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

29 septembre 2023

Séance 2 :

→ Cours : Descente de gradient

→ TD : Notebook Python

DESCENTE DE GRADIENT (cf Section 2.1. polycopié)

(1) Introduction

Problème : $\min_{w \in \mathbb{R}^d} f(w)$ $\quad f : \mathbb{R}^d \to \mathbb{R}$

↳ On suppose que f est C^1 / de classe C^1, càd qu'il existe en tout point $w \in \mathbb{R}^d$ un vecteur $\nabla f(w) \in \mathbb{R}^d$ (gradient de f en w)

↳ $\nabla f(w)$ caractérise les variations de f au voisinage de w

En particulier,

Condition nécessaire d'optimalité à l'ordre 1 → $[w$ est un minimum local de $f] \Rightarrow [\nabla f(w) = 0_{\mathbb{R}^d}]$

ce qui est équivalent à

$[\nabla f(w) \neq 0_{\mathbb{R}^d}] \Rightarrow [w$ n'est pas un minimum local de $f]$

$\Rightarrow [\exists v$ proche de w tel que $f(v) < f(w)]$

Idée des méthodes dites de gradient

- Partant d'un point $w \in \mathbb{R}^d$ tel que $\nabla f(w) \neq 0_{\mathbb{R}^d}$, déterminer v tel que $f(v) < f(w)$
- Si $\nabla f(v) = 0$, terminer, sinon recommencer avec $w = v$

(2) Algorithme

Itération de descente de gradient (récursion qui définit l'algorithme)

$\forall k \in \mathbb{N}$, $w_{k+1} = w_k - \alpha_k \nabla f(w_k)$

- w_{k+1} : Itéré "suivant" (itération $k+1$) $\in \mathbb{R}^d$
- w_k : "Itéré courant à l'itération k" $w_k \in \mathbb{R}^d$
- $\nabla f(w_k)$: Gradient de f en $w_k \in \mathbb{R}^d$
- α_k : réel > 0 "Taille de pas" (stepsize)

$\Rightarrow w_{k+1}$ est obtenu en se déplaçant de w_k dans la direction opposée au gradient d'une taille de pas α_k.

Pseudo-code (pour l'algorithme de descente de gradient)

Initialisation : . $w_0 \in \mathbb{R}^d$, $k = 0$.
. Calculer $\nabla f(w_0) \in \mathbb{R}^d$.

Tant que [critère d'arrêt non satisfait]

- Calculer $\alpha_k > 0$.
- Définir $w_{k+1} = w_k - \alpha_k \nabla f(w_k)$
- Calculer $\nabla f(w_{k+1})$.
- $k \leftarrow k+1$

Fin Tantque

Critère d'arrêt :

↳ Se décompose typiquement en 2 conditions

1) Critère de convergence

Ex) $\|\nabla f(w_k)\| = 0$ (objectif des méthodes de gradient, souvent pas atteint en pratique)

$\|\nabla f(w_k)\| \leq \varepsilon$ avec $\varepsilon > 0$ une précision donnée (ex : $\varepsilon = 10^{-5}$)

$f(w_k) - \min_{w \in \mathbb{R}^d} f(w) \leq \varepsilon$ avec $\varepsilon > 0$

(⚠ Requiert de connaître la valeur optimale du problème)

$f(w_k) - f_{target} \leq \varepsilon$ $\varepsilon > 0$

$f_{target} \in \mathbb{R}$ approximation de la valeur optimale (ou valeur "cible")

2) Critère de budget

↳ Limite de coût à ne pas dépasser (garantir que l'algorithme va s'arrêter)

↳ Coût classiques

- Nombre d'itérations : $k \leq k_{max}$
- Nombre d'évaluations de f ou de ∇f
- Temps d'exécution (temps réel ou temps CPU)
- Coût mémoire

Choix de la taille de pas (ou longueur de pas) α_k

stepsize — steplength

$w_{k+1} = w_k - \alpha_k \nabla f(w_k)$ $\quad$ $w_{k+1} - w_k = -\alpha_k \nabla f(w_k)$: pas

$\alpha_k > 0$ requis pour faire de la descente de gradient (si $\alpha_k < 0$ montée de gradient, càd maximisation)

$\alpha_k = 0$ rien ne le pousse

3 grandes familles de choix de α_k

1) Taille de pas constante

$\forall k \in \mathbb{N}, \quad \alpha_k = \alpha > 0$ $\quad$ α choisie a priori (fait partie de l'initialisation)

⊕ Sous certaines hypothèses sur f, on peut montrer que ce choix fonctionne avec α suffisamment petit

⊕ Garantit que α ne tend pas vers 0 (numériquement !)

⊖ Toutes les valeurs de α ne fonctionnent pas

⊖ Choix indépendant de k (et donc de $f(w_k)$)

2) Tailles de pas décroissantes

Définir $\{\alpha_k\}_{k \in \mathbb{N}}$ a priori de sorte que $\alpha_k \xrightarrow[k \to +\infty]{} 0$

Ex : $\alpha_k = \frac{1}{k+1}$ $\forall k \in \mathbb{N}$ $\qquad \alpha_{k+1} \leq \alpha_k$

⊕ Pour k suffisamment grand, α_k sera toujours une "bonne" taille de pas

⊖ Après un certain nombre d'itérations, $\alpha_k \approx 0$ et donc l'algorithme peut stagner

⊖ Dépend de k mais pas de w_k

3) Tailles de pas adaptatives

Choisir α_k à chaque itération en fonction de w_k et de $\nabla f(w_k)$

⊕ Choix adapté à l'itération pour garantir (notamment) que $f(w_{k+1}) < f(w_k)$

⊖ Plus coûteux que des choix a priori

Exemple : Recherche linéaire type Armijo

- Point de départ : w_k, $\nabla f(w_k) \neq 0$. $c \in (0,1)$
- Poser $\alpha_k = 1$.
- Tant que $\left[f(w_k - \alpha_k \nabla f(w_k)) \geq f(w_k) - c\alpha_k \|\nabla f(w_k)\|^2 \right]$

  $\alpha_k \leftarrow \alpha_k / 2$

  Fin Tant Que

Valeurs testées : $\alpha_k = 1, \alpha_k = 1/2, 1/4, \ldots \longrightarrow 0$

On sort de la boucle avec α_k qui vérifie

$$f(w_k - \alpha_k \nabla f(w_k)) < f(w_k) - c\alpha_k \|\nabla f(w_k)\|^2 < f(w_k)$$

Garantie de décroissance de la fonction

NB : ③ très fréquent dans les logiciels d'optimisation

① ② plutôt utilisés en sciences des données

③ Bases théoriques de la descente de gradient

↳ On se place dans un cadre plus restreint que ci-dessus

$\min_{w \in \mathbb{R}^d} f(w)$ $\quad f : \mathbb{R}^d \to \mathbb{R}$ est $C^{1,1}_L$ (de classe $C^{1,1}_L$)

Def : f est $C^{1,1}_L$ pour $L > 0$ si

- f est C^1
- $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$, $w \mapsto \nabla f(w)$ est L-lipschitzienne, càd

$$\forall (v,w) \in (\mathbb{R}^d)^2, \quad \|\nabla f(v) - \nabla f(w)\| \leq L \|v - w\|$$

(L : constante de Lipschitz)

↳ Si f est $C^{1,1}_L$, cela donne des indications sur le choix de α_k et sur la convergence de l'algorithme de descente de gradient.

Théorème Supposons que f est $C^{1,1}_L$ et que l'on applique la descente de gradient à f. Soit w_k le k-ième itéré de la descente de gradient, on suppose que $\nabla f(w_k) \neq 0$. Alors :

a) Si $\alpha_k \in \left(0, \frac{2}{L}\right)$ $\left(0 < \alpha_k < \frac{2}{L}\right)$, alors

$$f(w_k - \alpha_k \nabla f(w_k)) \leq f(w_k)$$

⇒ Si α_k est suffisamment petit, c'est un "bon choix"

b) Si $\alpha_k = \frac{1}{L}$ (choix de taille de pas constante), alors

$$f(w_k - \alpha_k \nabla f(w_k)) < f(w_k) \underbrace{- \frac{1}{2L} \|\nabla f(w_k)\|^2}_{-c\alpha_k \|\nabla f(w_k)\|^2 \text{ avec } c = 1/2}$$

Remarques :

→ La condition de recherche linéaire est une approximation de la décroissance obtenue pour $\alpha_k = \frac{1}{L}$

→ Si L est connue, alors il est courant de choisir $\alpha_k = \frac{1}{L}$

→ Sinon, les différents choix visent à estimer cette constante