COMET TSI

Panorama des techniques d'optimisation

Applications pour l'imagerie

Thomas Oberlin

ISAE-SUPAERO, DISC, ANITI, thomas.oberlin@isae-supaero.fr

26 juin 2024

Motivation

Pourquoi l'optimisation?

- Outil central en traitement du signal, en apprentissage et analyse de données
- Outil fondamental pour les problèmes inverses : estimer des paramètres à partir d'observations
- En imagerie : traitement d'image, reconstruction d'images (tomographie, compressed sensing)
- Acquisition et traitements sont de plus en plus intimement liés (computational imaging)

Quelle optimisation?

- Continue : on minimise une fonction à variables réelles
- Avec ou sans contraintes
- Différentiable ou non

Avertissement

- Sélection de méthodes biaisée par mon expérience et mes connaissances de chercheur en traitement d'images et en apprentissage
- Un peu de théorie, beaucoup d'algorithmes, quelques exemples
- Simplifier la théorie $\rightarrow$ certaines hypothèses ne seront pas explicitées, ce n'est pas un cours parfaitement rigoureux mathématiquement
- Peu de citations et de références

Bibliographie

Fondamentaux de l'optimisation

- Jorge Nocedal et Stephen J. Wright. *Numerical optimization*. Springer, 1999.
- Stephen P. Boyd et Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- Dimitri P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.
- Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer, 2004.

Livres plus appliqués ou spécialisés

- Léon Bottou, Frank E. Curtis, et Jorge Nocedal. *Optimization methods for large-scale machine learning.* SIAM review 60.2, 2018.
- Charu C. Aggarwal. *Linear algebra and optimization for machine learning*. Springer International Publishing, 2020.
- Patrick L. Combettes et Jean-Christophe Pesquet. *Proximal splitting methods in signal processing.* Fixed-point algorithms for inverse problems in science and engineering, 2011.

Plan de la séance

Plan de la section

1. Bases mathématiques
   - Notations
   - Dérivées
   - Développements de Taylor
   - Convexité

2. Optimisation différentiable sans contraintes

3. Optimisation sous contraintes

4. Optimisation non lisse et *splitting*

5. Cas d'étude

6. Conclusion

Quelques notations

f	la fonctionnelle à minimiser, $f : \mathbb{R}^n \to \mathbb{R} \cup +\{\infty\}$
x	un vecteur colonne de $\mathbb{R}^n$: $x = (x_1, x_2, \cdots, x_n)^T$
$.^T$	la transposition
$(e_i)_i$	la base canonique de $\mathbb{R}^n$: $e_i = (0, \cdots, 0, 1, 0 \cdots, 0)^T$
A	une matrice, d'éléments A_{ij}, i indique la ligne, j la colonne
I_n	la matrice identité de taille $n \times n$
$\\|x\\|_p$	la norme p de x : $\\|x\\|_p^p = \sum_i \|x_i\|^p$, par défaut $p = 2$
$\\|A\\|_{p \to q}$	la norme subordonnée $\\|A\\|_{p \to q} = \max_{x \neq 0} \frac{\\|Ax\\|_q}{\\|x\\|_p}$ plus grande vp de la matrice
$\\|A\\|_F$	la norme de Frobenius : $\\|A\\|_F^2 = \sum_{i,j} \|A_{ij}\|^2$
$A \succ 0$	A est définie positive
$A \succeq 0$	A est semi-définie positive
∇f	le gradient de f, une fonction de $\mathbb{R}^n$ dans $\mathbb{R}^n$
$\nabla^2 f$	la Hessienne de f, $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$
$\frac{\partial f}{\partial x_i}$	la dérivée partielle de f par rapport à sa i-ème variable

Dérivées partielles et gradient

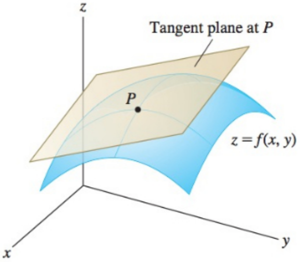
La dérivée partielle est, comme en dimension 1, la limite d'un taux d'acroissement

$$\frac{\partial f}{\partial x_i}(x) = \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon e_i) - f(x)}{\varepsilon} \tag{1}$$

Le gradient est la fonction vectorielle composée des dérivées partielles :

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_i}, \cdots, \frac{\partial f}{\partial x_n} \right) \tag{2}$$

Le gradient définit la différentielle, qui est une approximation linéaire locale (plan tangent) : $f(x+h) \approx f(x) + \nabla f(x)^T h$.



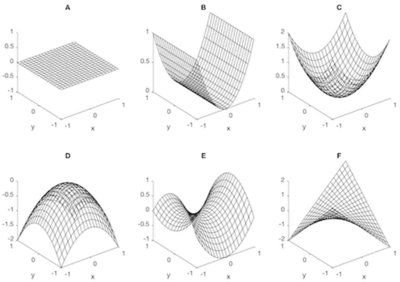
Hessienne

ca permet de caractériser le minimum grace au signe des vp

La Hessienne de f est la matrice

$$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial {x_2}^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial {x_n}^2} \end{pmatrix} \tag{3}$$

La hessienne capture la courbure. En particulier, les signes de ses valeurs propres caractérisent le type de point critique (minimum, maximum, point-selle), et le 1er vecteur propre donne la direction de courbure principale.



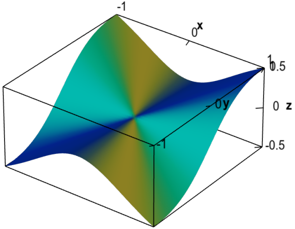
Différentielle et gradient

Développement de f à l'ordre 1 :

$$f(x+h) = f(x) + \nabla f(x)^T h + o(\|h\|), \qquad (4)$$

où $o(\|h\|)$ signifie que ce résidu tend vers 0 "plus vite" que $\|h\|$, quand $\|h\|$ tend vers 0.

Attention, la notion de différentiabilité ("il existe un plan tangent") est plus forte que l'existence de dérivées directionnelles.



Développements d'ordre 2 et majoration

Développement de f à l'ordre 2 :

$$f(x+h) = f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(x) h + o(\|h\|^2). \tag{5}$$

Si ∇f est L-Lipschitz (ie, $\forall x, y,\ \|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$), alors on a la majoration suivante au voisinage de x:

$$f(x+h) \leq f(x) + \nabla f(x)^T h + \frac{L}{2} \|h\|^2. \tag{6}$$

permet de démontrer la conv. de la desc grad

Si f est 2 fois différentiable on peut poser $L = \sup_x \left\| \nabla^2 f(x) \right\|_{2 \to 2}$.

Convexité

Ensemble convexe

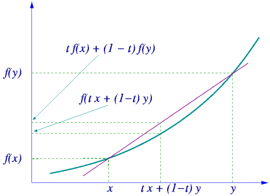
Un ensemble E est convexe si et seulement si $\forall x_1, x_2 \in E$, $\forall t \in]0,1[$,

$$tx_1 + (1-t)x_2 \in E.$$

Fonction convexe

La fonction f est convexe sur un ensemble convexe $E \subset \mathbb{R}^n$ si et seulement si $\forall x_1, x_2 \in E$, $\forall t \in]0,1[$,

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$$



Convexité stricte et forte

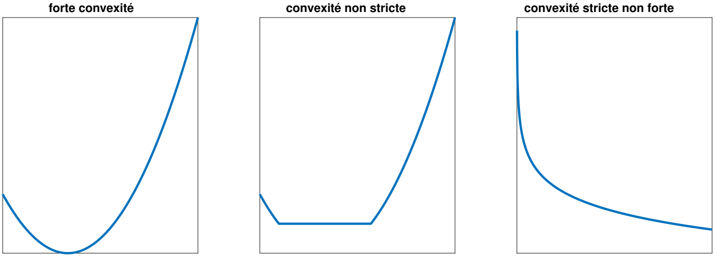
Fonction strictement convexe

Convexité stricte = égalité stricte dans la définition.

Fonction fortement convexe

La fonction f est μ-fortement convexe si et seulement si $\forall x_1, x_2 \in E$, $\forall t \in [0, 1]$,

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) - \frac{\mu}{2} t(1-t) \left\| x_1 - x_2 \right\|_2^2 .$$



Caractérisation différentielle de la convexité

Si f est de classe $\mathcal{C}^2$, on a les équivalences suivantes :

- f est convexe $\Leftrightarrow \nabla^2 f(x) \succeq 0 \; \forall x$
- f est strictement convexe $\Leftrightarrow \nabla^2 f(x) \succ 0 \; \forall x$
- f est μ-fortement convexe $\Leftrightarrow \nabla^2 f(x) - \mu I_n \succeq 0 \; \forall x$
- f a un gradient L-Lipschitz $\Leftrightarrow \nabla^2 f(x) - L I_n \preceq 0 \; \forall x$

Cas d'une forme quadratique

Si $f(x) = x^T Q x$ avec Q symétrique, alors

- $\mu = \lambda_{\mathsf{min}}(Q)$ la plus petite valeur propre
- $L = \lambda_{\mathsf{max}}(Q)$ la plus grande valeur propre
- $\kappa = L/\mu$ est le conditionnement de la matrice Q

Plan de la section

Conditions d'optimalité

$$\min_x f(x)$$

Au premier ordre (f est de classe $\mathcal{C}^1$)

- Condition nécessaire : si x_* est un minimum local, alors $\nabla f(x_*) = 0$.
- Condition nécessaire et suffisante : si f est convexe, x_* est un minimum global $\Leftrightarrow \nabla f(x_*) = 0$

Au second ordre (f est de classe $\mathcal{C}^2$)

- Condition nécessaire : si x_* est un minimum local, alors $\nabla f(x_*) = 0$ et $\nabla^2 f(x_*)$ est semi-définie positive (valeurs propres ≥ 0).
- Condition suffisante : si $\nabla f(x_*) = 0$ et $\nabla^2 f(x_*)$ est définie-positive, alors x_* est un minimiseur local strict.

Descente de gradient

Descente de gradient pour une fonction $f \in \mathcal{C}^1$:

Initialiser x_0; choisir des pas γ_k; itérer

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k). \tag{7}$$

Convergence

- Si ∇f est L-Lipschitz et $\gamma_k < \frac{2}{L}$, alors la descente de gradient converge vers un point stationnaire. Choix typique : $\gamma_k = \frac{1}{L}$.
- Si de plus f est convexe, on a une convergence linéaire :

$$f(x_k) - f_* \leq \frac{2}{k+4} \|x_0 - x_*\|_2^2 \tag{8}$$

- Si de plus f est μ-fortement convexe, on a une convergence exponentielle :

$$\|x_k - x_*\| \leq \left(1 - \frac{\mu}{L}\right)^k \|x_0 - x_*\|_2 \leq e^{-\frac{\mu k}{L}} \|x_0 - x_*\|_2 \tag{9}$$

Recherche de pas

Formulation

Au point courant x_k, on a identifié une direction de descente p (par exemple $p = -\nabla f(x_k)$). Le problème s'écrit

$$\min_{\gamma} f(x_k + \gamma p). \tag{10}$$

Intuitions

- Problème "facile", car 1D, souvent unimodal
- Inutile de chercher l'optimum, mieux vaut mettre à jour la direction

Armijo

- Trouver γ le plus grand possible tel que

$$f(x_k) - f(x_{k+1}) \geq \mu\gamma \left| p^T \nabla f(x_k) \right| \tag{11}$$

- Algorithme : choisir $\mu \approx 0.25$ et $\rho \in]0, 1[$. Commencer avec une borne supérieure sur γ. Tant que la condition n'est pas remplie, poser $\gamma = \gamma\rho$.

Gradient accéléré (Nesterov)

Gradient accéléré (Nesterov)

On suppose que f est convexe et son gradient est L-Lipschitz.

1. Initialisation : $t_0 = 1$, $\gamma = 1/L$, $k = 0$, point initial x_0, $y_0 = x_0$.
2. Itérations :
    - $x_{k+1} = y_k - \gamma \nabla f(x_k)$
    - $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
    - $y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k)$
    - $k := k + 1$

- Idée : les itérés successifs peuvent fournir une information de second ordre qui aident à la convergence
- Convergence quadratique :

$$f(x_k) - f_* \leq 2L \frac{\|x_0 - x_*\|_2^2}{(k+1)^2}.$$

- Appelé aussi gradient à moment, en particulier dans le contexte stochastique

Méthode de Newton

Méthode de Newton

- On suppose f de classe $\mathcal{C}^2$ et $\nabla^2 f(x) \succ 0 \ \forall x$.
- Méthode de Newton : direction de descente optimale à l'ordre 2 :

$$x_{k+1} = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k).$$

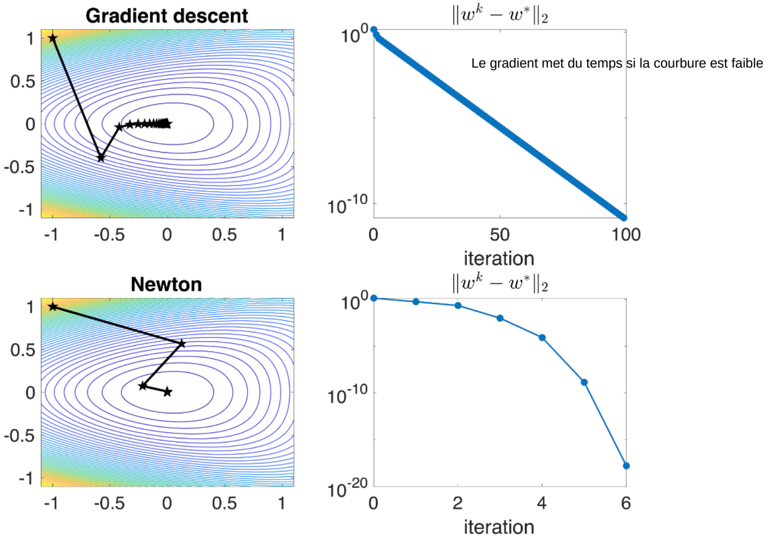
Convergence quadratique

$$\lim_{k \to +\infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|^2} = \tau > 0.$$

Remarques

- Très efficace, mais coûteux et impossible en grande dimension
- On peut ajouter un pas comme pour le gradient (*damped Newton*), pour notamment assurer des conditions de type Armijo ou Wolfe
- La convergence globale n'est pas assurée si la hessienne n'est pas globalement définie-positive

Illustration : gradient vs Newton



[Cornell University, CS 4/5780]

Méthodes de quasi-Newton

- Quasi-Newton : Newton sans matrice Hessienne.
- On minimise l'approximation quadratique :

$$f(x) \approx f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k).$$

- La solution est

$$x_{k+1} = x_k - \gamma_k B_k^{-1} \nabla f(x_k).$$

- Si $B_k = \nabla^2 f(x_k)$, on retrouve la méthode de Newton.

On remplace Bk par autre chose, si possible léger à calculer

BFGS (Broyden–Fletcher–Goldfarb–Shanno)

Quelles conditions pour B_{k+1}?

- Contrainte d'égalité du gradient entre f et l'approximation pour 2 itérés successifs :
$$\gamma_k B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k).$$
- En posant $s_k = x_{k+1} - x_k$ et $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, cela s'écrit
$$B_{k+1} s_k = y_k.$$

BFGS

- Mise à jour de l'approximation de la Hessienne :
$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}.$$
- Plus efficace de travailler sur l'inverse $H_k := B_k^{-1}$, la mise à jour devient :
$$H_{k+1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) H_k \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}. \tag{12}$$

BFGS en pratique

- Initialisation B_0 : inverse d'une approximation numérique de la Hessienne en x_0 si possible. Sinon, identité ou matrice diagonale.
- Convergence : en général super-linéaire, si le pas γ_k est bien choisi :

$$\lim_{k \to +\infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} = 0.$$

- BFGS en grande dimension : très coûteux voire impossible de stocker et calculer avec une matrice dense H_k, même factorisée (Cholesky). On peut à la place stocker p paires (s_k, y_k) et itérer p fois l'équation (12). L'algorithme s'appelle *L-BFGS*.

Moindres carrés non linéaires

On considère la fonction $F : \mathbb{R}^n \to \mathbb{R}^m$, avec $m > n$. On cherche à résoudre le problème $y = F(x)$ au sens des moindres carrés :

$$\min_x f(x) := \|F(x)\|_2^2 . \tag{13}$$

- F peut être un modèle physique ou un modèle statistique (régression non linéaire)
- On note $F(x) = (F_1(x), \cdots, F_m(x))$, la dérivée de F, appelée matrice jacobienne, s'écrit $J_F(x) = (\nabla F_1(x), \cdots, \nabla F_m(x))^T$.
- Le gradient de f s'écrit

$$\nabla f(x) = J_F(x)^T F(x).$$

- La Hessienne de f est

$$\nabla^2 f(x) = J_F(x)^T J_F(x) + \sum_{i=1}^m F_i(x) \nabla^2 F_i(x).$$

Algorithme de Gauss-Newton

C'est la méthode de Newton, dans laquelle on approche la matrice Hessienne par $\nabla^2 f(x) = J_F(x)^T J_F(x)$, en supposant que $F_i(x) \approx 0$ au voisinage de l'optimum.

Itérations de l'algorithme de Gauss-Newton

$$x_{k+1} = x_k - \left[J_F(x_k)^T J_F(x_k)\right]^{-1} J_F(x_k)^T F(x_k).$$

Remarques

- Comme dans la méthode de Newton, on ne calcule pas l'inverse mais on résout plutôt un système linéaire
- Il faut que $J_F(x_k)^T J_F(x_k)$ soit inversible
- Le système peut se résoudre via une décomposition QR, ou un algorithme de gradient conjugué

Algorithme de Levenberg-Marquardt

- Aka *Damped least-squares method* (DLS)
- Motivation : stabiliser l'algorithme, notamment dans les cas où l'approximation de la Hessienne est singulière ou mal conditionnée.
- Formulation initiale :

$$x_{k+1} = x_k - \left[J_F(x_k)^T J_F(x_k) + \lambda I\right]^{-1} J_F(x_k)^T F(x_k).$$

- Formulation qui tient compte des différences de courbure dans chaque direction :

$$x_{k+1} = x_k - \left[J_F(x_k)^T J_F(x_k) + \lambda \operatorname{diag}(J_F(x_k)^T J_F(x_k))\right]^{-1} J_F(x_k)^T F(x_k).$$

Influence du paramètre λ

- Si $\lambda \to 0$, on retrouve la méthode de Gauss-Newton
- Si λ est grand, on se rapproche d'une descente de gradient classique
- Plusieurs stratégies existent pour mettre à jour λ au fur et à mesure des itérations

Plan de la section

Optimisation sous contraintes

Le problème s'écrit maintenant

$$\min_{x \in \Omega} f(x), \text{ avec } \Omega = \{x | h_i(x) = 0,\ i \in \mathcal{E}, g_i(x) \leq 0,\ i \in \mathcal{I}\} \tag{14}$$

Lagrangien

On définit le Lagrangien associé au problème (14) par

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i \in \mathcal{E}} \lambda_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x) \tag{15}$$

Optimum = point selle du Lagrangien

x_0 est solution du problème (14) si et seulement si il est solution de

$$\min_{x} \max_{\lambda} \mathcal{L}(x, \lambda).$$

Conditions d'optimalité

Définitions

- $x \in \Omega$ est appelé point admissible du problème (14)
- Une contrainte d'inégalité $g_i(x) \leq 0$ est dite active (ou saturée) en x_* si $g_i(x_*) = 0$. Elle est dite inactive sinon.

Conditions dites de KKT (Karush,Kuhn,Tucker)

Si f, g et h sont de classe $\mathcal{C}^1$, si x_* est une solution locale de (14) et si les contraintes d'inégalité sont qualifiées, alors il existe un multiplicateur de Lagrange λ_* tel que :

$$\nabla_x \mathcal{L}(x_*, \lambda_*) = 0 \tag{16}$$

$$\lambda_{*i} \geq 0 \, \forall i \in \mathcal{I} \tag{17}$$

$$\lambda_{*i} g_i(x_*) = 0 \, \forall i \in \mathcal{I} \cup \mathcal{E} \tag{18}$$

La dernière condition implique qu'une contrainte est soit active, soit que son multiplicateur est nul.

- Un point satisfaisant ces conditions d'optimalité est dit stationnaire

Gradient projeté

On considère le problème

$$\min_x f(x) \text{ s. t. } x \in \Omega, \tag{19}$$

où $\Omega \subset \mathbb{R}^n$ est convexe, fermé, non vide.

Opérateur de projection sur Ω

Soit $x \in \mathbb{R}^n$, la projection de x sur Ω est l'application

$$p_\Omega(x) = \arg\min_{y\in\Omega} \frac{1}{2} \|x - y\|_2^2 .$$

Algorithme du gradient projeté

On enchaîne des descentes de gradient et des projections:

$$x_{k+1} = p_\Omega(x_k - \gamma \nabla f(x_k)).$$

- Convergence si choix adapté pour le pas γ.
- Variantes lagrangiennes : méthodes d'Uzawa et de Arrow-Hurwicz

Pénalisation des contraintes

On considère le problème

$$\min_x f(x) \text{ s. t. } g(x) = 0, \tag{20}$$

où $g : \mathbb{R}^n \to \mathbb{R}^m$ définit m contraintes d'égalité.

Méthode de la pénalisation

- Résoudre une version pénalisée sans contraintes :

$$\min_x f(x) + \mu \|g(x)\|_2^2$$

- Augmenter le paramètre $\mu > 0$ à chaque itération

Différentes pénalités possibles, par exemple

- Quadratique pour des contraintes d'égalité ou $\leq$
- Logarithmique pour des contraintes $x \geq 0$

Lagrangien augmenté

On reste sur le même problème sous contraintes d'égalité (20). Le Lagrangien augmenté s'écrit

$$\mathcal{L}_{\mathsf{aug}}(x;\lambda,\mu) = f(x) + \lambda^T g(x) + \frac{\mu}{2}\left\|g(x)\right\|_2^2 . \tag{21}$$

Méthode du Lagrangien augmenté

On cherche un point-selle de $\mathcal{L}_{\mathsf{aug}}$:

- Choix de $\mu > 0$, initialisation de λ_0, x_0, $k = 0$.
- Itérations
  - $x_{k+1} = \arg\min_x \mathcal{L}_{\mathsf{aug}}(x_k;\lambda_k,\mu)$
  - $\lambda_{k+1} = \lambda_k + \mu g(x_{k+1})$
  - Toutes les P itérations, on augmente μ

Plan de la section

1. Bases mathématiques

2. Optimisation différentiable sans contraintes

3. Optimisation sous contraintes

4. Optimisation non lisse et *splitting*
   - Motivation
   - Opérateur proximal
   - Algorithmes proximaux

5. Cas d'étude

6. Conclusion

Optimisation non-lisse

On considère à présent des problèmes du type

$$\min_x f(x) + g(x)$$

avec f différentiable et g non-différentiable

Exemples en apprentissage ou en imagerie

- f est une attache aux données, typiquement des moindres carrés
- g est une régularisation
- Pénalisations favorisant la parcimonie:
  - Parcimonie: $g(x) = \|x\|_0 = \#\{i | x_i \neq 0\}$
  - Relaxation convexe : $g(x) = \|x\|_1$ (LASSO)
  - Parcimonie dans un domaine transformé : ondelettes, variation totale, dictionnaires
  - Rang faible: norme nucléaire = norme 1 des valeurs singulières
  - Parcimonie structurée : ℓ_{12} (group-LASSO)

Opérateur proximal (Moreau 1965)

Définition

Soit $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ une fonction semi-continue inférieurement, convexe, de domaine non-vide. L'opérateur proximal de g est la fonction

$$\mathrm{prox}_g(x) = \arg\min_y g(y) + \frac{1}{2} \|x - y\|_2^2 . \tag{22}$$

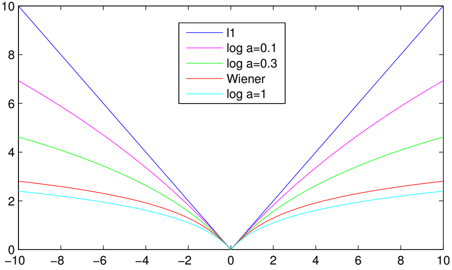
Exemple

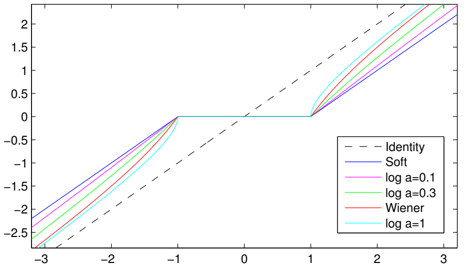
$$\mathrm{prox}_{\lambda \|\cdot\|_0}(x) = x \odot \mathbb{1}_{|x| > \lambda} \text{ seuillage dur}$$

$$\mathrm{prox}_{\lambda \|\cdot\|_1}(x) = \mathrm{signe}(x) \odot (|x| - \lambda)_+ \text{ seuillage doux}$$

Extension possible aux fonctions non convexes.

Opérateurs proximaux et parcimonie





Gradient proximal

Aka forward-backward ou ISTA, lorsque f est différentiable à gradient L-Lipschitz.

Algorithme : gradient + prox

$$x_{k+1} = \operatorname{prox}_{\gamma g}\left(x_k - \gamma \nabla f(x_k)\right). \tag{23}$$

- Comme pour le gradient, algorithme MM (majorization-minimization) avec une majorante quadratique
- Convergence linéaire si $0 < \gamma \leq 1/L$
- Variante accélérée à la Nesterov : FISTA
- Si $g(x) = \delta_\Omega(x)$, alors $\operatorname{prox}_g = p_\Omega$ et on retrouve le gradient projeté

Douglas-Rachford

Lorsqu'on connaît les prox de f et g, différentiables ou non.

Algorithme de Douglas-Rachford

- Initialiser y_0, fixer $\rho > 0$ et $\varepsilon \in]0,1[$ et $\gamma \in]\varepsilon, 2-\varepsilon[$
- Itérer

$$\begin{aligned} x_n &= \operatorname{prox}_{\rho g}(y_n) \\ y_{n+1} &= y_n + \gamma \left[\operatorname{prox}_{\rho f}\left(2x_n - y_n\right) - x_n\right] \end{aligned}$$

ADMM

On considère le problème

$$\min_{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m} f(x) + g(y) \text{ s. t. } Ax + By = c. \tag{24}$$

Son Lagrangien augmenté s'écrit :

$$\mathcal{L}_{\mathsf{aug}}(x, y, \lambda) = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{r}{2} \|Ax + By - c\|_2^2 .$$

ADMM : une adaptation du Lagrangien augmenté

$$\begin{aligned}
x_{k+1} &\in \arg\min_x \mathcal{L}_{\mathsf{aug}}(x, y_k, \lambda_k) \\
y_{k+1} &\in \arg\min_y \mathcal{L}_{\mathsf{aug}}(x_{k+1}, y, \lambda_k) \\
\lambda_{k+1} &= \lambda_k + r_k(Ax_{k+1} + By_{k+1} - c)
\end{aligned}$$

Si $A = -B = I$ et r_k bien choisi, on retrouve l'algorithme de Douglas-Rachford.

PALM[1]

On considère le problème suivant :

$$\min_{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m} f(x) + g(y) + H(x, y) \tag{25}$$

avec H une fonction de classe $\mathcal{C}^1$ et à gradients $L(y)$ et $L(x)$-Lipschitz, respectivement aux variables x et y. f et g sont des fonctions lsc et propres, possiblement non-convexes.

PALM = gradient proximal alterné non-convexe

- Initialisation de x_0, y_0, choix de $\gamma \in]0, 1[$
- Itérations

$$\begin{aligned} c_k &= \gamma / L_1(y_k) \\ x_{k+1} &= \operatorname{prox}_{c_k f}\left(x_k - c_k \nabla_x H(x_k, y_k)\right) \\ d_k &= \gamma / L_2(x_{k+1}) \\ y_{k+1} &= \operatorname{prox}_{d_k f}\left(y_k - d_k \nabla_y H(x_{k+1}, y_k)\right) \end{aligned}$$

[1] Bolte, Jérôme, Shoham Sabach, and Marc Teboulle. "Proximal alternating linearized minimization for nonconvex and nonsmooth problems." Mathematical Programming 146.1 (2014): 459-494.

Plan de la section

Déconvolution d'images

Modèle direct

$$y = Hx + b,$$

avec $x \in \mathbb{R}^n$ l'image idéale, $y \in \mathbb{R}^n$ la mesure, $b \in \mathbb{R}^n$ le bruit, et H la convolution 2D avec la PSF instrument.

Problème inverse

Si on suppose $b \sim \mathcal{N}(0, \sigma^2 I)$, il est naturel de minimiser la log-vraisemblance qui devient un terme de moindres carrés :

$$f(x) = -\log p(y|x) = \frac{1}{2\sigma^2} \|y - Hx\|_2^2 + \mathrm{Cst}\,.$$

- Solution unique si H inversible
- Calculable facilement en Fourier (avec approximation circulaire) : $\hat{x}_{\mathsf{MLE}} = (FH)^{-1} Fy$
- Mais H mal conditionnée : on doit régulariser

Régularisations quadratiques

Tikhonov

$$f(x) = \frac{1}{2\sigma^2} \|y - Hx\|_2^2 + \lambda \|x\|_2^2 .$$

- MAP avec une loi a priori normale
- Solution explicite : $\hat{x} = (H^T H + \lambda I)^{-1} H^T y$
- Calculable explicitement en Fourier

Sobolev

$$f(x) = \frac{1}{2\sigma^2} \|y - Hx\|_2^2 + \lambda \|Dx\|_2^2$$

avec D un opérateur de différences finies.

- Favorise des solutions spatialement lisses
- Calculable explicitement en Fourier (avec approximation circulaire)
- Problème : génère une image floue

Régularisation parcimonieuse dans une base

Les images naturelles sont parcimonieuses dans des bases bien choisies, comme les ondelettes. En notant W une transformée en ondelettes orthogonale, on peut ainsi régulariser le problème avec une norme ℓ_1 :

$$\hat{x} = \arg\min_x \frac{1}{2\sigma^2} \|y - Hx\|_2^2 + \lambda \|Wx\|_1$$

Algorithme du gradient proximal

- L'attache aux données $f(x) = \frac{1}{2\sigma^2} \|y - Hx\|_2^2$ est différentiable, son gradient vaut $\nabla f(x) = \frac{1}{\sigma^2} H^T(Hx - y)$ et est L-Lipschitz avec $L = \|H\|_{2\to 2}^2 / \sigma^2$.
- La régularisation $R(x) = \lambda \|Wx\|_1$ n'est pas différentiable mais on peut calculer facilement son opérateur proximal :

$$\mathrm{prox}_R(x) = \arg\min_y \lambda \|Wy\|_1 + \frac{1}{2} \|x - y\|_2^2 = W^{-1}\, \mathrm{ST}_\lambda(Wx).$$

Régularisation beaucoup utilisée : $R(x) = \|Dx\|_1$ ou $\|Dx\|_{12}$, variation totale, algorithme primal-dual dit de Chambolle-Pock.

Régularisations plug-and-play

Formulation par *splitting* avec une régularisation quelconque :

$$\min_{x,z} \frac{1}{2\sigma^2} \|y - Hx\|_2^2 + \lambda R(z) \text{ s. t. } x = z.$$

Half-quadratic splitting

$$x_{k+1} = \arg\min_x \frac{1}{2\sigma^2} \|Hx - y\|_2^2 + \mu \|x - z_k\|_2^2 \text{ solution explicite en Fourier}$$

$$z_{k+1} = \arg\min_z \lambda R(z) + \frac{\mu}{2} \|z - x_{k+1}\|_2^2 = \operatorname{prox}_{\frac{\lambda}{\mu} R}(x_{k+1}) \text{ débruitage}$$

Plug-and-play HQS[2] : remplacer la seconde étape par un débruiteur appris auparavant.

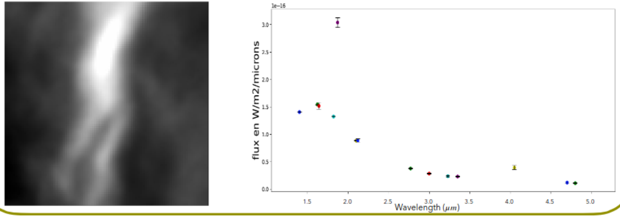
Plus de détails dans le poster et l'exposé de Maud Biquard, collaboration avec Marie Chabert, Florence Genin et Christophe Latry

[2] Zhang, Kai, et al. "Plug-and-play image restoration with deep denoiser prior." IEEE Transactions on Pattern Analysis and Machine Intelligence 44.10 (2021): 6360-6376.

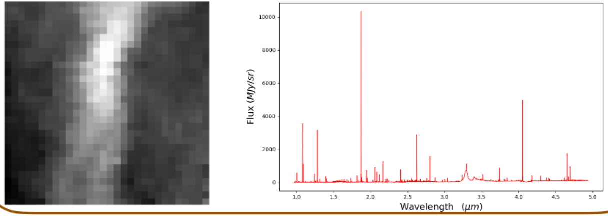
Fusion d'images pour le JWST

Avec Claire Guilloteau, Landry Marquis, Nicolas Dobigeon et Olivier Berné

JWST's NIRCam: high *spatial* resolution, low spectral resolution



JWST's NIRSpec: low spatial resolution, high *spectral* resolution



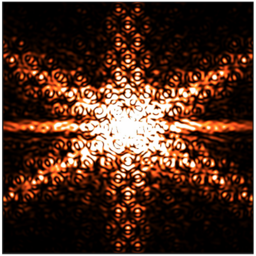
Data: E. Habart et al., PDRs4All: JWST's NIR and MIR imaging view of the Orion Nebula, A & A, 2023
E. Peters et al., PDRs4All: JWST's NIR spectroscopic view of the Orion Bar, A & A, 2024

Fusion d'images pour le JWST

Modèles directs linéaires mais complexes, régularisation spectrale (sous-espace) et spatiale (Sobolev), résolution par gradient conjugué[3]

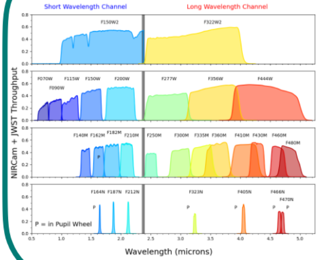
$$\gamma_m \| Y_m - L_m W_m(X) \|_F^2 + \gamma_h \| Y_h - L_h W_h(X) S \|_F^2 + \mu \| \nabla X \|_F^2$$

$$W_m = PSF(\lambda)$$



NIRCam point spread function at 3.3μm

$$L_m(filter)$$



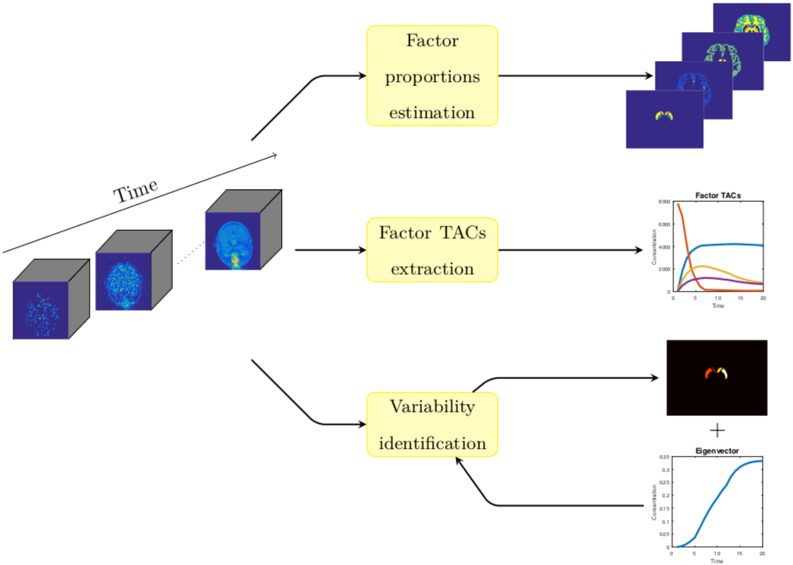
NIRCam filters

Sources: WebbPSF, STScI website

[3]Claire Guilloteau et al. *Hyperspectral and multispectral image fusion under spectrally varying spatial blurs–Application to high dimensional infrared astronomical imaging.* IEEE Transactions on Computational Imaging, 6, 2020.

Démélange spectral non-linéaire pour la TEP dynamique

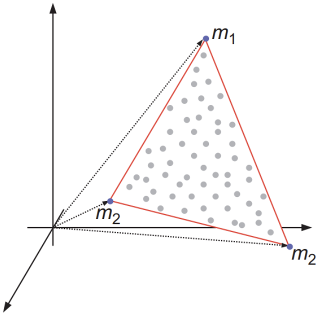
Avec Yanna Cruz Cavalcanti, Clovis Tauber et Nicolas Dobigeon



Démélange / analyse factorielle

Réduction de dimension / analyse factorielle (M : facteurs, A : coefficients)

$$X \approx MA = \sum_{k=1}^{K} M_k A^k$$



Démélange et simplexe
[Dobigeon et al., 2016]

$$\min_{M,A} \|X - MA\|_F^2 + \text{contraintes}$$

- ACP : $M^T M = I$
- NMF : $A, M \geq 0$
- Démélange : $A, M \geq 0$ et $1A = 1$

Démélange spectral non-linéaire

Modèle de mélange[4] :

$$x_n = a_{1,n}\left(\bar{m}_1 + \sum_{i=1}^{N_v} b_{i,n} v_i\right) + \sum_{k=2}^{K} a_{k,n} m_k. \tag{26}$$

Modèle direct en notation matricielle :

$$Y = MAH + \underbrace{\left[EA \circ VB\right]H}_{\Delta} + R \tag{27}$$

On résout

$$(M^*, A^*, B^*) \in \underset{M,A,B}{\arg\min} \left\{ \mathcal{J}(M,A,B) \text{ s.t. } M \geq 0,\ A \geq 0,\ 1A = 1, B \geq 0 \right\}$$

$$\text{avec } \mathcal{J}(M,A,B) = \frac{1}{2}\left\| Y - MAH - \left[EA \circ VB)\right] H \right\|_F^2 + \alpha \left\| AD \right\|_2^2 + \beta \left\| M - M_0 \right\| + \lambda \left\| B \right\|_1$$

[4] Yanna C. Cavalcanti et al. *Unmixing dynamic PET images with variable specific binding kinetics.* Medical image analysis, 49, 2018.

Résolution avec PALM

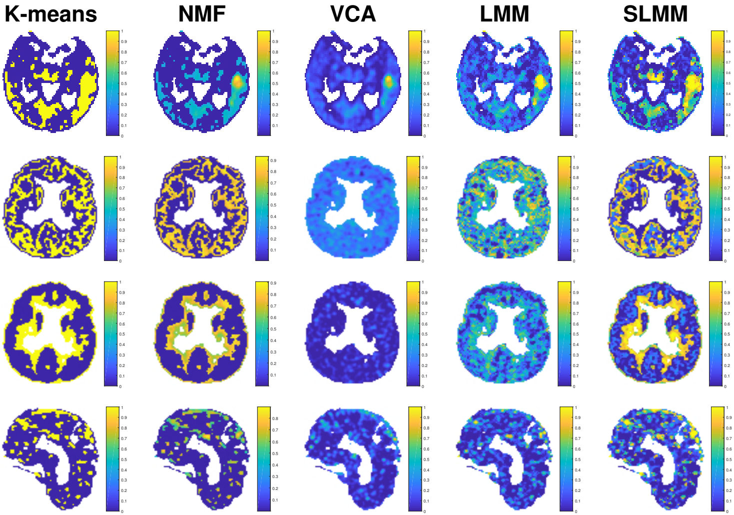
PALM : gradient proximal alterné. Pour calculer les itérations, il faut expliciter les gradients, les constantes de Lipschitz et les opérateurs proximaux. Exemple pour M:

$$M^{k+1} = \mathcal{P}_+\left(M^k - \frac{1}{L_M^k}\nabla_M \mathcal{J}(M^k, A^{k+1}, B^k)\right)$$

avec $\nabla_M \mathcal{J} = ((E_1 A \circ VB)\, H - Y)\, H^T A^T + M(AHH^T A^T) + \beta(M - M^0)$

et $L_M^k = \left\| AHH^T A^T \right\|_{2\to 2} + \beta$

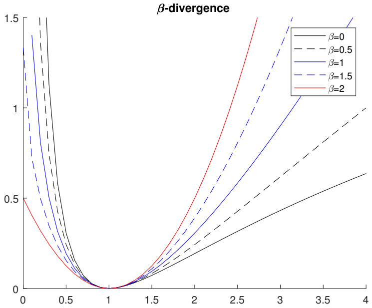
Quelques résultats



Cartes des abondances A avec, de haut en bas : matière grise spécifique, matière blanche, matière grise non spécifique, sang.

Influence de la fonction-coût

- Le bruit en TEP dynamique n'est pas Gaussien : il contient des composantes poissonniennes, gaussiennes et Gamma multiplicatives
- Les statistiques du bruit de mesure sont fortement modifiées par la reconstruction 3D, via un algorithme souvent propriétaire
- Une solution[5] : utiliser une famille de fonction-coût flexibles : les β-divergences



[5] Cavalcanti, Yanna Cruz, et al. *Factor analysis of dynamic PET images: beyond Gaussian noise.* IEEE Transactions on Medical Imaging, 2019.

Algorithme MM

Formulation majoration-minimisation avec mises à jour multiplicatives :

$$\tilde{Y} = M^k A^k + \left[E_1 A^k \cdot V B^k \right]$$

$$B^{k+1} = B^k \cdot \left[\frac{1_{N_v}^T A_{1,:} \cdot (V^T (Y \cdot \tilde{X}^{\beta-2}))}{1_{N_v}^T A_{1,:} \cdot (V^T \tilde{X}^{\beta-1}) + \lambda B^k \mathbf{\Gamma}_B} \right]^{\frac{1}{3-\beta}}$$

$$\tilde{X} = M^k A^k + \left[E A^k \cdot V B^{k+1} \right]$$

$$M_{2:K}^{k+1} = M_{2:K}^k \left[\frac{(Y \cdot \tilde{X}^{\beta-2}) A_{2:K}^T}{\tilde{X}^{\beta-1} A_{2:K}^T} \right]$$

$$\tilde{X} = M^{k+1} A^k + \left[E A^k \cdot V B^{k+1} \right]$$

$$A_1^{k+1} = A_1^k \cdot \left[\frac{1_L^T ((M_1 1_N^T + V B) \cdot (Y \cdot \tilde{X}^{\beta-2}) + \tilde{x}^\beta)}{1_L^T ((M_1 1_N^T + V B) \cdot \tilde{X}^{\beta-1} + Y \cdot \tilde{X}^{\beta-1})} \right]$$

$$A_{2:K}^{k+1} = A_{2:K}^k \cdot \left[\frac{M_{2:K}^T (Y \cdot \tilde{X}^{\beta-2}) + 1_{K-1,L} \tilde{X}^\beta}{M_{2:K}^T \tilde{X}^{\beta-1} + 1_{K-1,L} (Y \cdot \tilde{X}^{\beta-1})} \right]$$

Plan de la section

Take-home message

Formulation d'un problème d'optimisation

- Utiliser une fonction coût pertinente, possiblement basée sur la vraisemblance si on a un modèle statistique du bruit
- Ajouter des contraintes
- Ajouter des termes de régularisation

Choix d'un algorithme pour résoudre le problème

- Identifier le cadre : différentiable, convexe, contraintes, etc
- Tenir compte de la dimension du problème
- Modifier si besoin la formulation du problème
- Régler les paramètres de l'algorithme et bien initialiser

Autres cadres d'optimisation : programmation linéaire, en nombre entiers, programmation par contraintes, variable complexe...

Optimisation pour l'apprentissage

Descente de gradient stochastique

- En apprentissage, on a souvent des fonctions de coût qui s'écrivent

$$f(x) = \frac{1}{N} \sum_{x_i \in \text{Data}} g(x_i),$$

- Calcul d'un seul gradient = parcourir tout le jeu de données!
- On fait à la place une descente de gradient stochastique en approchant le gradient grâce à un minibatch :

$$\nabla f(x) \approx \frac{1}{\# B} \sum_{i \in B} g(x_i).$$

Quelques algorithmes de gradient stochastique

- Descente de gradient stochastique (SGD)
- Accélération à la Nesterov : SGD with momentum
- Pas adaptatifs : Adagrad, Adadelta, Adam