# Rapport MVA : Convex Optimization - HWK 3

Plassier Vincent

## 1 Réécriture du problème de minimisation

L'objectif de ce devoir maison est de résoudre le problème (LASSO), c'est-à-dire : minimiser$\frac{1}{2}\|Xw-y\|_2^2+\lambda\|w\|_1$. En posant $z=Xw+y$, ceci équivaut à

$$\begin{cases}\text{minimize } \frac{1}{2}\|Xw-y\|_2^2+\lambda\|w\|_1\\ \text{subject to } z=Xw+y\end{cases}$$

Ecrivons le lagrangien de ce problème :

$$\begin{aligned}\mathcal{L}(w,z,\mu)&=\frac{1}{2}\|z\|_2^2+\lambda\|w\|_1+\mu^T(Xw-y-z)\\ &\overset{\lambda\geq 0}{=}\inf_z(\frac{1}{2}\|z\|_2^2-\mu^Tz)+\mu^Ty+\lambda\inf_w(\|w\|_1+\frac{\mu^T}{\lambda}Xw)\\ &=-\frac{\|\mu\|_2^2}{2}+\mu^Ty+\lambda\inf_w(\|w\|_1+\frac{\mu^T}{\lambda}Xw)\end{aligned}$$

Définissons la fonction $f:w\longmapsto\|w\|_1$ et notons $f^*$ son conjugué. Soit $z\in\mathbb{R}^d$, si $\|z\|_\infty\leq 1$, alors :

$$\|w\|_1-w^Tz=\sum_{i=1}^d(|w_i|-z_iw_i)\geq\sum_{i=1}^d|w_i|(1-|z_i|)\geq 0$$

En prenant $w=0$, on obtient que $\lambda\inf_w(\|w\|_1+\frac{\mu^T}{\lambda}Xw)$=0.
Supposons désormais que $\|z\|_\infty>1$. Pour tout $R>0$, en prenant $z_i$ et $w\in\mathbb{R}^d$ tels que $|z_i|=\|z\|_\infty$ et $w_j=R\times\mathbf{1}_{j=i}\text{sgn}(z_i)$. On a :

$$f^*(z)\leq R(\|z\|_\infty-1)\underset{R\to+\infty}{\longrightarrow}-\infty$$

On en déduit que :

$$\inf_{w,z}\mathcal{L}(w,z,\mu)=\begin{cases}\mu^Ty-\frac{\|\mu\|_2^2}{2}\text{ si }\|\mu^TX\|_\infty\leq\lambda\\ -\infty\text{ sinon}\end{cases}$$

Comme $\max(-z)=-\min(-z)$ et que $\|\mu^TX\|_\infty\leq\lambda$. La deuxième condition équivaut à $\left|\sum_{i=1}^n\mu_iX_{i,j}\right|\leq\lambda$.
En posant $Q=\frac{\text{Id}}{2}$, $v=\mu$, $p=-y$, $b=(\lambda,\dots,\lambda)\in\mathbb{R}^{2d}$, et $A=(X,-X)\in R^{2d\times d}$. On obtient que le problème dual est équivalent à

$$\begin{cases}\text{minimize } v^TQv+p^Tv\\ \text{subject to } Av\preceq b\end{cases}$$

De plus, les contrainte sont affines donc qualifiées, par le théorème de Slatter on en déduit que le problème primal est équivalent au problème dual. D'où le problème (LASSO) correspond au (QP).

## 2 Implémentation la méthode par barrière

Afin d'implémenter la méthode de Newton pour résoudre l'algorithme centering_step, il est indispensable de déterminer le gradient ainsi que la hessienne de $v \longmapsto v^T Q v + p^T v - \sum_{i=1}^{2d} \ln(b_i - \sum_{i=1}^{d} a_{i,k} v_k)$. Calculons les dérivées partielles :

$$\forall i \in \{1, \dots, d\}, \quad -\ln\left(b_i - \sum_{i=1}^{d} a_{i,k} v_k\right) \xmapsto{\partial l} \frac{a_{i,l}}{b_i - \sum_{i=1}^{d} a_{i,k} v_k} \xmapsto{\partial l \partial m} \frac{a_{i,l} a_{i,m}}{\left(b_i - \sum_{l=1}^{d} a_{i,k} v_k\right)^2}$$

On en déduit que le gradient est :

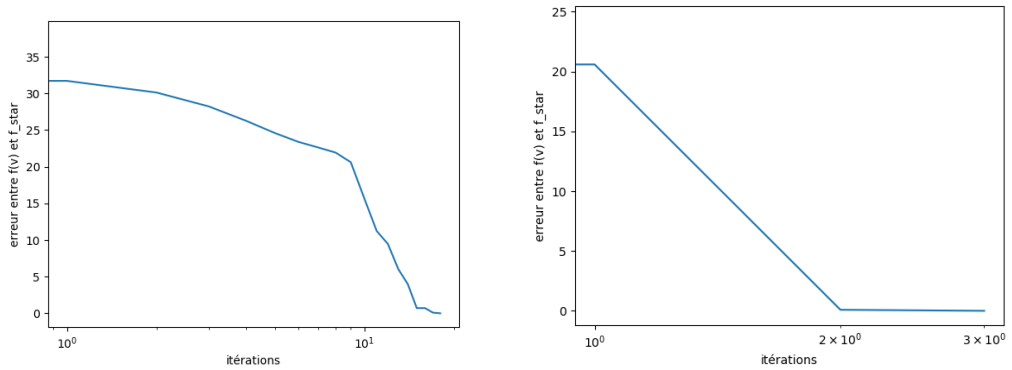
$$grad_l = t(Qv + p)_l + \sum_{i=1}^{d} \frac{a_{i,l}}{b_i - \sum_{i=1}^{d} a_{i,k} v_k}$$

Et également que la matrice hessienne du problème est :

$$H_{l,m} = 2tQ_{l,m} + \sum_{i=1}^{d} \frac{a_{i,l} a_{i,m}}{\left(b_i - \sum_{l=1}^{d} a_{i,k} v_k\right)^2}$$

## 3 Expérimentations de l'algorithme

Désormais, nous fixons $\lambda = 10$ et nous penons une dimension $d \sim \mathcal{U}([10, 100])$, ainsi que $X \sim \mathcal{N}(0_{d,d}, \mathrm{Id}_d)$ et $y \sim \mathcal{N}(0_d, \mathbf{1}_d)$.
Affichons l'erreur $f(v_t) - f^\star$ pour $\mu = 2$ ainsi que pour $\mu = 110$ :




(a) $\mu = 2$ (b) $\mu = 110$

FIGURE 1: $\|f(v_t) - f^\star\|_\infty$ en fonction de l'itération

Nous constatons que l'algorithme barr_method effectue un nombre d'itérations égal à int $\left[\frac{\ln(m)-\ln(\varepsilon)}{\ln(\mu)}\right] + 1$ avec $m$ le nombre de contraintes (ici $m = 2d$). On en déduit que le nombre d'itérations dans barr_method diminue lorsque $\mu$ augmente. Cependant pour choisir $\mu$, il faut également prendre en compte le nombre total d'utilisation de la méthode de Newton. En effet, celle-ci nécessite l'inversion de la Hessienne et est donc computationellement plus lourde. De plus, nous remarquons que la précision obtenue est meilleure lorsque $\mu^{\text{Nb\_iterations}}$ est grand. Donc à budget $k$ fixé, il faut mieux choisir $\mu \simeq \left(\frac{m}{\varepsilon}\right)^{1/k}$ tout en vaillant à ne pas trop utiliser la méthode de Newton.

Pour une tolérance $\varepsilon = 0.001$ avec un choix $\mu = 110$, l'algorithme converge en 3 itérations dans barr_method. Par contre, pour $\mu = 2$ l'algorithme nécessite 16 itérations. Affichons le nombre d'itérations qu'effectue l'algorithme barr_method lorsque $\varepsilon = 0.001$ et que $\mu$ varie :

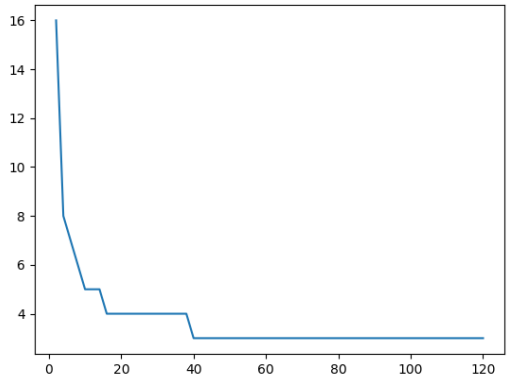



FIGURE 2: Nombre d'itérations dans barr_method en fonction de $\mu$

Nous remarquons que le nombre d'itérations dans l'algortihme barr_method est bien celui attendu, c'est-à dire int $\left[\frac{\ln(m)-\ln(\varepsilon)}{\ln(\mu)}\right] + 1$.
Maintenant, représentons le nombre d'utilisation de la méthode de Newton en fonction de $\mu$ en pour d=300 :

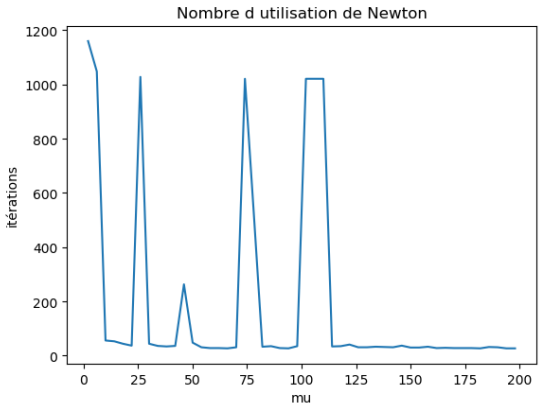



FIGURE 3: Nombre d'itérations dans Newton en fonction de $\mu$

Nous constatons que la complexité de l'algorithme peut devenir désastreuse en utilisant la recherche de Wolfe.