Outils d'optimisation pour les sciences des données et de la décision

M2 MIAGE, 2024-2025

Contrôle continu (Version du 29/11/2024)

Instructions

- Version courante : 29 novembre 2024.
  - 2024.11.29 : Précisions sur le rendu attendu.
  - 2024.11.18 : Première version.
- La version la plus récente du projet (mise à jour en cas de changement) est disponible à l'adresse **https://www.lamsade.dauphine.fr/∼croyer/ensdocs/ODD/ProjODD.pdf**.
- Le travail décrit ci-dessous peut être réalisé en binôme ou individuellement.
- Les réponses aux différentes questions ainsi que l'implémentation utilisée devront être envoyées au format ZIP à `clement.royer@lamsade.dauphine.fr`. Le nom de l'archive devra comporter le(s) nom(s) de chaque étudiant(e) impliqué(e) dans le projet.
- La date limite de rendu est fixée au **31 janvier 2025 AOE** (Anywhere On Earth).

Notations et remarques préliminaires

- Les dimensions des vecteurs ou matrices seront toujours supposées supérieures ou égales à 1.
- La notation $\|\cdot\|$ désignera la norme euclidienne.

Introduction

On s'intéresse dans ce projet à la résolution de problèmes d'optimisation non convexes de la forme

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w}), \tag{1}$$

où $f : \mathbb{R}^d \to \mathbb{R}$ est une fonction de classe $\mathcal{C}^2$. On sait que toute solution $\bar{\boldsymbol{w}} \in \mathbb{R}^d$ du problème vérifie $\nabla f(\bar{\boldsymbol{w}}) = \mathbf{0}$, mais l'inverse n'est pas forcément vraie. En revanche, pour de nombreux problèmes de sciences des données, on peut montrer que

$$\bar{\boldsymbol{w}} \in \underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{argmin}}\, f(\boldsymbol{w}) \quad \Leftrightarrow \quad \left[\nabla f(\bar{\boldsymbol{w}}) = \mathbf{0} \text{ et } \nabla^2 f(\bar{\boldsymbol{w}}) \succeq \mathbf{0}\right]. \tag{2}$$

On dit qu'un problème vérifiant (2) est **non convexe de manière bénigne**.

On cherchera donc dans la suite à développer des algorithmes qui convergent vers des points en lesquels les conditions d'ordre deux sont satisfaites de manière approchée. Plus précisément, on cherchera à atteindre un point $(\epsilon, \epsilon_{\mathrm{H}})$-optimal, dans le sens

$$\|\nabla f(\boldsymbol{w})\| \leq \epsilon \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(\boldsymbol{w})) \geq -\epsilon_{\mathrm{H}}. \tag{3}$$

Exemple d'illustration On considère le problème

$$\underset{\boldsymbol{w}\in\mathbb{R}^2}{\text{minimiser}}\, \varphi(\boldsymbol{w}) := (9\, w_1 - w_2)(11\, w_1 - w_2) + \frac{w_1^4}{2}. \tag{4}$$

Question 1 *Montrer qu'il existe trois vecteurs de $\mathbb{R}^2$ tels que le gradient de φ en ces points est nul.*

Question 2 *Donner les solutions du problème (4).*

Partie 1 : Algorithme déterministe

Algorithm 1: Algorithme d'ordre deux

1 **Initialisation**: Choisir $\boldsymbol{w}_0 \in \mathcal{F}$, fixer $\epsilon \in (0,1)$ et $\epsilon_{\mathrm{H}} \in (0,1)$.

2 **Pour** $k = 0, 1, \ldots$

1) Calculer $\nabla f(\boldsymbol{w}_k)$.

2) Si $\|\nabla f(\boldsymbol{w}_k)\| > \epsilon$, poser $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$, où $\alpha_k > 0$, et sauter les étapes suivantes. Sinon, aller en 3).

3) Calculer la plus petite valeur propre de la matrice hessienne $\lambda_k = \lambda_{\min}(\nabla^2 f(\boldsymbol{w}_k))$.

4) Si $\lambda_k < -\epsilon_{\mathrm{H}}$, poser $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \beta_k \boldsymbol{v}_k$, où $\beta_k > 0$ et $\boldsymbol{v}_k \in \mathbb{R}^d$ vérifie

$$\nabla f(\boldsymbol{w}_k)^{\mathrm{T}} \boldsymbol{v}_k \leq 0, \quad \nabla^2 f(\boldsymbol{w}_k) \boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k, \quad \|\boldsymbol{v}_k\| = 1.$$

5) Sinon, terminer et renvoyer $\boldsymbol{x}_k$.

3 **FinPour**

On travaillera sous l'hypothèse suivante, qui est plus forte que celle que l'on a utilisé durant le cours pour la méthode de gradient.

Hypothèse 1 *La fonction f est $\mathcal{C}_L^{1,1}$ et $\mathcal{C}_{L_{\mathrm{H}}}^{2,2}$.*

Sous l'hypothèse 1, on a pour tous $(\boldsymbol{x}, \boldsymbol{s}) \in (\mathbb{R}^n)^2$,

$$f(\boldsymbol{x} + \boldsymbol{s}) \quad \leq \quad f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{s} + \frac{L}{2} \|\boldsymbol{s}\|^2 \tag{5}$$

et

$$f(\boldsymbol{x} + \boldsymbol{s}) \quad \leq \quad f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{s} + \frac{1}{2} \boldsymbol{s}^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}) \boldsymbol{s} + \frac{L_{\mathrm{H}}}{6} \|\boldsymbol{s}\|^3. \tag{6}$$

On rappelle que sous l'hypothèse $f \in \mathcal{C}_L^{1,1}$, le choix $\alpha_k = \frac{1}{L}$ permet de garantir à la fois qu'un pas de descente de gradient réduit la valeur de fonction et garantit la convergence de la descente de gradient en $\mathcal{O}(\epsilon^{-2})$ itérations. L'idée de la question suivante est d'adapter cette théorie pour l'algorithme 1.

Question 3 *a) Supposons que l'itération k correspond à un pas de courbure négative (càd que $\lambda_k < -\epsilon_{\mathrm{H}}$ et $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \beta_k \boldsymbol{v}_k$). Montrer que si $\beta_k = \frac{1}{L_{\mathrm{H}}}$, alors*

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) - \frac{1}{3} |\lambda_k|^3 \leq f(\boldsymbol{x}_k) - \frac{1}{3} \epsilon_{\mathrm{H}}^3.$$

b) En déduire que l'algorithme atteint un point vérifiant (3) en au plus $\mathcal{O}(\max\{\epsilon^{-2}, \epsilon_{\mathrm{H}}^{-3}\})$ itérations.

c) Cette complexité est-elle meilleure que celle de la descente de gradient classique ? Justifier votre réponse.

Partie 2 : Algorithme probabiliste

On s'intéresse dans cette partie à un problème non convexe de la forme

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\ \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w}), \tag{7}$$

où chaque fonction $f_i : \mathbb{R}^d \to \mathbb{R}$ est de classe $\mathcal{C}^2$, et potentiellement non convexe.

Algorithm 2: Algorithme d'ordre deux

1 **Initialisation**: Choisir $\boldsymbol{w}_0 \in \mathcal{F}$, fixer $\epsilon \in (0,1)$ et $\epsilon_{\mathrm{H}} \in (0,1)$.

2 **Pour** $k = 0, 1, \ldots$

1) Tirer un indice i_k aléatoirement dans $\{1, \ldots, n\}$.

2) Calculer $\nabla f_{i_k}(\boldsymbol{w}_k)$.

3) Si $\|\nabla f_{i_k}(\boldsymbol{w}_k)\| > \epsilon$, poser $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f_{i_k}(\boldsymbol{w}_k)$, où $\alpha_k > 0$, et sauter les étapes suivantes. Sinon, aller en 3).

4) Calculer la plus petite valeur propre de la matrice hessienne $\lambda_k = \lambda_{\min}(\nabla^2 f_{i_k}(\boldsymbol{w}_k))$.

5) Si $\lambda_k < -\epsilon_{\mathrm{H}}$, poser $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \beta_k \boldsymbol{v}_k$, où $\beta_k > 0$ et $\boldsymbol{v}_k \in \mathbb{R}^d$ vérifie

$$\nabla^2 f_{i_k}(\boldsymbol{w}_k)\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k, \quad \|\boldsymbol{v}_k\| = |\lambda_k|.$$

3 **FinPour**

On s'intéressera à l'analyse de cet algorithme sous les hypothèses suivantes. L'hypothèse 2 correspond à celle faite dans le cadre d'une descente de gradient classique, tandis que la seconde est spécifique à l'utilisation des directions de courbure négative.

Hypothèse 2 *Pour tout indice i_k tel que $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f_{i_k}$, on a*

i) $\mathbb{E}_{i_k}\left[\nabla f_{i_k}(\boldsymbol{w}_k)\right] = \nabla f(\boldsymbol{w}_k)$;

ii) $\mathbb{E}_{i_k}\left[\|\nabla f_{i_k}(\boldsymbol{w}_k)\right] \leq \sigma$ *avec* $\sigma > 0$.

Hypothèse 3 *Pour tout indice i_k tel que $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \beta_k \boldsymbol{v}_k$, on a*

i) $\mathbb{E}_{i_k}\left[\nabla f(\boldsymbol{w}_k)^{\mathrm{T}} \boldsymbol{v}_k\right] = 0$;

ii) $\mathbb{E}_{i_k}\left[\boldsymbol{v}_k^{\mathrm{T}} \nabla^2 f(\boldsymbol{w}_k) \boldsymbol{v}_k\right] = \lambda_{\min}(\nabla^2 f(\boldsymbol{w}_k))^3$;

iii) $\mathbb{E}_{i_k}\left[\|\boldsymbol{v}_k\|^3\right] \leq \sigma_{\mathrm{H}}$ *avec* $\sigma_{\mathrm{H}} > 0$.

Question 4 *Sous les hypothèses 2 et 3, montrer que*

$$\mathbb{E}_{i_k}\left[\boldsymbol{w}_{k+1}\right] < f(\boldsymbol{w}_k) \tag{8}$$

si α_k et β_k sont choisis comme des valeurs constantes que l'on précisera.

Partie 3 : Implémentation

On s'intéresse dans cette partie à l'implémentation de la méthode étudiée plus haut, ainsi qu'à son application sur un problème de factorisation de matrice.

Question 5 *Implémenter l'algorithme 1 et l'appliquer au problème (4) en partant des points suivants :*

$$\boldsymbol{w}_{01} = \begin{bmatrix} 1 \\ 11 \end{bmatrix}, \ \boldsymbol{w}_{02} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \boldsymbol{w}_{03} = \begin{bmatrix} 10^{-6} \\ 10^{8} \end{bmatrix}.$$

Factorisation de matrice Étant donnée une matrice $\boldsymbol{M} \in \mathbb{R}^{d \times d}$, on dit que $\boldsymbol{M}$ est de rang 1 s'il existe $\boldsymbol{m} \in \mathbb{R}^d$ tel que $\boldsymbol{M} = \boldsymbol{m}[\boldsymbol{m}]^{\mathrm{T}}$.

Le problème de factorisation de matrice de rang 1 consiste à déterminer un vecteur $\boldsymbol{u}$ tel que $\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} \approx \boldsymbol{M}$ à partir d'observations de $\boldsymbol{M}$. Dans ce projet, on formulera cela via le problème

$$\underset{\boldsymbol{u} \in \mathbb{R}^d}{\text{minimiser}} \ f(\boldsymbol{u}) := \frac{1}{2d^2} \|\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} - \boldsymbol{M}\|_F^2 = \frac{1}{2d^2} \sum_{i=1}^{d} \sum_{j=1}^{d} (u_i\, u_j - M_{ij})^2. \tag{9}$$

Pour tous $(i, j) \in \{1, \ldots, d\}^2$, on pose $f_{ij}(\boldsymbol{u}) = \frac{1}{2}(u_i\, u_j - M_{ij})^2$. On a alors [1]

$$\nabla f_{ij}(\boldsymbol{u}) = u_j(u_i\, u_j - M_{ij})\boldsymbol{e}_i + u_i(u_i\, u_j - M_{ij})\boldsymbol{e}_j$$

et

$$\nabla^2 f_{ij}(\boldsymbol{u}) = u_j^2 \boldsymbol{E}_{ii} + (2u_i\, u_j - M_{ij})\boldsymbol{E}_{ij} + (2u_i\, u_j - M_{ij})\boldsymbol{E}_{ji} + u_i^2 \boldsymbol{E}_{jj}.$$

Question 6 *Générer une matrice de taille $d = 30$ de la forme $\boldsymbol{M} = \boldsymbol{m}\,\boldsymbol{m}^{\mathrm{T}}$, où $\boldsymbol{m} \in \mathbb{R}^{30}$ est un vecteur distribué suivant une loi normale $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$.*

Question 7 *Comparer la descente de gradient classique vue en cours à la descente de gradient avec courbure négative décrite dans l'algorithme 1 sur le problème implémenté dans la question 5.*

- *Pour un même budget d'itérations, quelle méthode obtient le point avec la meilleure valeur de fonction ?*
- *Combien l'algorithme 1 effectue-t-il de pas de courbure négative ? La méthode vous paraît-elle intéressante en pratique ?*

Question 8 *Comparer la méthode déterministe de l'algorithme 1 avec celle de l'algorithme 2 sur le problème implémenté dans la question précédente.*

- *Afficher une comparaison en nombres d'époques. Les conclusions sont-elles les mêmes que dans le cas de la descente de gradient et du gradient stochastique ?*
- *Utiliser plusieurs valeurs de ϵ et observer le comportement de l'algorithme 2 en fonction de cette valeur.*

[1] On note $\boldsymbol{e}_i$ le vecteur de $\mathbb{R}^d$ dont toutes les composantes sont nulles sauf la $i$ème égale à 1, et $\boldsymbol{E}_{ij} \in \mathbb{R}^{d \times d}$ la matrice dont seul le coefficient en position (i, j) est non nul, et égal à 1.