Prox}_{1, \tau g}$. Du coup, si on indique par \mathcal{T}_λ la fonction de \mathbb{R}^n vers \mathbb{R}^n qui applique T_λ à chaque composante et si on appelle $f(x) = \frac{\|Ax-y\|^2}{2}$, l'algorithme de gradient proximal demande à considérer la suite donnée par

$$x_{k+1} = \mathcal{T}_{\tau\lambda}(x_k - \tau \nabla f(x_k)) = \mathcal{T}_{\tau\lambda}(x_k - \tau A^t(Ax_k - y)).$$

- Le problème de la fonction $x \mapsto F(x) := f(x) + g(x) = \frac{\|Ax-y\|^2}{2} + \lambda \sum_{i=1}^n [x_i]_-$ est qu'elle n'est peut-être pas coercive. Prenons une suite $x(n)$ de vecteurs et supposons $\|x(n)\| \rightarrow \infty$. Il faut montrer, pour montrer que F est coercive, que l'on a $F(x(n)) \rightarrow +\infty$. Comme $F \geq 0$, la seule possibilité pour ne pas avoir $F(x(n)) \rightarrow +\infty$ est que $g(x(n))$ soit bornée, donc on aurait $[(x(n))_i]_- \leq C$ pour tout n et tout i . Ensuite, on utilise $F(x) \geq \frac{1}{2} \|A(x) - y\|^2 \geq \frac{1}{2} |\sum_j A^{ij} x_j - y_i|^2$ et donc pour pas avoir $F(x(n)) \rightarrow +\infty$ il faudrait que $\sum_j A^{ij} x(n)_j$ soit borné. Mais comme les A^{ij} sont strictement positifs et que l'on a $x(n)_j \geq [x(n)_j]_+ - C$ on trouve une borne supérieure sur chaque $[x(n)_j]_+$. Finalement tant la partie positive que la partie négative des composantes de $x(n)$ serait bornée, ce qui est en contradiction avec $\|x(n)\| \rightarrow \infty$. Donc, si $A^{ij} > 0$ pour tout i, j , la fonction F est coercive, et continue, et admet donc un minimiseur.