Travail realisé par : Anass El Moubaraki ; Mohamed Maadili

Méthodes de second ordre pour résoudre les problèmes d'optimisation convexe semi-lisse. Application à la résolution de problèmes d'optimisation combinatoire.

Résumé : *Le travail présenté dans cet article consiste en la résolution de problème d'optimisation convexe semi-lisse (fréquemment utilisé en imagerie et en traitement de signal pour des processus de débruitage, de compression et de réduction de dimension) à partir d'algorithme de second ordre dit proximal-quasi-newton. Nous allons nous baser sur des outils d'algèbre linéaire, analyse convexe et analyse hilbertienne pour comprendre la théorie sous-jacente et implémenter en matlab l'algorithme.*

Mots clefs : *optimisation convexe semi lisse , algorithme proximal , Quasi-Newton , dualité de Fenchel , optimisation combinatoire*

Projet encadré par :

Mr Sebastien Bourguignon
Enseignant-Chercheur (LS2N - CNRS/École Centrale de Nantes/Nantes Université).

Mr Gwenael Samain
Doctorant (LS2N - CNRS/École Centrale de Nantes/Nantes Université).

Table des matières

1 Introduction

Dans ce projet, nous désirons confronter des algorithmes d'optimisation de première ordre aux algorithmes de second ordre dans le contexte de la résolution des problèmes d'optimisation combinatoire (pénalisation l0). Pour clarifier nos propos, nous considérons le problème suivant (1)

$$x^* \in \underset{x \in \mathbb{R}^N}{\text{Argmin}} \frac{1}{2}||y - \mathbf{A}x||_2^2 + \lambda||x||_0 \ \ (\text{s.c. } ||x||_\infty \leq B) \tag{1}$$

La résolution du problème ci-dessus peut se faire en utilisant un algorithme de séparation et évaluation dit algorithme de Branch-and bound en anglais.

Petite description de l'algorithme Branch-and-Bound : [4]

L'algorithme se base principalement sur le parcours judicieux d'un ensemble de solutions réalisables en alternant deux étapes :

- la séparation qui consiste à diviser selon une règle de décision un problème complexe que nous assimilons à un noeud parent en sous problèmes que nous assimilons à des noeuds fils tout en constituant un arbre de recherche
- l'évaluation qui consiste à déterminer les caractéristiques d'un noeud pour conclure quant à son élagage et par conséquent réduire l'espace de recherche de la solution optimale.

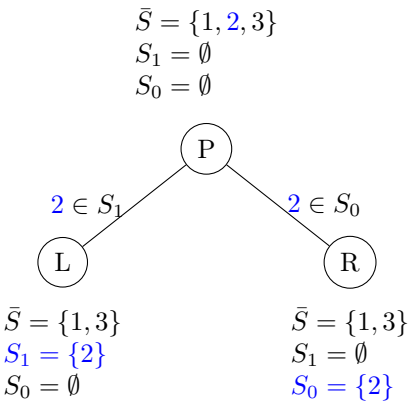
La division d'un problème en sous problèmes se fait en imposant des contraintes de nullité sur les coordonnées du vecteurs de pentes x.
Considérons $\overline{S}$, S_0, S_1 tq :

$$S_1 = \{t \in [\![1, N]\!] \ tq \ x_t \in \mathbb{R}^*\}$$

$$S_0 = \{t \in [\![1, N]\!] \ tq \ x_t = 0\}$$

$$\overline{S} = \overline{(S_1 \cup S_0)} \cap [\![1, N]\!]$$

Exemple :



Définition du sous problème : Considérons un noeud caractérisé par une configuration $\overline{S}$, S_0, S_1 nous définissons un sous problème par :

$$\begin{cases} \min\limits_{(x_{S_1\cup\overline{S}})\in\times\mathbb{R}^{|S_1|+|\overline{S}|}} \frac{1}{2}||y-\mathbf{A}_{S_1\cup\overline{S}}x_{S_1\cup\overline{S}}|| + \lambda||x_{\overline{S}}||_0 + \lambda|S_1| \\ \text{s.c } ||x_{S_1\cup\overline{S}}||_\infty \leq B \end{cases}$$

Définition des caractéristiques d'un sous problème :

La borne inférieure :

$$(LB): \quad \min_{(x_{S_1\cup\overline{S}})\in\times\mathbb{R}^{|S_1|+|\overline{S}|}} \frac{1}{2}||y-\mathbf{A}_{S_1\cup\overline{S}}x_{S_1\cup\overline{S}}||+\frac{\lambda}{B}||x_{\overline{S}}||_1+\iota_{[-B,B]^{|S_1|+|\overline{S}|}}(x_{S_1\cup\overline{S}})+\lambda|S_1| \tag{2}$$

avec :

$$\iota_{\mathcal{B}}(x) = \begin{cases} 0 \; si \; x \in \mathcal{B} \\ +\infty \; sinon \end{cases}$$

La borne supérieure :

$$(UB): \quad \min_{(x_{S_1})\in\times\mathbb{R}^{|S_1|}} \frac{1}{2}||y-\mathbf{A}_{S_1}x_{S_1}|| + \iota_{[-B,B]^{|S_1|}}(x_{S_1}) + \lambda|S_1| \tag{3}$$

Le calcul de la borne inférieure et de la borne supérieure se fait par résolution de problèmes d'optimisation convexe semi-lisse. A cette étape de l'algorithme Branch-and-Bound nous pouvons utiliser des méthodes de résolution de premier ordre (ISTA, Forward-backward,...) ou des méthodes de second ordre a priori plus rapides, objet de notre projet d'option. En effet, dans la littérature, les méthodes de second ordre n'ont donné lieu qu'à quelques travaux, ces dernières étant plus coûteuses sur des problèmes de grande taille. Ici le cadre applicatif (pour l'optimisation l0 par Branch and Bound) est plutôt celui de problèmes de petite taille, d'où l'intérêt de l'étude. Nous nous contenterons de présenter un schéma de résolution du problème (2), ce dernier étant plus général que le problème (3).

2 Formulation mathématique de notre problème

2.1 Problème d'optimisation à résoudre

Soit $(p,n) \in \mathbb{N}^* \times \mathbb{N}^*$, $\mathbf{B} \in \mathcal{M}_{n,p}(\mathbb{R})$, $k_1 \in [\![1,p]\!]$ et $(\lambda, B) \in \mathbb{R}^+ \times \mathbb{R}^{+*}$.

Nous cherchons :

$$z^* \in \underset{z\in\mathbb{R}^p}{\text{Argmin}} \frac{1}{2}||y - \mathbf{B}z||_2^2 + \mu(\sum_{i=1}^{k_1} |z_i|) + \iota_{\mathcal{B}}(z) \tag{4}$$

avec :
$z = x_{\overline{S}\cup S_1}$, $p = |S_1| + |\overline{S}|$, $\mathbf{B} = \mathbf{A}_{\overline{S}\cup S_1}$ et $k_1 = |\overline{S}|$, $\mu = \frac{\lambda}{B}$, $\mathcal{B} = [-B, B]^p$.

Nous posons dans ce qui suit :

$f : z \to \frac{1}{2}||y - \mathbf{B}z||_2^2$ est une fonction convexe de classe C^2 sur $\mathbb{R}^p$.

$g : z \to \mu(\sum_{i=1}^{k_1} |z_i|) + \iota_{\mathcal{B}}(z)$ est une fonction convexe semi-continue inférieurement non-différentiable sur $\mathbb{R}^p$.

2.2 Existence d'une solution à notre problème

Proposition 1.

$$(\exists z^* \in \mathbb{R}^p)\ tq\ z^* \in \underset{z\in\mathbb{R}^p}{\text{Argmin}}\ f(z) + g(z) \tag{5}$$

Démonstration. Remarquons dans un premier temps que :

$$(\forall z \in \mathbb{R}^p) :\ f(z) + g(z) \geq 0$$

Ainsi : $\exists a \in \mathbb{R}^+\ tq\ a = \inf_{z\in\mathbb{R}^p} f(z) + g(z)$

Considérons alors une suite $(z_k)_k$ tel que $(f(z_k) + g(z_k))_k$ converge vers a. Une telle suite existe par caractérisation séquentielle de l'infimum.

Montrons que $(z_k)_k$ est bornée :

Nous avons $(f(z_k) + g(z_k))_k$ qui converge vers a donc

$$(\forall \epsilon \in \mathbb{R}^+)(\exists N_\epsilon \in \mathbb{N})\ tq\ k \geq N_\epsilon \Rightarrow |f(z_k) + g(z_k) - a| < \epsilon \tag{6}$$

Fixons alors un ϵ et N_ϵ alors

$$k \geq N_\epsilon \Rightarrow z_k \in \mathcal{B} \tag{7}$$

En effet si $k \geq N_\epsilon$ et $z_k \in \mathcal{B}^c$ alors $g(z_k) = \infty$ ce qui contredit (6)
L'implication (7) nous permet de conclure que $(z_k)_k$ est bornée.

En appliquant le théorème de Bolzano-Weirstrass, nous pouvons extraire $(z_{\phi(k)})_k$ qui converge vers une valeur z^* et $z^* \in \mathcal{B}$ ($\mathcal{B}$ étant ferme borné)

Montons dorénavant que $f(z^*) + g(z^*) = a$

$f + g$ est continue sur son domaine admissible $\mathcal{B}$. Par suite, $(f(z_{\phi(k)}) + g(z_{\phi(k)}))_k$ converge vers $f(z^*) + g(z^*)$. De plus, $(f(z_{\phi(k)}) + g(z_{\phi(k)}))_k$ est une sous suite de $(f(z_k) + g(z_k))_k$, elle converge donc vers a.

Par unicité de la limite dans $\mathbb{R}$ (espace topologique séparé pour la topologie induite par la norme euclidienne) : $f(z^*) + g(z^*) = a$

□

2.3 Démarche algorithmique

Nous voulons nous inspirer de la méthode quasi-Newton classique pour la résolution de ce problème d'optimisation. Nous désirons alors construire un algorithme itératif qui se base sur l'équation suivante à la $k - eme$ itération :

$$z_{k+1} = \text{prox}_g^{\mathbf{B}_k}(z_k - \mathbf{B}_k^{-1}\nabla f(z_k)) \tag{8}$$

avec :

—

$$(\forall x \in \mathbb{R}^p)(\forall \mathbf{M} \in \mathcal{S}_{++}(p)) \;\; \text{prox}_g^{\mathbf{M}}(x) = \underset{w \in \mathbb{R}^p}{argmin} \;\; g(w) + \frac{1}{2}\langle x - z, \mathbf{M}(x - w)\rangle$$

— $\mathbf{B}_k$ est une approximation de la hessienne de f qui s'écrit sous la forme d'une matrice diagonale et d'une matrice de rang 1 et qui vérifie l'équation de la sécante (9) (voir algorithme 1 pour plus de détails) :

$$\mathbf{B}_k(z_k - z_{k-1}) = \nabla f(z_k) - \nabla f(z_{k-1}) \tag{9}$$

2.3.1 Calcul de l'approximation de la hessienne

Nous nous sommes basés sur la bibliographie [2] et [3] pour acquérir l'approche théorique qui nous permettra d'expliciter le calcul de l'équation (8). En effet, dans un premier temps, nous avons utilisé l'algorithme 1 pour calculer une approximation de la hessienne $(\mathbf{B}_k)$ sous la forme d'une diagonale plus matrice de rang 1 et ce à chaque itération k.

Algorithm 1 Pseudocode pour le calcul de $\mathbf{B}_k^{-1} = \mathbf{H}_k$ [2]

Entrées: k, z_k, z_{k-1}, $\nabla f(z_k)$, $\nabla f(z_{k-1})$, $\tau_0 > 0$, $0 < \gamma < 1$, $0 < \tau_{min} < \tau_{max}$

$y_k \leftarrow \nabla f(z_k) - \nabla f(z_{k-1})$
$s_k \leftarrow z_k - z_{k-1}$
if $k = 0$ **then**
  $\mathbf{H}_0 \leftarrow \tau_0 \mathbf{I}$ $\{\mathbf{B}_0 = \mathbf{H}_0^{-1}\}$
end if
$\tau_k \leftarrow \frac{\langle s_k, y_k \rangle}{||y_k||^2}$
Projection de τ_k sur $[\tau_{min}, \tau_{max}]$
$\mathbf{H}_0 \leftarrow \gamma \tau_k \mathbf{I}$
if $\langle s_k - \mathbf{H}_0 y_k, y_k \rangle < 10^{-8} ||y_k||_2 ||s_k - \mathbf{H}_0 y_k||_2$ **then**
  $u_k \leftarrow 0$
else
  $u_k \leftarrow \frac{1}{\sqrt{\langle s_k - \mathbf{H}_0 y_k, y_k \rangle}} (s_k - \mathbf{H}_0 y_k)$
end if
$\mathbf{H}_k \leftarrow \mathbf{H}_0 + u_k u_k^T$
$\mathbf{B}_k \leftarrow \frac{1}{\gamma \tau_k} \mathbf{I} - \frac{1}{\gamma \tau_k (||u_k||^2 + \gamma \tau_k)} u_k u_k^T$ { Application du lemme d'inversion de Sherman-Morrison }

Sorties: $\mathbf{H}_k$, $\mathbf{B}_k$

2.3.2 Formulation explicite de l'opérateur proximal induit par l'approximation de la hessienne

Dans cette partie, nous allons explicité le calcul de l'opérateur proximal induit par l'approximation de la hessienne (calculée ci-dessus). En effet, une fois $\mathbf{B}_k$ calculé , nous pouvons simplifier (8) à partir des deux théorèmes 1 et 2

Théorème 1. *[3] Soient g une fonction propre, convexe et semi-continue inferieurement définie sur un espace de hilbert $(\mathcal{H}, \langle , \rangle)$ de dimension finie égale à p et $\boldsymbol{V} \in \mathcal{S}_{++}(p)$. Nous supposons que $\boldsymbol{V} = \boldsymbol{P} \pm \boldsymbol{Q}$ où $\boldsymbol{P} \in \mathcal{S}_{++}(p)$ et $\boldsymbol{Q} = \sum_{i=1}^{r} u_i u_i^T$ est une matrice de rang r. Nous considérons $\boldsymbol{U} = (u_1, u_2, ..., u_r)$ alors :*

$$(\forall x \in \mathbb{R}^p) : \ \text{prox}_g^V(x) = \text{prox}_g^P(x \mp \boldsymbol{P}^{-1} \boldsymbol{U} \alpha^*) \tag{10}$$

où

$$\alpha^* \textit{ racine de } \mathcal{L}(\alpha) = \boldsymbol{U}^T (x - \text{prox}_h^{\boldsymbol{P}}(x \mp \boldsymbol{P}^{-1} \boldsymbol{U} \alpha)) + \alpha$$

$\mathcal{L} : \mathbb{R}^r \to \mathbb{R}^r$ est fortement monotone, continue lipschitzienne de constante de Lipschitz $1 + |||\boldsymbol{P}^{-\frac{1}{2}} \boldsymbol{U}|||^2$.

Corollaire :

Dans notre cas :

— $\mathbf{V} = \mathbf{B}_k$
— $\mathbf{P} = \frac{1}{\gamma \tau_k} \mathbf{I}$
— $\mathbf{Q} = \frac{1}{\gamma \tau_k (||u_k||^2 + \gamma \tau_k)} u_k u_k^T = v_k v_k^T \quad (v_k = \frac{1}{\sqrt{\gamma \tau_k (||u_k||^2 + \gamma \tau_k)}} u_k)$

— $\mathbf{U} = v_k$

Ainsi :

$$\text{prox}_g^{\mathbf{B}_k}(x) = \text{prox}_g^{\frac{1}{\gamma\tau_k}\mathbf{I}}(x + \gamma\tau_k v_k \alpha^*) \tag{11}$$

$\alpha^* \in \mathbb{R}$ est une racine de :

$$\mathcal{L} : \alpha \to v_k^T(x - \text{prox}_g^{\frac{1}{\gamma\tau_k}\mathbf{I}}(x + \gamma\tau_k v_k \alpha)) + \alpha$$

Théorème 2. *[2] Soient $g = \sum_{i=1}^{p} g_i$ une fonction séparable propre, convexe et semi-continue inférieurement définie sur un espace de hilbert $(\mathcal{H}, \langle , \rangle)$ de dimension finie et $\boldsymbol{D}$ une matrice diagonale définie positive. Nous écrivons $\boldsymbol{D} = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_p)$ alors :*

$$(\forall x \in \mathbb{R}^p) : \ \text{prox}_g^{\boldsymbol{D}}(x) = \left(\text{prox}_{\frac{g_i}{\lambda_i}}(x_i)\right)_{1 \leq i \leq p} \tag{12}$$

Corollaire :

Rappelons que notre fonction $g : z \to \mu(\sum_{i=1}^{k_1} |z_i|) + \iota_{\mathcal{B}}(z)$ est séparable.

Dans notre cas :

— $\mathbf{D} = \frac{1}{\gamma\tau_k}\mathbf{I}$

Par suite :

$$\text{prox}_g^{\frac{1}{\gamma\tau_k}\mathbf{I}}(x+\gamma\tau_k v_k\alpha^*) = \text{prox}_{\gamma\tau_k g}(x+\gamma\tau_k v_k\alpha^*) = \left(\text{prox}_{\gamma\tau_k g_i}((x+\gamma\tau_k v_k\alpha^*)_i)\right)_{1 \leq i \leq p} \tag{13}$$

avec :

$$g_i = \begin{cases} \iota_{[-B,B]} + \frac{\lambda}{B}|.| \ si \ i \leq k_1 \\ \iota_{[-B,B]} \ sinon \end{cases}$$

L'expression de $\text{prox}_{\gamma\tau_k g}$ est bien connu compte tenu de la separabilite de g (voir équations (14) et (15)).

Soit $r > 0$ et $w \in \mathbb{R}$ et $\beta = \frac{r\lambda}{B}$ alors :

$$\text{prox}_{r(\iota_{[-B,B]} + \frac{\lambda}{B}|.|)}(w) = \begin{cases} B \ si \ w \geq \beta + B \\ w - \beta \ si \ \beta \leq w \leq \beta + B \\ -B \ si \ w \leq -\beta - B \\ w + \beta \ si \ -\beta - B \leq w \leq -\beta \\ 0 \ si \ |w| \leq \beta \end{cases} \tag{14}$$

$$\text{prox}_{r\iota_{[-B,B]}}(w) = \begin{cases} -B \ si \ w < -B \\ B \ si \ w > B \\ w \ si \ -B \leq w \leq B \end{cases} \tag{15}$$

Remarque 1. *L'évaluation de* $\text{prox}_g^{\boldsymbol{B}_k}(x)$ *repose alors sur le calcul de* α^**,le problème devient "simple" car il se ramène à la recherche itérative de racines d'une fonction de variable réelle qui est linéaire par morceaux (*$\mathcal{L}$*) fortement monotone de* $\mathbb{R}$ *vers* $\mathbb{R}$ *donc strictement croissante . Cela dit, en triant les points de changement de dynamique de cette dernière on pourra connaître le segment qui intersecte l'axe des abscisses et par suite la racine de la fonction qui est unique compte tenu de la monotonie forte.*

Proposition 2. *[3] Soit* $x \in \mathbb{R}^p$*,* g *une fonction séparable sur* $\mathbb{R}^p$ *et* $\boldsymbol{D} = diag((d_i)_{1\leq i\leq p})$*. Nous supposons que :*

$$(\forall i \in [\![1,p]\!]) : \text{prox}_{\frac{g_i}{d_i}}(x_i) = a_{i,j}x_i + b_{i,j} \text{ si } t_{i,j} < x_i < t_{i,j+1} \ , \ j \in [\![1,c_i]\!] \text{ avec } (t_{i,1} = -\infty \ t_{i,c_i} = +\infty)$$

et nous cherchons la racine α^* *de :*

$$\mathcal{L} : \alpha \to u^T(x - \left(\text{prox}_{\frac{g}{d_i}}(x_i + \alpha\frac{u_i}{d_i})\right)_{1\leq i\leq p}) + \alpha$$

alors nous pouvons déterminer α^* *en triant les points* $\left(-\frac{d_i}{u_i}(x_i - t_{i,j})\right)_{(i,j)\in[\![1,p]\!]\times\{1,..,c_i\}}$ *parmi lesquels se trouvent les points de changement de dynamique de* $\mathcal{L}$*.*

Nous pouvons adapter ce résultat à notre problème. En effet :
- $\mathbf{D} = \frac{1}{\gamma\tau_k}\mathbf{I}$.
- $\forall i \ d_i = \frac{1}{\gamma\tau_k}$
- $u = v_k$
- $i \leq k_1 \Rightarrow (c_i = 6 \ , \ t_{i,j} \in \{-\infty, \beta_k + B, -B - \beta_k, \beta_k, -\beta_k \ + \infty\} \ , \ a_{i,j} \in \{1,0\}$ et $b_{i,j} \in \{0, B, -B, \beta, -\beta\})$
- $i > k_1 \Rightarrow (c_i = 4 \ , \ t_{i,j} \in \{+\infty, B, -B, -\infty\} \ , \ a_{i,j} \in \{1,0\}$ et $b_{i,j} \in \{0, B, -B\})$

avec $\beta_k = \frac{\gamma\tau_k\lambda}{B}$

Nous présentons dans 2 un pseudo-code pour déterminer α^* de façon exacte.

Algorithm 2 Pseudo-code pour la détermination de α^*

Entrées: Liste$=\bigcup_{i=1}^{p}\{-\frac{d_i}{u_i}(x_i - t_{i,j}) \ : \ j = 1,..,c_i\}$
Liste $\leftarrow$ Tri(Liste) {Nous trions notre liste}
$\theta^- \leftarrow max\{e \in Liste \ t.q \ \mathcal{L}(e) \leq 0\}$
$\theta^+ \leftarrow min\{e \in Liste \ t.q \ \mathcal{L}(e) \geq 0\}$ {Parcours itératif ou méthode de la bissectrice}
$a \leftarrow \frac{\mathcal{L}(\theta^+) - \mathcal{L}(\theta^-)}{\theta^+ - \theta^-}$ {Si $\theta^+ = +\infty$ ou $\theta^- = -\infty$, nous prenons $\theta_1 \in \mathbb{R}$ et $\theta_2 \in \mathbb{R}$ tq $\theta^- \leq \theta_1 \leq \theta_2 \leq \theta^+$}
$b \leftarrow \mathcal{L}(\theta^+) - a \times \theta^+$
$\alpha^* \leftarrow -\frac{b}{a}$

2.3.3 Définition d'un critère d'arrêt pertinent

Nous définissons comme critère d'arrêt pertinent le saut de dualité entre notre fonction objectif et son dual. Pour ce faire, nous utilisons les résultats sur la dualité de Fenchel démontrés dans [1]. En effet, nous cherchons

$$z^* \in \underset{z\in\mathbb{R}^p}{\text{Argmin}} \frac{1}{2}||y - \mathbf{B}z||_2^2 + \mu(\sum_{i=1}^{k_1}|z_i|) + \iota_{\mathcal{B}}(z)$$

Notre problème s'écrit alors comme :

$$z^* \in \underset{z\in\mathbb{R}^p}{\text{Argmin}}\ F(\mathbf{B}z) + G(z) \tag{16}$$

Proposition 3.

$$\min_{z\in\mathbb{R}^p}\max_{w\in\mathbb{R}^n} \langle \boldsymbol{B}z, w\rangle + G(z) - F^*(w) = \min_{z\in\mathbb{R}^p} F(\boldsymbol{B}z) + G(z) = \max_{w\in\mathbb{R}^n} -F^*(w) - G^*(-\boldsymbol{B}^T w)$$

où, F^* désigne la fonction conjuguée de F. Elle vérifie :

$$(\forall w \in \mathbb{R}^n)\ F^*(w) = \max_{v\in\mathbb{R}^n} \langle v, w\rangle - F(v)$$

Démonstration. La preuve repose sur l'identité de Fenchel-Moreau [1]

En effet, si F est une fonction propre, convexe et semi-continue inférieurement alors :

$$F^{**} = F$$

Ainsi :

$$\forall w \in \mathbb{R}^n :\ F(w) = F^{**}(w) = \max_{v\in\mathbb{R}^n} \langle v, w\rangle - F^*(z)$$

Par conséquent :

$$\begin{aligned}
\min_{z\in\mathbb{R}^p}\max_{w\in\mathbb{R}^n} \langle \mathbf{B}z, w\rangle + G(z) - F^*(w) &= \min_{z\in\mathbb{R}^p} G(z) + \max_{z\in\mathbb{R}^n} \langle \mathbf{B}z, w\rangle - F^*(w) \\
&= \min_{z\in\mathbb{R}^p} G(z) + F^{**}(\mathbf{B}z) \\
&= \min_{z\in\mathbb{R}^p} G(z) + F(\mathbf{B}z)
\end{aligned}$$

D'autre part :

$$\begin{aligned}
\min_{z\in\mathbb{R}^p}\max_{w\in\mathbb{R}^n} \langle \mathbf{B}z, w\rangle + G(z) - F^*(w) &= \max_{w\in\mathbb{R}^n} -F^*(w) + \min_{z\in\mathbb{R}^p} \langle \mathbf{B}z, w\rangle + G(z) \\
&= \max_{w\in\mathbb{R}^n} -F^*(w) - \max_{z\in\mathbb{R}^p} -\langle z, \mathbf{B}^T w\rangle - G(z) \\
&= \max_{w\in\mathbb{R}^n} -F^*(w) - G^*(-\mathbf{B}^T w)
\end{aligned}$$

□

En pratique, nous calculons à chaque itération k, l'expression du dual $-F^*(w_k) - G^*(-\mathbf{B}^T w_k)$ et celle de la fonction objectif $G(z_k) + F(\mathbf{B}z_k)$ mais aussi le saut de dualité définie comme étant la différence des deux. Cette quantité est toujours positive et s'annule si convergence vers le point selle du primal-dual (z^*, w^*). En effet, la propriété de dualité faible de Fenchel stipule que :

$$\min_{z\in\mathbb{R}^p} F(\mathbf{B}z) + G(z) \geq \max_{w\in\mathbb{R}^n}\ -F^*(w) - G^*(-\mathbf{B}^T w)$$

Maintenant que nous avons les expressions de l'approximation de la hessienne $\mathbf{B}_k$, celle de l'opérateur proximal induit par $\mathbf{B}_k$ et avons défini un critère d'arrêt pertinent, nous présentons dans 3, le schéma algorithmique final.

Algorithm 3 Algorithme de second ordre [2]

Entrées: ϵ, λ, B, k_1, z_0,i_{max}
  On calcule objectif(0) et dual(0)
  $i = 0$
  saut dualite(0) $\leftarrow$ objectif(0) $-$ $dual(0)$
  while $i \leq i_{max}$ and saut dualite(i) $\geq \epsilon$ **do**
    $z_{i+1} \leftarrow \text{prox}_g^{\mathbf{B}_i}(z_i - \mathbf{B}_i^{-1}\nabla f(z_i))$
    $i \leftarrow i + 1$
    On calcule objectif(i) et dual(i)
    saut dualite(i) $\leftarrow$ objectif(i) $-$ dual(i)
  end while
Sorties: i, z_i, objectif(i)

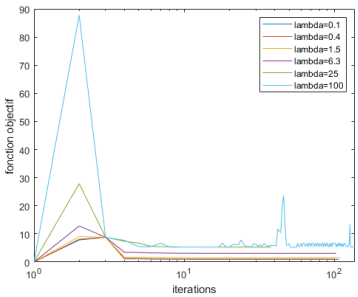
Remarque 2. *Nous pouvons aussi améliorer notre algorithme en ajoutant une recherche linéaire selon une direction de descente.. Cela revient à combiner notre algorithme de second ordre avec une descente de gradient à pas variable ou l'on doit vérifier les conditions de Wolfe.*

3 Résultats numériques

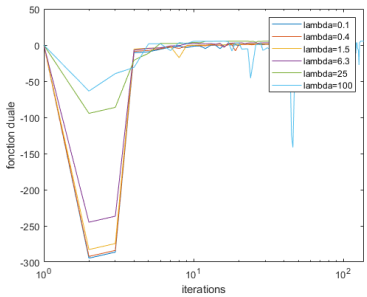
3.1 Vérification de la convergence

Nous commençons par vérifier la convergence de notre notre algorithme de second ordre. Pour se faire, nous synthétisons un vecteur y, un dictionnaire $\mathbf{B}$ en prenant comme paramètres $(n, p, k_1) = (100, 10, 8)$ avec une corrélation statistique entre les colonnes qui vaut $\rho = 0.5$. Nous fixons la valeur du seuil de convergence à $\epsilon = 10^{-8}$ et choisissons $B = 10$. Enfin, nous explorons une grille logarithmique de valeurs de λ contenant six valeurs distribuées entre 0.1 et 100.

Dans un premier temps, nous traçons dans les figures 1a et 1b l'évolution de la fonction objectif et du dual en fonction du nombre d'itérations pour différentes valeurs du paramètre de régularisation λ.



(a) Évolution de la fonction objectif en fonction du nombre d'itérations



(b) Évolution du duale en fonction du nombre d'itérations

Dans un deuxième temps, nous traçons l'évolution de l'écart de dualité en fonction des itérations dans la figure 2.

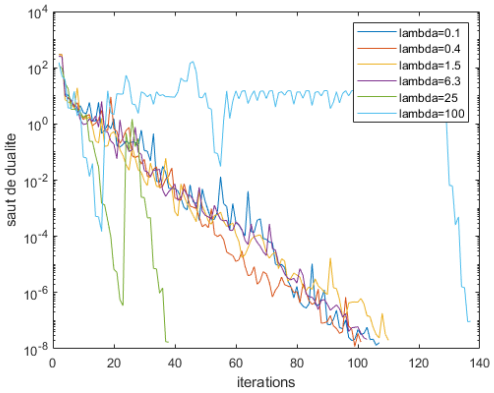


FIGURE 2 – Évolution du saut de dualité en fonction du nombre d'itérations

Commentaire :

Nous pouvons déduire que l'algorithme converge bien vers un minimum de la fonction objectif et un maximum du dual et le saut de dualité vérifie la propriété de Fenchel . En se basant sur ces deux métriques, nous pouvons déduire que l'algorithme est fonctionnel dans le sens où il converge bien vers la bonne solution. Vérifions alors le comportement des minimiseurs en fonction du paramètre de régularisation. Pour se faire, nous tracons dans la figure 3 l'allure de z^* pour différentes configurations de λ.

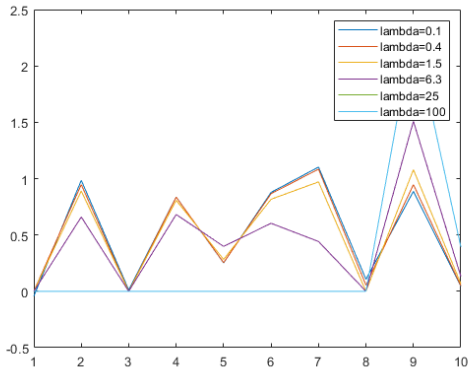


FIGURE 3 – Allure des estimées pour différentes valeurs de λ

Commentaire :

L'analyse de la figure 3 met en évidence une évolution de la parcimonie sur z^*. En effet, plus λ est grand, plus $||z^*||_0$ diminue. En examinant les deux dernières valeurs de λ, nous observons que nous avons atteint un degré de parcimonie maximal ($\sum_{i=1}^{k_1} |z_i^*| = 0$).

3.2 Comparaison Ista, Icd , Prox-quasi-newton

Dans cette partie nous désirons comparer les coûts temporels et le nombre d'itérations des algorithmes ICD et ISTA à celui de notre algorithme proximal quasi Newton.

Pour se faire, nous choisissons trois paramètres à varier :

— Le coefficient de régularisation λ.

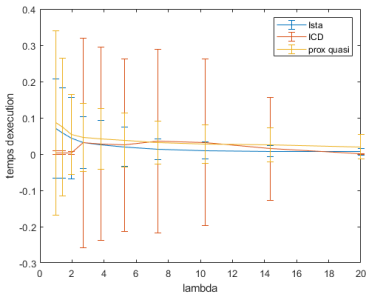
— La corrélation entre les colonnes du dictionnaire ρ.

— Le nombre de colonnes p.

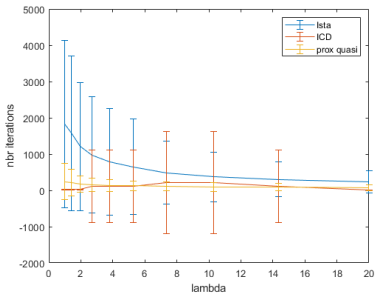
Pour chaque paramètre nous explorons une grille de 10 valeurs et stockons les valeurs du temps de calcul et du nombre d'itérations pour chaque algorithme dans un tenseur de taille $10 \times 10 \times 10 \times 3 \times 2$

3.2.1 Évolution du coût temporel et nombres d'itérations en fonction du coefficient de régularisation λ

Nous traçons dans les figures 4a et 4bl'évolution du nombre d'itérations et du temps de calcul en fonction du paramètre de régularisation. Pour cela , nous moyennons sur toutes les valeurs de temps de calcul ou nombre d'itérations associés à une certaine valeur de λ. Dans les courbes, nous faisons figurer les écarts types sous forme de barre d'erreur vertical.



(a) Évolution du temps de calcul en fonction de λ



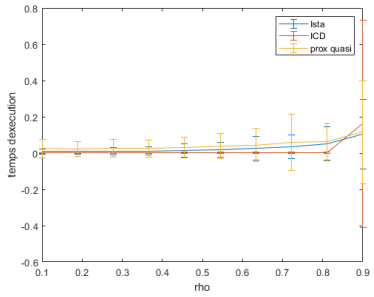
(b) Évolution du nombre d'itérations en fonction de λ

Commentaire :

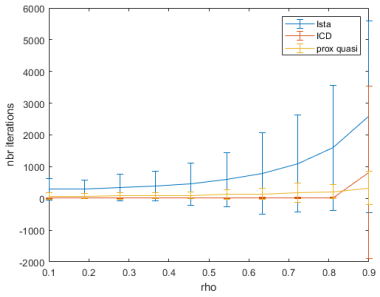
Nous pouvons remarquer que l'algorithme de second ordre (prox-quasi) présente un bon compromis nombre d'itérations/coût temporel.

3.2.2 Évolution du coût temporel et du nombre d'itérations en fonction du degré de corrélation statistique entre les colonnes du dictionnaire

Nous traçons dans les figures 5a et 5b l'évolution du nombre d'itérations et du temps de calcul en fonction du degré de corrélation. Pour cela , nous moyennons sur toutes les valeurs de temps de calcul ou nombre d'itérations associés à un certain ρ. Dans les courbes, nous faisons figurer les écarts types sous forme de barre d'erreur vertical.



(a) Évolution du temps de calcul en fonction de ρ



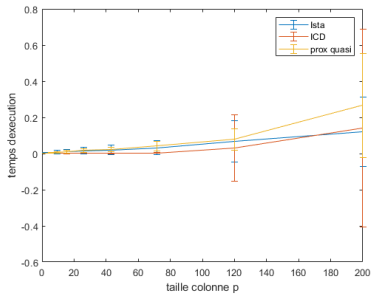
(b) Évolution du nombre d'itérations en fonction de ρ

Commentaire :

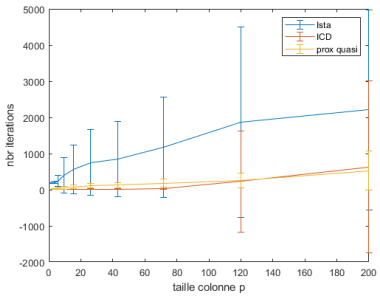
Nous pouvons remarquer que notre algorithme de second ordre est très performant en terme de nombre d'itérations et de coût de calcul pour des dictionnaires dont la corrélation entre les colonnes est supérieure à 0.9.

3.2.3 Évolution du coût temporel et du nombre d'itérations en fonction de la dimension p (nombre de colonnes du dictionnaire)

Nous traçons dans les figures 6a et 6b l'évolution du nombre d'itérations et du temps de calcul en fonction du nombre de colonnes du dictionnaire. Pour cela , nous moyennons sur toutes les valeurs de temps de calcul ou nombre d'itérations associés à un certaine valeur de p



(a) Évolution du temps de calcul en fonction de p



(b) Évolution du nombre d'itérations en fonction de p

Commentaire :

Nous pouvons remarquer que l'algorithme de second ordre est plus efficace pour des dictionnaires avec un nombre de colonnes inférieur aux nombre de lignes. Nous pouvons aussi remarquer que le temps de calcul et nombre d'itérations ne varient pas beaucoup pour certaine valeur de p (écart type relativement faible par rapport a l'algorithme Ista et Icd)

4 Pistes d'amelioration de l'algorithme

Nous proposons dans cette partie une preuve de convergence sous une hypothèse forte sur $\mathbf{B}_k$. En effet, en écrivant $(\forall k \in \mathbb{N}) : \mathbf{B}_k = d_k\mathbf{I} - v_k v_k^T$ nous supposons que :

$$(\exists \epsilon > 0)(\forall k \in \mathbb{N}) : d_k \geq ||v_k||^2 + d_{k+1} + \epsilon \tag{17}$$

Proposition 4. *Nous considérons le problème défini en* (4).
Soient $\epsilon > 0$, $d > 0$ *et* $(z_k)_k$ *la suite définie par récurrence comme suit :*

$$z_{k+1} = \text{prox}_g^{\boldsymbol{B}_k}(z_k - \boldsymbol{B}_k^{-1}\nabla f(z_k))$$

On suppose : $(\exists \epsilon > 0)(\forall k \in \mathbb{N}) : d_k \geq ||v_k||^2 + d_{k+1} + \epsilon$

alors : $(z_k)_k$ *converge vers un minimiseur* z^* *et la convergence est monotone.*

Démonstration. Nous commençons par rappeler quelques propriétés utiles [1]

Soient f et g deux fonctions convexes semi-continues inferieurement définies sur un espace $\mathcal{X}$ tel que $dom(f) \cap dom(g) \neq \emptyset$ alors :

$$(\forall x \in \mathcal{X}) : \partial(f+g)(x) = \partial f(x) + \partial g(x)$$

De plus si f est convexe de classe C^1 :

$$(\forall x \in \mathcal{X}) : \partial f(x) = \{\nabla f(x)\}$$

Nous posons dans ce qui suit :

$$F = f + g$$

$$\begin{aligned}
z_{k+1} = \text{prox}_g^{\mathbf{B}_k}(z_k - \mathbf{B}_k^{-1}\nabla f(z_k)) &\Leftrightarrow z_{k+1} = \underset{z\in\mathbb{R}^p}{argmin}\ f(z_k) + \langle z - z_k, \nabla f(z_k)\rangle + \frac{1}{2}||z - z_k||^2_{\mathbf{B}_k} + g(z) \\
&\Leftrightarrow 0 \in \partial(\langle . - z_k, \nabla f(z_k)\rangle + \frac{1}{2}||. - z_k||^2_{\mathbf{B}_k} + g)(z_{k+1}) \\
&\Leftrightarrow 0 \in \{\nabla f(z_k)\} + \{\mathbf{B}_k(z_{k+1} - z_k)\} + \partial g(z_{k+1}) \\
&\Leftrightarrow -\nabla f(z_k) + \mathbf{B}_k(z_k - z_{k+1}) \in \partial g(z_{k+1}) \\
&\Leftrightarrow \nabla f(z_{k+1}) - \nabla f(z_k) + \mathbf{B}_k(z_k - z_{k+1}) \in \partial g(z_{k+1}) + \{\nabla f(z_{k+1})\} \\
&\Leftrightarrow \nabla f(z_{k+1}) - \nabla f(z_k) + \mathbf{B}_k(z_k - z_{k+1}) \in \partial(f+g)(z_{k+1}) \\
&\Leftrightarrow \forall z :\ F(z) \geq F(z_{k+1}) + \langle z - z_{k+1}, \nabla f(z_{k+1}) - \nabla f(z_k) + \mathbf{B}_k(z_k - z_{k+1})\rangle
\end{aligned}$$

En particulier nous avons :

$$F(z_k) \geq F(z_{k+1}) + \langle z_k - z_{k+1}, \nabla f(z_{k+1}) - \nabla f(z_k) + \mathbf{B}_k(z_k - z_{k+1})\rangle \tag{18}$$

$$\begin{aligned}
18 &\Leftrightarrow F(z_k) \geq F(z_{k+1}) + \langle z_k - z_{k+1}, \nabla f(z_{k+1}) - \nabla f(z_k) - v_k v_k^T(z_k - z_{k+1})\rangle + d_k||z_k - z_{k+1}||^2 \\
&\Leftrightarrow F(z_k) \geq F(z_{k+1}) + \langle z_k - z_{k+1}, \nabla f(z_{k+1}) - \nabla f(z_k)\rangle + (d_k - ||v_k||^2)||z_k - z_{k+1}||^2 \\
&\Leftrightarrow F(z_k) \geq F(z_{k+1}) - \langle z_{k+1} - z_k, \mathbf{B}_{k+1}(z_{k+1} - z_k)\rangle + (d_k - ||v_k||^2)||z_k - z_{k+1}||^2 \\
&\Leftrightarrow F(z_k) \geq F(z_{k+1}) + (d_k - ||v_k||^2 - d_{k+1})||z_k - z_{k+1}||^2
\end{aligned}$$

alors :

$$F(z_k) \geq F(z_{k+1}) + (d_k - ||v_k||^2 - d_{k+1})||z_k - z_{k+1}||^2 \geq \epsilon||z_k - z_{k+1}||^2 \quad (19)$$

En procédant a un télescopage, nous montrons que : $\sum(d_k - ||v_k||^2 - d_{k+1})||z_k - z_{k+1}||^2$ est convergente et par suite son terme général $((d_k - ||v_k||^2 - d_{k+1})||z_k - z_{k+1}||^2)_k$ converge vers 0. Nous montrons aussi que $(||z_k - z_{k+1}||^2)_k$ et $(d_k||z_k - z_{k+1}||^2$ converge vers 0

Remarquons alors qu'il s'en suit que : $(||\nabla f(z_k) - \nabla f(z_{k+1}) + \mathbf{B}_k(z_k - z_{k+1})||)_k$ converge aussi vers 0 puisque

$$\begin{aligned}(\forall z)(\forall k) : ||\nabla f(z_k) - \nabla f(z_{k+1}) + \mathbf{B}_k(z_k - z_{k+1})|| &\leq ||\nabla f(z_k) - \nabla f(z_{k+1})|| + ||\mathbf{B}_k(z_k - z_{k+1})|| \\ &\leq |||\mathbf{B}^T\mathbf{B}|||_2||z_k - z_{k+1}|| + d_k||z_k - z_{k+1}|| \\ &\leq (|||\mathbf{B}^T\mathbf{B}|||_2 + d_0)||z_k - z_{k+1}||\end{aligned}$$

Nous avons :

$$\forall z : \ F(z) \geq F(z_{k+1}) + \langle z - z_{k+1}, \nabla f(z_{k+1}) - \nabla f(z_k) + \mathbf{B}_k(z_k - z_{k+1})\rangle \quad (20)$$

En prenant en compte que $(z_k)_k$ est bornée (contrainte imposée par l'indicatrice) et en passant à la limite nous obtenons : $(F(z_k))_k$ converge vers $\underset{z\in\mathbb{R}^p}{min} F(z)$

Considérons une sous suite $(z_{\phi(k)})_k$ de $(z_k)_k$ qui converge vers une valeur z^*. Une telle suite existe car $(z_k)_k$ est bornée en dimension finie. Montrons alors que $(z_k)_k$ converge vers z^*.

Pour cela, nous reprenons l'équation 19 et nous sommons de n à $\phi(n)$. Cette sommation a un sens car une extraction ϕ vérifie toujours :

$$(\forall n \in \mathbb{N}) : \phi(n) \geq n$$

Nous obtenons alors en appliquant au passage l'inégalité triangulaire :

$$F(z_n) \geq F(z_{\phi(n)}) + \epsilon||z_n - z_{\phi(n)}||^2$$

or $(F(z_n))_n$ converge vers $\underset{z\in\mathbb{R}^p}{min} F(z)$ alors $(F(z_{\phi(n)}))_n$ converge aussi vers la même valeur. On peut donc déduire que $(||z_n - z_{\phi(n)}||)_n$ converge vers 0 et par conséquent $(z_n)_n$ converge vers z^*

□

Avantages :

L'avantage de cette construction est qu'elle permet d'avoir un algorithme monotone qui converge bien vers le minimum sans perte de généralités.En effet, nous pouvons prendre $d_{k+1} = d_k - ||v_k||^2$ et donc créer un algorithme L-BFGS avec une mémoire d'ordre 1. Cela revient à choisir de façon récursive le terme diagonal (τ_k) dans l'approximation de l'inverse de la hessienne afin d'assurer une décroissance forte de la fonction objectif.

5 Bibliographie

Références

[1] Heinz H Bauschke, Patrick L Combettes, Heinz H Bauschke, and Patrick L Combettes. *Correction to : Convex analysis and monotone operator theory in hilbert spaces*. Springer, 2017.

[2] Stephen Becker and Jalal Fadili. A quasi-newton proximal splitting method. *Advances in neural information processing systems*, 25, 2012.

[3] Stephen Becker, Jalal Fadili, and Peter Ochs. On quasi-newton forward-backward splitting : Proximal calculus and convergence. *SIAM Journal on Optimization*, 29(4) :2445–2481, 2019.

[4] Ramzi Ben Mhenni. *Méthodes de programmation en nombres mixtes pour l'optimisation parcimonieuse en traitement du signal*. PhD thesis, École Centrale de Nantes (ECN), 2020.