Optimisation convexe : DM 3

① Interior Point Method

- $\phi_t(x) = t\left(\frac{1}{2}x^T Q x + p^T x\right) - \sum_{i=1}^{m} \log\left(b - a_i^T x\right)$

où a_i est la $i^{ème}$ ligne de A.

$$\nabla \log\left(b - a_i^T x\right) = \frac{\nabla(b - a_i^T x)}{b - a_i^T x} = \frac{-a_i}{b - a_i^T x}$$

donc $$\boxed{\nabla \phi_t(x) = t(Qx + p) + \sum_{i=1}^{m} \frac{a_i}{b - a_i^T x}}$$

$$\nabla^T\left(\frac{1}{b - a_i^T x}\right) = -\frac{\nabla^T(b - a_i^T x)}{(b - a_i^T x)^2} = \frac{a_i^T}{(b - a_i^T x)^2}$$

donc $$\boxed{\nabla^2 \phi_t(x) = tQ + \sum_{i=1}^{m} \frac{a_i a_i^T}{(b - a_i^T x)^2}}$$

③ Support Vector Machine Problem

1) Primal: On cherche $\begin{pmatrix} w \\ \xi \end{pmatrix} \in Int\mathcal{D}$ tel que $y_i(w^T x_i) \geq 1-\xi_i$ et $\xi_i \geq 0$, $i=1,\dots,m$

Ici $\mathcal{D} = \mathbb{R}^d \times \mathbb{R}^m$ $(w \in \mathbb{R}^d, \xi \in \mathbb{R}^m)$ donc $Int\mathcal{D} = \mathbb{R}^d \times \mathbb{R}^m$

donc l'ensemble des points strictement faisables est : $\left\{ \begin{pmatrix} w \\ \xi \end{pmatrix} \in \mathbb{R}^d \times \mathbb{R}^m \,/\, \xi_i > 1-y_i(w^T x_i) \text{ et } \xi_i > 0 \right\}_{i=1,\dots,m}$

On peut donc prendre par exemple : $w = 0_{\mathbb{R}^d}$ et $\xi = 2 \times 1_m \leftarrow \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} m$ fois

car alors $\xi_i > 0$ et $\xi_i > 1 - y_i(0_{\mathbb{R}^d}^T x_i) = 1$.

Dual: On cherche $\lambda \in Int\mathcal{D}$ tel que $0 < \lambda < \frac{1}{2m}$

Ici $\mathcal{D} = \mathbb{R}^m$ donc $Int\mathcal{D} = \mathbb{R}^m$. L'ensemble de points strictement faisables est donc :

$\left\{ \lambda \in \mathbb{R}^m \,/\, 0 < \lambda < \frac{1_m}{2m} \right\}$. On peut donc prendre par exemple :

$$\lambda = \frac{1_m}{2 \times 2m}$$

2) Reformulons le dual et le primal pour les écrire sous la forme $\min_x \frac{1}{2} x^T Q x + p^T x$ s.t $Ax \leq b$:

Primal: $\frac{1}{2}\|w\|_2^2 + \frac{1}{2m}\sum_{i=1}^{m} \xi_i = \frac{1}{2} w^T w + \frac{1}{2m} 1_m^T \xi$

donc on pose $Q = \begin{pmatrix} I_d & 0 \\ 0 & 0 \end{pmatrix} \begin{matrix} d \\ m \end{matrix}$ où $I_d = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} d$

$Q \in M_{m+d}$ et est symétrique positive

$p = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{2m} \\ \vdots \\ \frac{1}{m} \end{pmatrix}_m^d$ On a donc $\frac{1}{2}\|w\|_2^2 + \frac{1}{2m}\sum_{i=1}^m \xi_i = \frac{1}{2}(w,\xi) Q \begin{pmatrix} w \\ \xi \end{pmatrix} + p^T \begin{pmatrix} w \\ \xi \end{pmatrix}$

Ensuite par les contraintes: on pose $A = \begin{pmatrix} -\mathrm{diag}(y)X & -I_m \\ 0 & -I_m \end{pmatrix}_m^m$, $A \in M_{2m,m+d}$

où $\mathrm{diag}(y) = \begin{pmatrix} y_1 & & 0 \\ & \ddots & \\ 0 & & y_m \end{pmatrix}$ donc $\mathrm{diag}(y)X = \begin{pmatrix} x_1^T y_1 \\ \vdots \\ x_m^T y_m \end{pmatrix}$

et $I_m = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}_m^m$

$b = \begin{pmatrix} -\frac{1}{m} \\ 0_{\mathbb{R}^m} \end{pmatrix}$ et les contraintes s'écrivant alors: $A \begin{pmatrix} w \\ \xi \end{pmatrix} \leq b$

On a donc $\min_{(w,\xi)} \frac{1}{2}\|w\|_2^2 + \frac{1}{2m}\sum_{i=1}^m \xi_i$ st $y_i(w^T x_i) \geq 1 - \xi_i$, $\xi_i \geq 0$ $\forall i = 1 \dots m$

qui s'écrit: $\min_{(w,\xi)} \frac{1}{2}(w,\xi) Q \begin{pmatrix} w \\ \xi \end{pmatrix} + p^T \begin{pmatrix} w \\ \xi \end{pmatrix}$ st $A \begin{pmatrix} w \\ \xi \end{pmatrix} \leq b$

<u>Duale:</u> On prend $-\left(-\frac{1}{2}\left\|\sum_{i=1}^m \lambda_i y_i x_i\right\|_2^2\right) + \frac{1}{m}\mathbb{1}^T\lambda$ pour avoir le problème de minimisation équivalent

$$\frac{1}{2}\left\|\sum_{i=1}^m \lambda_i y_i x_i\right\|_2^2 = \frac{1}{2}\left(\sum_{i=1}^m \sum_{j=1}^m \lambda_i \lambda_j y_i y_j x_i^T x_j\right)$$

On pose $B = \mathrm{diag}(y)X$ et alors on pose $Q = BB^T$, $Q \in M_m$ et est symétrique positive

$p = -\mathbb{1}_m$

On a donc $\frac{1}{2}\left\|\sum_{i=1}^{m}\lambda_i y_i x_i\right\|_2^2 - 1_m^T\lambda = \frac{1}{2}\lambda^T Q\lambda + p^T\lambda$

Puis pour les contraintes : on pose $A = \begin{pmatrix} I_m \\ -I_m \end{pmatrix}_m^m \quad A \in \mathcal{M}_{2m,m}$

et $b = \begin{pmatrix} \frac{1}{2m} 1_m \\ 0_{\mathbb{R}^m} \end{pmatrix}_m^m$ et les contraintes s'écrivant alors : $A\lambda \leq b$

On a donc $\min_\lambda \frac{1}{2}\left\|\sum_{i=1}^{m}\lambda_i y_i x_i\right\|_2^2 - 1_m^T\lambda$ s.t. $0 \leq \lambda \leq \frac{1}{2m}$ qui est

équivalent au dual qui s'écrit : $\min_\lambda \frac{1}{2}\lambda^T Q\lambda + p^T\lambda$ s.t. $A\lambda \leq b$.

Implémentation / Commentaires:

Tous les graphiques sont en annexe. On utilise toujours une tolérance de 10^{-3}.

Test des méthodes sur une fonction très simple: (fichier test_dampedpy et test_LS.py)

On va commencer par étudier une exemple très simple avant la question 3).

On étudie le problème: $\min_x \frac{1}{2}x^2 + x \quad s.t. \quad x \geq 0 \quad , x \in \mathbb{R}$

On a donc $Q=1$, $p=1$, $A=-1$ et $b=0$

On cherche son dual: le Lagrangien s'écrit, $L(x,\lambda) = \frac{1}{2}x^2 + x - \lambda x$ pour $\lambda \geq 0$ et $x \in \mathbb{R}$

La fonction duale est: $g(\lambda) = \inf_{x \in \mathbb{R}} \left(\frac{1}{2}x^2 + x(1-\lambda) \right)$, $\lambda \geq 0$ (L convexe en x)

$\frac{\partial L(x,\lambda)}{\partial x} = 0 \iff x = \lambda - 1$.

Donc $g(\lambda) = \frac{1}{2}(\lambda-1)^2 - (\lambda-1)^2 = -\frac{1}{2}(\lambda-1)^2 = -\frac{1}{2}\lambda^2 + \lambda - \frac{1}{2}$

et ainsi, le problème dual est: $\max_{\lambda \in \mathbb{R}} -\frac{1}{2}\lambda^2 + \lambda - \frac{1}{2} \quad s.t. \quad \lambda \geq 0$.

On a $x_0 = 1$ et $\lambda_0 = 1$ des points strictement faisables pour le primal et le dual respectivement.

$\eta = f_0(x_0) - g(\lambda_0) = \frac{1}{2} + 1 - 0 = \frac{3}{2}$ le "duality gap" de départ.

On prend donc $t_0 = \frac{m}{2} = \frac{1}{2} = \frac{2}{3}$ (m ← nb de contraintes) comme t de départ de la méthode log-barrière. On prend aussi x_0 en initialisation.

On sait en utilisant les KKT conditions que :

$$\begin{cases} -x \leq 0 \\ \lambda \geq 0 \\ -\lambda x = 0 \\ x = \lambda - 1 \end{cases} \Leftrightarrow \begin{cases} x \geq 0 \\ \lambda \geq 0 \\ \lambda(\lambda - 1) = 0 \\ x = \lambda - 1 \end{cases} \Leftrightarrow \begin{cases} x \geq 0 \\ \lambda \geq 0 \\ \lambda = 0 \text{ ou } \lambda = 1 \\ x = \lambda - 1 \end{cases}$$

donc le x optimal $x^* = 0$. On va lancer la méthode log-barrière en utilisant "damped Newton" puis une autre fois en utilisant "LS Newton" pour déjà vérifier que les deux méthodes fonctionnent dans ce cas très simple. (On utilise $\alpha = 0{,}01$ et $\beta = 0{,}5$ pour "LS Newton")

On trace également les courbes "duality gap versus iterations" pour $\mu = 2, 15, 50$ et 100.

On observe déjà que le x optimal est très proche de 0 pour chaque μ et pour les deux méthodes (utilisant "damped Newton" et "LS Newton").

On peut observer ensuite sur les graphes que pour les deux méthodes, quand μ est petit, il y a plus d'"outer iterations" et moins d'"inner iterations" et inversement quand μ est grand. On observe également que pour $\mu = 2$, l'algorithme se termine dans les 25-30 itérations dans les deux méthodes alors qu'il se termine en une quinzaine d'itérations (pour les deux méthodes) pour $\mu = 15, 50$ et 200.

On peut donc déjà être rassuré sur l'implémentation des deux méthodes qui donnent des résultats cohérents dans cet exemple très simple (et notamment sur le fait que

de mêmes jusqu'à 100

le marché d'itérations ne voisie pas beaucoup pour μ au-delà de 10 et qu'il est plus grand pour μ petit).

3) On va maintenant tester plus sérieusement nos algorithmes en effectuant la classification du dataset Iris (pour les classes Iris-versicolor et Iris-virginica). Tout d'abord, on centre nos données en leur soustrayant la moyenne des données. On affecte ensuite à la classe Iris-versicolor le label -1 et le label 1 à la classe Iris-virginica. On sélectionne ensuite aléatoirement les données (80% du dataset) avec lesquelles on va chercher l'hyperplan (passant par l'origine) séparant le "mieux" ces données selon notre critère de risque. Le reste des données sera utilisé pour tester la performance du classifieur.

Pour initialiser la méthode log-barrière on va utiliser les points strictement faisables qu'on avait trouvé précédemment (question 1) SVM problem) afin de calculer la duality gap initiale et on prend alors $t_0 = \frac{m}{2}$ (m = nbre de contraintes). On prend également le point strictement faisable du primal en initialisation ainsi que $\mu = 10$ (car on a vu en cours et vérifié ici dans l'exemple très simple que le choix de μ n'était pas critique pour μ dans un assez large intervalle (environ 3 à 100) donc prendre μ vers 10-20 est satisfaisant).

On prend aussi dans le cas où LS Newton est utilisé $\alpha = 0{,}01$ et $\beta = 0{,}5$.

On obtient à la fin de la méthode, une solution w qu'on utilise pour classer les points des 20% du dataset restant avec la règle : $w^T x_i > 0 \rightarrow$ label 1, $w^T x_i \leq 0 \rightarrow$ label -1.

On a fait cela pour les deux méthodes log-barrière (avec damped Newton et LS Newton) et pour Z allant de 0.1 à 10 par pas de 0,1. On trace alors la courbe représentant l'erreur de classification des 20% restant en fonction de Z. On relance un calcul comme de fois (6) fait l'algorithme pour déterminer une tendance générale de l'erreur en fonction de Z. (Remarque : Taux d'erreur $= \frac{1}{m_1} \sum_{i=1}^{m_1} \mathbb{1}_{\{y_i \neq g(x_i)\}}$ où m_1 : nb de données dans les 20% du dataset restant, $g(x_i) = 2\,\mathbb{1}_{\{w^T x_i > 0\}} - 1$)

On observe sur les graphiques pour les deux méthodes une tendance générale qui semble être que le taux d'erreur augmente rapidement avec Z entre 0.1 et 2 environ puis plafonne ensuite. Cependant, même si le taux semble augmenter rapidement entre 0.1 et 2 il ne semble pas dépasser généralement les 15%. De cette tendance générale sur nos 12 graphiques on en retire donc qu'il semblerait judicieux de prendre un Z petit ce qu'on fera pour la question 4) en prenant $Z = 0.1$.

On termine cette question en générant délibérément un nouveau dataset très. Pour cela on génère des données centrées par des gaussiennes de $\mathbb{R}^4$ avec comme matrice

de variance-covariance, la matrice de variance-covariance empirique de nos données du dataset d'origine. On utilise ensuite l'hyperplan séparateur obtenu (avec $C=0.1$) via les 80% de notre dataset d'origine afin de classer les données générées. Ainsi, on "décante" nos nouvelles données en leur gardant la moyenne de nos données d'origine.

4) Dans cette question, toujours en utilisant le 3ème dataset, on va tracer le "duality gap versus itérations" pour les problèmes primal et dual résolus par la méthode log-barrière. On utilise donc pour être consistant avec la question précédente les 80% du dataset. On prend le même t_0 et les mêmes points initiaux strictement faisables que précédemment ainsi que $C=0.1$ (et $\alpha=0,01$ et $\beta=0,5$ pour LS Newton). On fait cela pour $\mu = 2, 15, 50$ et 100.

On obtient quatres graphiques, donc pour chaque méthode log-barrière (donc pas et LS Newton) (un pour le primal et un pour le dual pour chaque méthode).

Remarque: Pour tracer ces courbes, on utilise le fait qu'on a un duality gap égal à $\frac{m}{t}$ à chaque ième itération (et on a bien sûr constant pendant les mêmes itérations).

Dans tous les cas, on observe toujours le fait que quand μ est petit on a

plus d'outer itérations et moins d'inner itérations et inversement quand μ est grand.

Pour la variante damped Newton, on observe que pour les μ "ordinaires" 2 et 100 l'algorithme se termine vers 175 itérations pour le primal et 145 itérations pour le dual alors que pour $\mu = 15$ et 50 on a vers 140-150 itérations pour le primal et vers 110-120 itérations pour le dual. On a donc globalement une performance légèrement meilleure de 30 itérations environ dans la résolution du dual qui peut donc nous donner $d^* = f^*$ par dualité forte un peu plus rapidement.

Pour la variante LS Newton, on observe que les performances sont similaires pour le primal et le dual, vers 60 itérations pour $\mu = 2$ et vers 40 itérations pour $\mu = 15, 50$ et 100. On a donc une meilleure performance de cette variante par rapport à celle de damped Newton. Cela peut paraître cohérent du fait que dans la variante LS on cherche sur la ligne $x + t\Delta x$ un t approximativement optimal (dans le sens de la minimisation).

Remarque: On a adopté la "backtracking linesearch" en s'assurant d'abord que $x + t\Delta x$ est dans le domaine de la fonction à minimiser avant de chercher un t approximativement optimal. Alors que la variante damped Newton permet directement d'être dans le domaine de la fonction à minimiser en prenant $t = \frac{1}{1+\lambda(x)}$ (Newton decrement) dans $x + t\Delta x$ qui agit

comme une pénalisation pour contrer le risque d'être hors domaine lorsque $\lambda(x)$ est trop grand. Cependant, ce choix de $t = \frac{1}{1+\lambda(x)}$ n'est pas forcément optimal dans le sens de la minimisation sur la ligne $x + t\Delta x$. La variante LS semble donc une fois adaptée pour ne pas sortir du domaine plus efficace.

Pour finir, en termes de choix de p, pour la variante LS $p = 2$ nous donne un surplus d'une vingtaine d'itérations par rapport à $p = 15, 50$ et 100. On peut donc alors penser que le choix d'un p entre 10 et 100 environ n'aura pas un impact critique sur la performance (et sera mieux qu'un p petit comme $p = 2$).

Par ailleurs, pour la variante damped Newton c'est $p = 15$ et 50 qui semblent être un peu plus performants. On pourrait alors prendre dans ce cas p entre ces deux valeurs (15 et 50) environ.