

邻近点 算法

MODULE B3

Algorithme du point proximal

Sauf mention contraire, \mathcal{X} (resp. \mathcal{Y}) est un espace de HILBERT, muni du produit scalaire noté $\langle \cdot, \cdot \rangle$ et on note $\| \cdot \|$ la norme qui découle du produit scalaire.

Dans ce module, on s'intéresse au problème de minimisation suivant

$$\min_{x \in \mathcal{X}} J(x) \quad (\mathcal{P})$$

où $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ est une fonction s.c.i. On va introduire un algorithme pour résoudre ce problème, basé sur l'utilisation de l'opérateur proximal.

1 Algorithme du point proximal (PPA) 邻近点算法

1.1 Algorithme du gradient implicite 隐式梯度算法

Soit $J : \mathcal{X} \rightarrow \mathbb{R}$ une fonction différentiable. On suppose qu'elle admet un minimiseur. Si elle est convexe régulière, alors on sait qu'on peut appliquer la méthode du gradient explicite à pas constant pour en déterminer un minimiseur (module B2 : Méthodes du gradient explicite). Les itérations de cet algorithme sont, rappelons-le, données par

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = x_k - \tau \nabla J(x_k)$$

On sait que $-\nabla J(x_k)$ est une direction de descente pour J au point x_k , et que, si τ est correctement choisi, alors la suite des $J(x_k)$ décroît.

Un algorithme assez proche est celui dit du gradient implicite. Il ne diffère de la méthode du gradient explicite que du choix du point où le gradient est calculé :

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = x_k - \tau \nabla J(x_{k+1})$$

Le qualificatif d'implicite (et, de même, celui d'explicite) se justifie donc par le fait que, dans cet algorithme, le gradient est évalué en un point futur x_{k+1} , contrairement à la méthode précédente où il est évalué au point courant x_k .

En anglais, on parle plutôt de *forward* (explicite) et de *backward* (implicite) step.

Il est évident que, écrit de cette manière, l'algorithme du gradient implicite n'est pas exploitable en pratique, puisque le gradient de J doit être évalué en un point inconnu. Cependant, il suffit de réarranger la définition de l'itérée x_{k+1} pour s'apercevoir qu'elle obéit à l'égalité

$$x_{k+1} + \tau \nabla J(x_{k+1}) = x_k$$

Ainsi, si J est convexe, alors la proposition 5 du module A5 : Opérateur proximal de MOREAU assure que

$$x_{k+1} = \text{prox}_{\tau J}(x_k)$$

de sorte que, si l'opérateur proximal associé à J est évaluable en tout point de \mathcal{X} , alors les itérations de l'algorithme du gradient implicite peuvent être calculées.

1.2 Description de l'algorithme PPA PPA 算法的描述

Intéressons-nous donc plus particulièrement à l'algorithme suivant :

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} \in \text{prox}_{\tau J}(x_k)$$

(notons que, si J n'est pas convexe, alors l'opérateur proximal est multi-valué mais bien défini si J est minorée, et les itérations ne sont donc pas définies de manière unique). Cet algorithme est connu sous le nom d'*algorithme du point proximal* (ou PPA – *Proximal Point Algorithm* en anglais). Il a été introduit par MARTINET en 1970. On a vu dans le paragraphe précédent que, si J est convexe et différentiable, alors cet algorithme est équivalent à la méthode du gradient implicite.

Contrairement à l'algorithme du gradient implicite, l'algorithme du point proximal ne nécessite pas la différentiabilité de la fonction objectif J , puisque l'opérateur proximal peut être bien défini pour des fonctions non différentiables. Cependant, pour que cet algorithme soit bien défini, il faut que l'opérateur proximal soit bien défini en tout point de \mathcal{X} (ou, du moins, en tout point du domaine de J si l'on est capable de trouver au moins un point x_0 dans $\text{dom } J$).

Proposition 1 邻近点在定义域中

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction s.c.i. Soit $x^0 \in \mathcal{X}$. Alors

$$\text{prox}_J(x^0) \subset \text{dom } J$$

DÉMONSTRATION : Par définition du minimiseur d'une fonction, les éléments de l'ensemble $\text{prox}_J(x^0)$ appartiennent au domaine de la fonction

$$x \mapsto J(x) + \frac{1}{2} \|x - x^0\|^2$$

qui partage même domaine que J . ■

Ainsi, l'algorithme du point proximal génère une suite de points appartenant au domaine de J , c'est-à-dire un point admissible (ou réalisable) pour le problème (\mathcal{P}) . Il s'agit d'une propriété intéressante, car elle assure que, quel que soit le moment où l'on stoppe les itérations de la méthode, le point courant, à défaut d'être un minimiseur recherché, est au moins un point admissible. Par ailleurs, cela permet d'appliquer des mesures sur la "qualité" du point courant, beaucoup de telles mesures étant raisonnablement définies uniquement pour les points admissibles.

Un autre manière d'interpréter les itérations de l'algorithme du point proximal est de les voir comme des itérations du point fixe. Cependant, l'opérateur proximal n'étant pas contractant (mais lipschitzien de constante 1 – cf. corollaire 1 du module A5 : Opérateur proximal de MOREAU lorsque la fonction est convexe), cette propriété n'est pas immédiatement exploitable.

Terminons enfin en notant à nouveau que, d'après la proposition 7 du module A5 : Opérateur proximal de MOREAU,

$$\begin{aligned} x_{k+1} \in \text{prox}_{\tau J}(x_k) &\implies x_k - x_{k+1} \in \partial(\tau J)(x_{k+1}) \\ &\implies x_{k+1} \in x_k - \tau \partial J(x_{k+1}) \end{aligned}$$

Autrement dit, si x_{k+1} est bien défini, alors il existe un sous-gradient $p_{k+1} \in \partial J(x_{k+1})$ tel que

$$x_{k+1} = x_k - \tau p_{k+1}$$

pour tout $k \in \mathbb{N}$. Ainsi, l'algorithme du point proximal est un cas particulier de la *méthode du sous-gradient implicite*. L'identité entre ces deux méthodes est vérifiée lorsque J est convexe (d'après la proposition 5 du module **A5 : Opérateur proximal de MOREAU**).

1.3 Avantages et inconvénients de la méthode PPA PPA 算法的优点和缺点

Commençons par lister les avantages de l'algorithme du point proximal par rapport à la méthode du gradient explicite : 邻近点算法 相比于 显式梯度法的 优劣

- aucune régularité n'est nécessaire, ni pour appliquer la méthode (J n'est pas besoin d'être différentiable car son gradient n'intervient pas dans la définition des itérées), ni pour en étudier la convergence (le gradient de J n'a pas besoin d'être lipschitzien) ;
- en particulier, cette méthode est permet en théorie de résoudre des problèmes d'optimisation sous contraintes ;
- comparé à la méthode du sous-gradient explicite, il n'est pas besoin de savoir évaluer le sous-gradient de J en un point quelconque ;
- on verra que le pas de temps τ n'est pas majoré ; il peut être choisi arbitrairement tout en assurant la convergence de la méthode. Néanmoins, il faut noter que la vitesse de convergence du schéma dépend elle de ce choix.

De manière attendue, la méthode du point proximal présente des points faibles, parfois insurmontables :

- l'évaluation de l'opérateur proximal en un point quelconque peut être une tâche ardue (il suffit pour s'en convaincre de considérer le cas particulier des projections orthogonales sur un convexe fermé) ;
- on remplace un problème de minimisation (\mathcal{P}) par une série de problèmes d'optimisation auxiliaires, augmentant *a priori* la complexité de la tâche à résoudre.

Concernant ce dernier point, il faut toutefois nuancer le propos en remarquant que les problèmes auxiliaires (évaluation de l'opérateur proximal) peuvent présenter des propriétés plus intéressantes que celles du problème initial (\mathcal{P}). En effet, si le problème initial est convexe, alors il s'agit de problèmes fortement convexes, dont la solution existe et est unique. Par ailleurs, lorsqu'une formule explicite n'existe pas pour définir le point proximal, on a recours à des méthodes d'optimisation convexe pour approcher ce point. Or, comme on l'a vu pour la méthode du gradient explicite et comme on le verra dans les modules suivants, une grande majorité de telles méthodes ont des comportements très intéressants en présence de fonctions fortement convexes (notamment en matière de vitesse de convergence).

Il faut cependant prendre garde au fait que, la plupart du temps, le point calculé de cette manière approche uniquement le point proximal, et que l'erreur commise peut affecter grandement le comportement de l'algorithme du point proximal si elle n'est pas suffisamment contrôlée. On parle d'ailleurs à ce sujet de *méthodes inexactes*, pour les distinguer des méthodes dans lesquelles chaque itération est calculée de manière exacte. Des travaux étudient la convergence de telles méthodes, en fournissant des bornes sur l'erreur d'approximation.

1.4 La méthode PPA dans le cas convexe 下凸函数情形的PPA方法

Lorsque J est supposée **convexe**, on peut expliciter un lien de l'algorithme du point proximal avec la méthode du gradient explicite. Si on **réécrit** différemment les **itérations** de l'algorithme du point proximal, on obtient

$$x_{k+1} = \text{prox}_{\tau J}(x_k) = x_k - x_k + \text{prox}_{\tau J}(x_k)$$

Or, la **proposition 12** du module **A5 : Opérateur proximal de MOREAU** assure que, dans ce cas,

$$x_{k+1} = x_k - \tau \nabla({}^{\tau}J)(x_k)$$

où on rappelle que l'**enveloppe de MOREAU** d'indice $\tau > 0$ (appelée encore régularisation de MOREAU–YOSIDA) est définie par

$$\forall x^0 \in \mathcal{X}, \quad {}^{\tau}J(x^0) = \min_{x \in \mathcal{X}} \left\{ J(x) + \frac{1}{2\tau} \|x - x^0\|^2 \right\}$$

Ainsi, les itérations de l'algorithme du point proximal peuvent être **vues comme** celles de la **méthode du gradient explicite à pas constant $1/\tau$** , appliquée au problème d'optimisation

$$\min_{x \in \mathcal{X}} {}^{\tau}J(x) \quad (\mathcal{P})_{\text{M-Y}}$$

Puisque la fonction objectif J du **problème initial** (\mathcal{P}) et son **enveloppe de MOREAU** ${}^{\tau}J$ partagent **mêmes minimiseurs** (proposition 14 du module **A5 : Opérateur proximal de MOREAU**), on en déduit que les problèmes (\mathcal{P}) et $(\mathcal{P})_{\text{M-Y}}$ sont **équivalents**.

Pour tout $\tau > 0$, la fonction ${}^{\tau}J$ est une fonction **convexe** et $(1/\tau)$ -régulière (proposition 13 et corollaire 2 du module **A5 : Opérateur proximal de MOREAU**). Aussi, on peut utiliser les résultats de convergence établis pour la méthode du gradient explicite, en notant que la condition sur le pas de temps est toujours satisfaite ici, puisque

$$\tau = \frac{1}{1/\tau} < \frac{2}{1/\tau}$$

où $1/\tau$ est la **constante de LIPSCHITZ** de ∇J .

On termine cette section avec l'observation suivante : par définition de l'**enveloppe de MOREAU d'indice τ** , on a

$$\forall x^0 \in \mathcal{X}, \quad {}^{\tau}J(x^0) = J(\text{prox}_{\tau J}(x^0)) + \frac{1}{2\tau} \|\text{prox}_{\tau J}(x^0) - x^0\|^2$$

En particulier, on a pour tout $k \in \mathbb{N}$

$${}^{\tau}J(x_k) = J(x_{k+1}) + \frac{1}{2\tau} \|x_{k+1} - x_k\|^2 \leq J(x_k)$$

et tout **minimiseur x^* de J**

$${}^{\tau}J(x^*) = J(x^*) + \frac{1}{2\tau} \|x^* - x^*\|^2 = J(x^*)$$

2 Résultats de convergence 收敛结论

2.1 Convergence dans le cas convexe 下凸函数情形的收敛

En utilisant le fait que, dans le cas d'une fonction objectif **convexe**, les itérées générées par l'algorithme du point proximal de pas de temps τ peuvent être **vues comme** celles générées par la **méthode du gradient explicite** de même pas de temps, mais appliquée à la minimisation de la régularisation de MOREAU–YOSIDA ${}^{\tau}J$, on peut **déduire** des

résultats de convergence de la méthode du gradient explicite (module **B2 : Méthodes du gradient explicite**) les résultats qui suivent. Ainsi, on utilise la régularité de ${}^\tau J$, et le fait que le pas de temps satisfait automatiquement les contraintes apparaissant dans les résultats du module **B2 : Méthodes du gradient explicite**, pour montrer que

Lemme 1 — Moreau 包络函数值的收敛性

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction **convexe**, s.c.i. et propre. On suppose que J est **minorée**. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors la suite $({}^\tau J(x_k))_{k \in \mathbb{N}}$ est décroissante et converge vers une valeur J^* . De plus, on a

$${}^\tau J(x_{k+1}) \leq {}^\tau J(x_k) - \frac{1}{2\tau} \|x_{k+1} - x_k\|^2$$

DÉMONSTRATION : Il s'agit de l'application directe de la proposition 4 du module **B2 : Méthodes du gradient explicite**, en remarquant que

$${}^\tau J(x_k) = \frac{1}{\tau} (x_k - \text{prox}_{\tau J}(x_k)) = \frac{1}{\tau} (x_k - x_{k+1})$$

d'après la définition de x_{k+1} . ■

En particulier, on en déduit d'après les calculs qui précèdent que, pour tout $k \in \mathbb{N}$,

$$\begin{aligned} J(x_{k+2}) + \frac{1}{2\tau} \|x_{k+2} - x_{k+1}\|^2 &\leq J(x_{k+1}) + \frac{1}{2\tau} \|x_{k+1} - x_k\|^2 - \frac{1}{2\tau} \|x_{k+1} - x_k\|^2 \\ &\leq J(x_{k+1}) \end{aligned}$$

ce qui entraîne la décroissance, et donc la convergence, de la suite des $(J(x_k))_{k \in \mathbb{N}}$:

Proposition 2 — 原函数值的收敛性

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction **convexe**, s.c.i. et propre. On suppose que J est **minorée**. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors la suite $(J(x_k))_{k \in \mathbb{N}}$ est décroissante et converge vers une valeur J^* . De plus, on a

$$J(x_{k+1}) \leq J(x_k) - \frac{1}{2\tau} \|x_{k+1} - x_k\|^2$$

De même, on va appliquer la proposition 5 du module **B2 : Méthodes du gradient explicite** sur la régularisation de MOREAU-YOSIDA ${}^\tau J$. On obtient alors le lemme suivant :

Lemme 2 — Moreau 包络函数的梯度 收敛到 0

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction **convexe**, **s.c.i.** et **propre**. On suppose que J est **minorée**. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors on a

$$\lim_{k \rightarrow +\infty} \|\nabla^\tau J(x_k)\| = \lim_{k \rightarrow +\infty} \|x_k - \text{prox}_{\tau J}(x_k)\| = \lim_{k \rightarrow +\infty} \|x_{k+1} - x_k\| = 0$$

Notons qu'il est possible de **démontrer** la **décroissance** de la **norme du sous-gradient** (ce qui est utile pour définir un critère d'arrêt) :

Proposition 3 — 原函数的梯度 收敛到 0

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction **convexe**, **s.c.i.** et **propre**. On suppose que J est **minorée**. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Pour tout $k \in \mathbb{N}$, on pose

$$p_{k+1} = \frac{1}{\tau} (x_k - x_{k+1})$$

Alors la suite $(\|p_k\|)_{k \in \mathbb{N}^*}$ est **décroissante** et **converge vers 0**.

DÉMONSTRATION : Soit $k \in \mathbb{N}^*$. On a d'après ce qui précède

$$p_k \in \partial J(x_k) \quad \text{et} \quad p_{k+1} \in \partial J(x_{k+1})$$

On déduit immédiatement du lemme précédent que la suite des p_k **converge vers 0**.

Démontrons maintenant la **monotonie** de la **suite** $(\|p_k\|)_{k \in \mathbb{N}}$. La **monotonie** du **sous-différentiel** d'une fonction **convexe** (démontrée à la proposition 6 du module **A2 : Sous-différentiabilité**) assure donc que

$$\langle p_{k+1} - p_k, p_{k+1} \rangle = -\frac{1}{\tau} \langle p_{k+1} - p_k, x_{k+1} - x_k \rangle \leq 0$$

Il s'ensuit que, en utilisant l'**inégalité** de **CAUCHY-SCHWARZ**,

$$\|p_{k+1}\|^2 \leq \langle p_k, p_{k+1} \rangle \leq \|p_k\| \cdot \|p_{k+1}\|$$

Ainsi, si p_{k+1} est non nul, on obtient le résultat désiré en simplifiant par $\|p_{k+1}\|$. Si $p_{k+1} = 0$, alors $x_k = x_{k+1}$ et on peut vérifier que, pour tout $n \geq k+1$, on a $p_n = 0$. Ainsi, l'affirmation reste vraie. ■

On va à présent exploiter la convexité de la régularisation de MOREAU-YOSIDA. On sait que, dans ce cas (lemme 1 du module **B2 : Méthodes du gradient explicite**), la suite des itérées x_k est bornée. Par ailleurs, la proposition 6 du module **B2 : Méthodes du gradient explicite** s'écrit

Proposition 4 — 生成函数列 收敛到 最小值点的函数值

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction **convexe**, **s.c.i.** et **propre**. On suppose que J admet un **minimiseur** x^* . Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors la suite $(J(x_k))_{k \in \mathbb{N}}$ converge vers $J(x^*)$ et

$$\forall k \in \mathbb{N}, \quad 0 \leq J(x_{k+1}) - J(x^*) \leq \frac{2 \|x_0 - x^*\|^2}{\tau} \frac{1}{k+1}$$

DÉMONSTRATION : Il suffit de noter que $\tau J(x_k) \geq J(x_{k+1})$. ■

Notons que l'on peut en réalité obtenir une **meilleure majoration** de l'erreur :

Proposition 5 (Convergence du critère vers le minimum)

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction **convexe**, **s.c.i.** et **propre**. On suppose que J admet un **minimiseur** x^* . Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors la suite $(J(x_k))_{k \in \mathbb{N}}$ converge vers $J(x^*)$ et

$$0 \leq J(x_k) - J(x^*) \leq \frac{\|x_0 - x^*\|^2}{2\tau} \frac{1}{k} - \frac{\|x_k - x^*\|^2}{2\tau} \frac{1}{k} - k \frac{\|x_{k-1} - x_k\|^2}{2\tau}$$

DÉMONSTRATION : Soit $k \in \mathbb{N}$. Posons

$$p_{k+1} = \frac{1}{\tau} (x_k - x_{k+1}) \in \partial J(x_{k+1})$$

Ainsi, la **convexité** de J assure que

$$\forall x \in \mathcal{X}, \quad J(x) - J(x_{k+1}) \geq \frac{1}{\tau} \langle x_k - x_{k+1}, x - x_{k+1} \rangle$$

Si $x = x^*$, alors, en utilisant une **identité remarquable**, on obtient que

$$J(x^*) - J(x_{k+1}) \geq \frac{1}{2\tau} \|x_k - x_{k+1}\|^2 - \frac{1}{2\tau} (\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2)$$

Si $x = x_k$, alors on a, en **multipliant par k** ,

$$k J(x_k) - (k+1) J(x_{k+1}) + J(x_{k+1}) \geq \frac{k}{\tau} \|x_k - x_{k+1}\|^2$$

Sommons les deux inégalités précédentes :

$$\begin{aligned} & J(x^*) + k J(x_k) - (k+1) J(x_{k+1}) \\ & \geq \frac{1+2k}{2\tau} \|x_k - x_{k+1}\|^2 - \frac{1}{2\tau} (\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2) \end{aligned}$$

Ainsi, en **sommant** cette inégalité pour k entre 0 et $K-1$, on obtient

$$K J(x^*) - K J(x_K) \geq \sum_{k=0}^{K-1} \frac{1+2k}{2\tau} \|x_k - x_{k+1}\|^2 - \frac{1}{2\tau} (\|x_0 - x^*\|^2 - \|x_K - x^*\|^2)$$

Pour minorer la somme, on utilise la décroissance de la suite des $\|x_k - x_{k+1}\|$ (proposition 3), qui assure que, pour tout $k \in \llbracket 0; K-1 \rrbracket$, les $\|x_k - x_{k+1}\|$ sont supérieurs à $\|x_{K-1} - x_K\|$. Par ailleurs, on a

$$\sum_{k=0}^{K-1} \frac{1+2k}{2\tau} = \sum_{k=0}^{K-1} \frac{1}{2\tau} + \sum_{k=0}^{K-1} \frac{k}{\tau} = \frac{K}{2\tau} + \frac{K(K-1)}{2\tau} = \frac{K^2}{2\tau}$$

Il suffit alors de combiner cette égalité avec l'inégalité précédente et de simplifier par $-K$ pour obtenir le résultat désiré. ■

Enfin, la proposition 7 du module **B2 : Méthodes du gradient explicite** devient (car τJ et J ont même minimiseurs) :

Proposition 6 生成点列 收敛到 最小值点

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe, s.c.i. et propre. On suppose que J admet un minimiseur. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors la suite $(x_k)_{k \in \mathbb{N}}$ converge vers un minimiseur de J .

2.2 Convergence dans le cas général 一般情况下的收敛性

Désormais, on ne suppose plus la convexité de J . Un premier résultat important est le fait que l'algorithme du point proximal est une méthode de descente, et que, de la même manière que pour la méthode du gradient explicite, une décroissance est assurée :

Proposition 7 (Convergence du critère) 原函数值的收敛性

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction s.c.i., de domaine non vide. On suppose que J est minorée. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} \in \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors la suite $(J(x_k))_{k \in \mathbb{N}}$ est décroissante et converge vers une valeur J^* . De plus, on a

$$J(x_{k+1}) \leq J(x_k) - \frac{1}{2\tau} \|x_{k+1} - x_k\|^2$$

DÉMONSTRATION : Soit $k \in \mathbb{N}$. Par définition d'un minimiseur et du point proximal, on a pour tout $x \in \mathcal{X}$

$$J(x_{k+1}) + \frac{1}{2\tau} \|x_{k+1} - x_k\|^2 \leq J(x) + \frac{1}{2\tau} \|x - x_k\|^2$$

En particulier, en prenant $x = x_k$, on en déduit la décroissance de la suite des $J(x_k)$, ainsi que l'inégalité annoncée. La fonction J étant supposée minorée, la suite $(J(x_k))_{k \in \mathbb{N}}$ est convergente. ■

On remarque qu'un sous-gradient de J au point courant s'exprime de manière simple à l'aide des itérées :

Lemme 3 次微分の显式表达

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction s.c.i., de domaine non vide. On suppose que J est minorée. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} \in \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors pour tout $k \in \mathbb{N}^*$,

$$\frac{1}{\tau} (x_{k-1} - x_k) \in \partial J(x_k)$$

DÉMONSTRATION : On applique la règle de FERMAT sur la définition de x_{k+1} :

$$0 \in \partial J(x_{k+1}) + \frac{1}{\tau} (x_{k+1} - x_k)$$

En réarrangeant les termes, on obtient le résultat attendu. ■

On peut alors en déduire le résultat suivant sur la convergence des sous-gradients :

Proposition 8 (Convergence du critère d'optimalité) 次微分列 收敛到 0

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction s.c.i., de domaine non vide. On suppose que J est minorée. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} \in \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Pour tout $k \in \mathbb{N}$, on pose

$$p_{k+1} = \frac{1}{\tau} (x_k - x_{k+1}) \in \partial J(x_{k+1})$$

Alors la suite $(p_k)_{k \in \mathbb{N}^*}$ converge vers 0.

DÉMONSTRATION : On applique la proposition 7 ce qui donne

$$\forall k \in \mathbb{N}, \quad \left\| \frac{1}{\tau} (x_k - x_{k+1}) \right\| \leq \frac{1}{\tau} (J(x_k) - J(x_{k+1}))$$

Puisque les suites $(J(x_k))_{k \in \mathbb{N}}$ et $(J(x_{k+1}))_{k \in \mathbb{N}}$ ont même limite, un passage à la limite permet de conclure par encadrement. ■

Pour démontrer la convergence des itérées, on va, comme pour la méthode du gradient explicite, démontrer ou supposer le caractère borné de la suite des itérées.

Proposition 9 生成点列 有界

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction coercive, s.c.i., de domaine non vide. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Alors la suite $(x_k)_{k \in \mathbb{N}}$ est bornée.

DÉMONSTRATION : Il s'agit d'une conséquence directe de la décroissance de la suite des $J(x_k)$ et du fait que l'ensemble de niveau inférieur $\text{niv}_{\leq J(x_0)} J$ est borné. ■

不凸, 但满足 KL 条件的

Lorsque J n'est pas convexe, mais KL (voir module **A3 : Propriété de KURDYKA–ŁOJASIEWICZ**), alors on a le résultat suivant :

Proposition 10 (Convergence des itérés dans le cas KL) (KL 情形) 生成点列 收敛到 最小值点

Soit $J : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction **coercive**. On suppose que le domaine de J est fermé et non vide, et que J est continue sur son domaine. Soit $(x_k)_{k \in \mathbb{N}}$ une suite générée par l'algorithme suivant

$$x_0 \in \mathcal{X} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = \text{prox}_{\tau J}(x_k)$$

avec $\tau > 0$. Soit x^* un point dans l'adhérence de la suite $(x_k)_{k \in \mathbb{N}}$. On suppose que J satisfait la propriété KL en x^* . Alors la suite $(x_k)_{k \in \mathbb{N}}$ converge vers x^* . De plus, x^* est un point critique de J .

DÉMONSTRATION : La convergence de la suite des itérées est une conséquence du théorème d'ATTOUCH, BOLTE & SVAITER (théorème 1 du module **A3 : Propriété de KURDYKA–ŁOJASIEWICZ**), où les trois conditions (C1)–(C3) sont respectivement données par les propositions [7] [8] et, pour la condition de continuité, par l'hypothèse sur la suite des x_k , qui admet une sous-suite $(J(x_{k_j}))_{j \in \mathbb{N}}$ de limite x^* parce qu'elle est bornée (proposition [9]), et par l'hypothèse sur J , qui assure que la suite des $J(x_{k_j})$ converge vers $J(x^*)$ car J est continue sur son domaine fermé. On utilise alors le fait que (proposition [8]) la suite des sous-gradients $(p_k)_{k \in \mathbb{N}}$ converge vers 0 et la fermeture du sous-différentiel pour en déduire que $0 \in \partial J(x^*)$. ■

On voit ainsi que remplacer le problème de minimisation initial par une suite de problèmes de minimisation (*a priori*) aussi complexes a un effet stabilisant. En effet, bien que le problème initial puisse avoir plusieurs solutions, sous des hypothèses relativement faibles (à savoir la convexité ou la coercivité et la propriété KL) sur la fonction objectif, l'algorithme du point proximal génère une suite qui converge vers un point unique.

3 Application

En pratique, il existe relativement peu de problèmes pour lesquels l'algorithme du point proximal constitue un progrès notable par rapport à la résolution de l'équation d'EULER. Autrement dit, résoudre l'inclusion

$$0 \in \partial J(x^*)$$

est généralement aussi complexe que d'évaluer l'opérateur proximal associé à J . L'intérêt véritable de l'algorithme du point proximal apparaîtra dans les modules suivants. On verra en effet de certains algorithmes d'optimisation peuvent être réécrits comme des itérations d'algorithme du point proximal. On utilisera alors les résultats démontrés dans ce module pour établir la convergence de ces algorithmes.

On va toutefois considérer un exemple ici, permettant d'illustrer (assez artificiellement) l'intérêt que peut avoir l'algorithme du point proximal. On s'intéresse au problème quadratique suivant :

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \langle Ax, x \rangle + \langle b, x \rangle + c \right\}$$

avec $A \in \mathcal{M}_{n,n}(\mathbb{R})$ une matrice, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$. La règle de FERMAT implique de résoudre le système linéaire suivant :

$$Ax = -b$$

Si, au lieu de résoudre ce système linéaire, on applique l'algorithme du point proximal, on est amené à résoudre le problème d'optimisation

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \langle Ax, x \rangle + \langle b, x \rangle + c + \frac{1}{2\tau} \|x - x_k\|_2^2 \right\}$$

ce qui revient à résoudre le système linéaire

$$Ax + b + \frac{1}{\tau} (x - x_k) = 0 \quad \text{soit} \quad \left(A + \frac{1}{\tau} I_n \right) x = \frac{1}{\tau} x_k - b$$

Quelques commentaires à propos de la convexité du problème : on peut vérifier que le problème initial est convexe si et seulement si A est semi-définie positive, c'est-à-dire si et seulement si

$$\forall x \in \mathbb{R}^n, \quad \langle Ax, x \rangle \geq 0$$

Dans ce cas, la matrice $A + I_n/\tau$ est inversible. Dans le cas général, il faut noter que, lorsque $\tau > 0$ est assez grand, $A + I_n/\tau$ est une matrice à diagonale strictement dominante, donc inversible. Par ailleurs, de telles matrices sont généralement bien conditionnées ; aussi, lorsque A est mal conditionnée, son inversion peut être numériquement instable. Un bon choix de pas de temps τ permet alors de transformer l'inversion d'une matrice mal conditionnée par une série d'inversions de matrices bien conditionnées.

Pour aller plus loin

Convergence d'algorithmes d'optimisation du premier ordre.

Comme on l'a vu dans ce module, l'algorithme du point proximal est en réalité d'usage assez limité dans les applications, car il existe relativement peu de problèmes intéressants pour lesquels l'opérateur proximal de la fonction objectif soit évaluable. Cependant, l'étude du comportement de cet algorithme est très utile, car on verra dans d'autres modules que certains algorithmes peuvent être interprétés comme des itérations du point proximal. Leur convergence découle alors de celle de l'algorithme du point proximal.