OUTILS D'OPTIMISATION POUR LES SDD

7 octobre 2024

Aujourd'hui :

CM Méthodes de gradient (+ illustration)

Vendredi : TD Exercices d'annales

MÉTHODES DE GRADIENT

↳ Classe d'algorithmes d'optimisation qui forme la base des méthodes utilisées aujourd'hui en sciences des données

⇒ Algorithme phare : la descente de gradient

(1) Descente de gradient

Cadre de travail : $\min_{w \in \mathbb{R}^d} f(w)$, avec $f : \mathbb{R}^d \to \mathbb{R}$ de classe C^1

($\forall w \in \mathbb{R}^d$, la dérivée de f en w existe et est représentée par le gradient $\nabla f(w) \in \mathbb{R}^d$)

On a vu que

$$\left[\bar{w} \in \underbrace{\operatorname*{argmin}_{w \in \mathbb{R}^d} f(w)}_{\text{ensemble des solutions du problème (minima globaux)}}\right] \Longrightarrow \left[\nabla f(\bar{w}) = 0_{\mathbb{R}^d}\right]$$

$$\nabla f(\bar{w}) = 0_{\mathbb{R}^d} \Leftrightarrow \|\nabla f(\bar{w})\| = 0$$

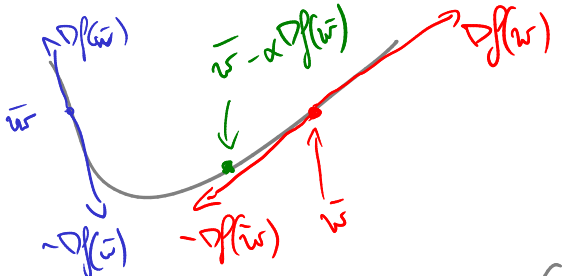
Cette implication s'écrit de manière équivalente

$$\left[\|\nabla f(\bar{w})\| \neq 0\right] \Longrightarrow \left[\bar{w} \notin \operatorname*{argmin}_{w \in \mathbb{R}^d} f(w)\right]$$

$$\Leftrightarrow \left[\exists\, w^+ \in \mathbb{R}^d,\ f(w^+) < f(\bar{w})\right]$$

On peut montrer que si $\|\nabla f(\bar{w})\| \neq 0$, alors il existe $w^+ \in \mathbb{R}^d$ de la forme $w^+ = \bar{w} - \alpha \nabla f(\bar{w})$ avec $\alpha > 0$ tel que $f(w^+) < f(\bar{w})$

⇒ On peut diminuer la valeur de f en bougeant dans la direction opposée au gradient !



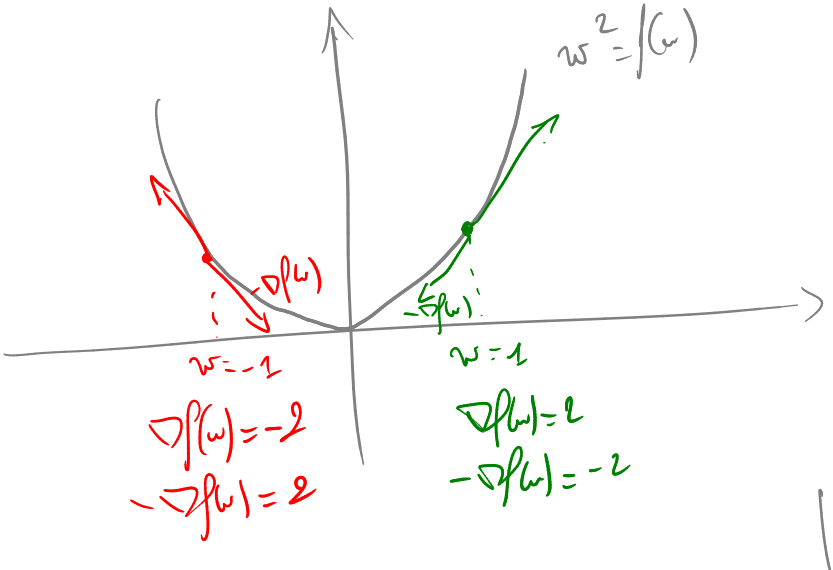
Remarques

- L'observation ci-dessus n'est valable qu'au voisinage de $\bar{w}$

$$f(w) \approx f(\bar{w}) + \nabla f(\bar{w})^T (w - \bar{w})$$

lorsque $\|w - \bar{w}\|$ est suffisamment petite)

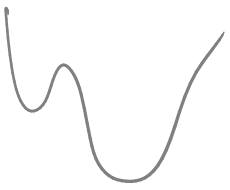
- On dit que la direction opposée au gradient est la direction de plus forte pente



$\nabla f(w) = 2w$

$\nabla f(w) > 0$ si $w > 0$

$\nabla f(w) < 0$ si $w < 0$



Algorithme de descente de gradient (pseudo-code)

Initialisation : Choisir $w_0 \in \mathbb{R}^d$ et $\alpha_0 > 0$

w_0 : point initial

α_0 : longueur de pas initiale

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

Calculer $w_{k+1} = w_k - \alpha_k \nabla f(w_k)$ ← Itération de la descente de gradient

Choisir $\alpha_{k+1} > 0$ ← longueur de pas de l'itération $k+1$

↳ Pour implémenter cet algorithme, on considère deux aspects fondamentaux :

- **Critère d'arrêt** : Souvent une combinaison de 2 critères :

1) Critère de convergence :

Ex) s'arrêter lorsque $\|\nabla f(w_k)\| < \varepsilon$ avec $\varepsilon > 0$

Ex) s'arrêter lorsque $f(w_k) - \min_{w \in \mathbb{R}^d} f(w) < \varepsilon$ avec $\varepsilon > 0$

↑ valeur optimale du problème

⚠ Demande de connaître la valeur optimale

Remarque : $< \varepsilon$ ou $\leq \varepsilon$ sont tous les deux utilisés en pratique

Ex) s'arrêter lorsque $\|w_k - w^*\| < \varepsilon$ avec $\varepsilon > 0$

où $w^* \in \operatorname{argmin}_{w \in \mathbb{R}^d} f(w)$

⚠ Un tel w^* n'est pas connu en général

2) Critère de budget

Ex) s'arrêter au bout de K itérations $(K \geq 1)$

Ex) s'arrêter au bout de N évaluations de f et/ou son gradient (peut différer du nombre d'itérations en fonction de la méthode utilisée)

Ex) s'arrêter au bout d'un certain temps { CPU / GPU ; tout court

- **Stratégie de choix de longueur de pas**

(En anglais, longueur de pas = stepsize / learning rate en ML)

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

Grandes familles de stratégies :

1) Longueur de pas constante : $\alpha_k = \alpha_0 = \alpha > 0 \quad \forall k \in \mathbb{N}$

⊕ Simple à calculer

⊕ Il existe des valeurs pour lesquelles l'algorithme va converger

⊕ Choix facile à calibrer pour un problème spécifique

⊖ Si α est mal choisi, l'algorithme peut converger très lentement, ou même diverger

⊖ Les valeurs théoriques pour lesquelles l'algorithme converge peuvent être impossibles à calculer en pratique

2) Longueur de pas décroissante : $\{\alpha_k\}_k \quad \alpha_k > 0$ $\alpha_k \searrow 0$

$\alpha_k \to 0$

$\alpha_{k+1} \leq \alpha_k \ \forall k$

⊕ Garantit que $f(w_{k+1}) < f(w_k)$ pour k suffisamment grand

⊕ Calculable avant de lancer l'algorithme (pas coûteux de calculer α_{k+1} à chaque itération)

⊖ Si $\{\alpha_k\}_k$ décroît trop rapidement, alors la méthode convergera trop lentement et produira des pas numériquement égaux à 0

⊖ Difficile de choisir la vitesse de décroissance adaptée à un problème donné a priori

3) Longueur de pas adaptative

↳ A chaque itération, on choisit α_k en fonction de w_k, $f(w_k)$, $\nabla f(w_k)$, càd en fonction de l'information disponible au point courant w_k

⊕ Produit en général de meilleures valeurs de α_k (en termes de décroissance de f) que des stratégies a priori

⊖ Coûte en général plus cher en appels à f et/ou à ∇f

Ex) Recherche linéaire avec retour arrière (backtracking) d'Armijo

par ex $\alpha = 1$ ⓞ Définir $\alpha > 0$.

ⓐ Partant de w_k, on calcule $f(w_k - \alpha \nabla f(w_k))$

par ex $c = 10^{-4}$ ⓑ Si $f(w_k - \alpha \nabla f(w_k)) < f(w_k) - c\alpha \|\nabla f(w_k)\|^2$ (où $c \in (0, \frac{1}{2})$)

↑ Condition d'Armijo dite de décroissance suffisante

alors poser $w_{k+1} = w_k - \alpha \nabla f(w_k)$

$\theta = \frac{1}{2}$ Sinon mettre à jour $\alpha \leftarrow \theta \alpha$ avec $\theta \in (0,1)$

et retourner en ⓐ

Exemple d'application : Moindres carrés linéaires

$$\underset{w \in \mathbb{R}^d}{\text{minimiser}} \; f(w) = \frac{1}{2n} \|Xw - y\|^2 \qquad X \in \mathbb{R}^{n \times d} \qquad y \in \mathbb{R}^n$$

↳ Pour ce problème, si $\nabla f(w) \neq 0_{\mathbb{R}^d}$, alors

$$f(w - \alpha \nabla f(w)) \leq f(w) - \frac{1}{2}\alpha \|\nabla f(w)\|^2$$

avec $\alpha = \frac{1}{L}$ et $L = \frac{\|X^T X\|}{n}$

② Analyse théorique

↳ On veut analyser l'algorithme de gradient et déterminer s'il converge ou non vers une solution du problème ⇒ Sous quelles conditions la méthode converge-t-elle ?

↳ Les résultats de convergence pour la descente de gradient sont toujours valides :
- pour une classe de problèmes (de fonctions f)
- pour un choix de longueur de pas (constante, décroissante, etc.)

$\Rightarrow$ Pour aujourd'hui, on regardera les fonctions C^1 à gradient Lipschitz et une façon de pas contrainte bien choisie.

Définition : Soit $f : \mathbb{R}^d \to \mathbb{R}$ une fonction de classe C^1.

On dit que f est C^1 à gradient L-lipschitzien avec $L > 0$ (ce que l'on note $f \in C^{1,1}_L$) si

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

Ex) $f(w) = \underbrace{\frac{1}{2n} \|Xw - y\|^2}_{\text{régression linéaire}}$ est $C^{1,1}_L$ avec $L = \frac{\|X^T X\|}{n}$

- Autres exemples : régression logistique, fonctions quadratiques, certains types de réseaux de neurones avec contraintes sur les paramètres

Remarque : Les résultats pour les fonctions $C^{1,1}_L$ s'adaptent à de nombreuses classes de fonctions

Proposition : Si f est $C^{1,1}_L$ (de $\mathbb{R}^d$ dans $\mathbb{R}$), alors

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

↑ Inégalité fondamentale pour les fonctions $C^{1,1}_L$ en optimisation

↳ L'inégalité fondamentale permet de trouver des conditions sur v qui garantissent que $f(v) < f(w)$ lorsque $\nabla f(w) \neq 0_{\mathbb{R}^d}$

Théorème : Soit $f : \mathbb{R}^d \to \mathbb{R}$ $C^{1,1}_L$ avec $L > 0$. Soit $w \in \mathbb{R}^d$ tel que $\nabla f(w) \neq 0_{\mathbb{R}^d}$. Alors

$$f\left(w - \frac{1}{L}\nabla f(w)\right) \leq f(w) - \frac{1}{2L}\|\nabla f(w)\|^2 < f(w)$$

Démo : On applique l'inégalité fondamentale avec $v = w - \frac{1}{L}\nabla f(w)$

$$f\left(w - \frac{1}{L}\nabla f(w)\right) \leq f(w) + \nabla f(w)^T\left(w - \frac{1}{L}\nabla f(w) - w\right) + \frac{L}{2}\|w - \frac{1}{L}\nabla f(w) - w\|^2$$

$$= f(w) + \nabla f(w)^T\left(-\frac{1}{L}\nabla f(w)\right) + \frac{L}{2}\|-\frac{1}{L}\nabla f(w)\|^2$$

$$= f(w) - \frac{1}{L}\nabla f(w)^T\nabla f(w) + \frac{L}{2} \times \left(-\frac{1}{L}\right)^2 \|\nabla f(w)\|^2$$

$\frac{L}{2} \times \left(-\frac{1}{L}\right)^2 = \frac{L}{2} \times \frac{1}{L^2} = \frac{1}{2L}$

$$= f(w) - \frac{1}{L}\nabla f(w)^T\nabla f(w) + \frac{1}{2L}\|\nabla f(w)\|^2$$

$\forall u \in \mathbb{R}^d,\ u^T u = \|u\|^2$

$$= f(w) - \frac{1}{L}\|\nabla f(w)\|^2 + \frac{1}{2L}\|\nabla f(w)\|^2$$

$$= f(w) - \frac{1}{2L}\|\nabla f(w)\|^2$$

Remarque : On peut montrer plus généralement que $\forall \alpha \in (0, \frac{2}{L})$,

$$f(w - \alpha\nabla f(w)) \leq f(w) - \frac{1}{2}\alpha\|\nabla f(w)\|^2 < f(w)$$

$\Rightarrow$ Lorsque la valeur de L est inconnue ou impossible à calculer, la recherche linéaire d'Armijo permet de satisfaire une condition de décroissance similaire

(cf plus haut : $f(w - \alpha\nabla f(w)) \leq f(w) - c\alpha\|\nabla f(w)\|^2$ avec $c \in (0, 1/2)$)

↳ Le théorème permet de garantir $f(w_{k+1}) < f(w_k)$ lorsque $\alpha_k = 1/L$

⇒ Propriété propre à l'itération k

Pour analyser l'algorithme en entier, on se donne un critère de convergence et on borne le nombre d'itérations nécessaire pour satisfaire ce critère ⇒ Borne de complexité

Théorème (Complexité de la descente de gradient dans le cas $C^{1,1}_L$)

On considère $\min_{w \in \mathbb{R}^d} f(w)$ avec $f : \mathbb{R}^d \to \mathbb{R}$ $C^{1,1}_L$.

On applique la descente de gradient en partant de $w_0 \in \mathbb{R}^d$ avec le pas $\alpha_k = \frac{1}{L}$ $\forall k \in \mathbb{N}$.

Alors, pour tout $\varepsilon > 0$, l'algorithme calcule w_k tel que $\|\nabla f(w_k)\| < \varepsilon$ en au plus $\lceil 2L\left(f(w_0) - \min_{w \in \mathbb{R}^d} f(w)\right) \varepsilon^{-2} \rceil$ itérations

$= O(\varepsilon^{-2})$ ← $O(\cdot)$: une constante qui ne dépend pas de ε

→ Borne de complexité : fonction de ε (précision souhaitée)

On s'intéresse à comment cette fonction augmente quand ε diminue

$O(\varepsilon^{-2}) = C\varepsilon^{-2}$

$\varepsilon = 10^{-2}$ → $10^4 C$ itérations

↳ $\varepsilon = 10^{-3}$ → $10^6 C$ itérations

divise la précision par 10 ⇒ multiplie le nombre d'itérations

→ On dira : "La complexité de la descente de gradient sur une fonction $C^{1,1}_L$ est en $O(\varepsilon^{-2})$"

(s'étend au-delà du choix $\alpha_k = \frac{1}{L}$)

Esquisse de preuve : On a vu précédemment que

$$f(w_{k+1}) = f\left(w_k - \frac{1}{L}\nabla f(w_k)\right) \leq f(w_k) - \frac{1}{2L}\|\nabla f(w_k)\|^2$$

Supposons que $\|\nabla f(w_k)\| \geq \varepsilon \quad \forall k = 0, \dots, K-1$

Alors, $\forall k = 0, \dots, K-1$, on a

$$f(w_{k+1}) \leq f(w_k) - \frac{1}{2L}\|\nabla f(w_k)\|^2 \leq f(w_k) - \frac{1}{2L}\varepsilon^2$$

d'où $$\frac{1}{2L}\varepsilon^2 \leq f(w_k) - f(w_{k+1}) \quad \forall k = 0, \dots, K-1$$

En sommant ces inégalités, on obtient

$$\sum_{k=0}^{K-1} \frac{1}{2L}\varepsilon^2 \leq \sum_{k=0}^{K-1} \left(f(w_k) - f(w_{k+1})\right)$$

$$\sum_{k=0}^{K-1} \frac{1}{2L}\varepsilon^2 = K\left(\frac{1}{2L}\varepsilon^2\right)$$

↑ Somme télescopique

$$f(w_0) - f(w_1) + f(w_1) - f(w_2) + \dots + f(w_{K-1}) - f(w_K)$$

$f(w_1) - f(w_1) = 0$, $f(w_{K-1}) - f(w_{K-1}) = 0$

$$\sum_{k=0}^{K-1}\left(f(w_k) - f(w_{k+1})\right) = f(w_0) - f(w_K)$$

$f(w_K) \geq \min_{w \in \mathbb{R}^d} f(w)$

$-f(w_K) \leq -\min_{w \in \mathbb{R}^d} f(w)$

Donc $$K\left(\frac{1}{2L}\varepsilon^2\right) \leq f(w_0) - f(w_K) \leq f(w_0) - \min_{w \in \mathbb{R}^d} f(w)$$

$$\Leftrightarrow \quad K \leq 2L\left(f(w_0) - \min_{w \in \mathbb{R}^d} f(w)\right)\varepsilon^{-2}$$

On vient de montrer que si $\|\nabla f(w_k)\| \geq \varepsilon \quad \forall k = 0, \dots, K-1$,

alors $K \leq 2L\left(f(w_0) - \min_{w \in \mathbb{R}^d} f(w)\right)\varepsilon^{-2}$

Par conséquent, si : $K \geq \left\lceil 2L\left(f(w_0) - \min_{w \in \mathbb{R}^d} f(w)\right)\varepsilon^{-2} \right\rceil$

(ΓA : plus petit entier supérieur ou égal à A)

il existe $k \in \{0, \dots, K-1\}$ tel que $\|\nabla f(w_k)\| < \varepsilon$.

↳ Lorsque la fonction f à minimiser est convexe en plus d'être $C^{1,1}_L$, on peut montrer des résultats plus forts sur la descente de gradient. avec longueur de pas $\alpha_k = \frac{1}{L}$ (l'algorithme ne change pas par rapport au théorème précédent !)

Propriétés de f	Critère de convergence	Borne de complexité
f $C^{1,1}_L$	$\|\nabla f(w_k)\| < \varepsilon$	$O(\varepsilon^{-2})$ itérations
f $C^{1,1}_L$ + convexe	$f(w_k) - \min_{w \in \mathbb{R}^d} f(w) < \varepsilon$	$O(\varepsilon^{-1})$ itérations
f $C^{1,1}_L$ + μ-fortement convexe	$\|w_k - w^*\| < \varepsilon$ avec $w^* \in \operatorname{argmin}_{w \in \mathbb{R}^d} f(w)$	$O\left(\frac{L}{\mu} \ln(\varepsilon^{-1})\right)$ itérations $= O(\ln(\varepsilon^{-1}))$ itérations

---

↳ Interprétation des bornes : Pour le même algorithme, on obtient de meilleures garanties (critère + borne de complexité) lorsque la fonction est fortement convexe que lorsqu'elle est convexe, et de meilleures garanties lorsque la fonction est convexe que lorsqu'elle ne l'est pas.

⇒ Pour cette raison, on dit qu'il est plus facile d'optimiser des fonctions fortement convexes que des fonctions convexes (et que des fonctions non convexes !)

Def: Une fonction $f: \mathbb{R}^d \to \mathbb{R}$ C^1 est dite μ-fortement convexe avec $\mu > 0$ si

$$\forall (v,w) \in (\mathbb{R}^d)^2, \quad f(v) \geq f(w) + \nabla f(w)^T (v-w) + \frac{\mu}{2}\|v-w\|^2$$

↳ Autre inégalité fondamentale

↳ Si f est $C^{1,1}_L$ et μ-fortement convexe, on a :

$$f(w) + \nabla f(w)^T(v-w) + \frac{L}{2}\|v-w\|^2 \geq f(v) \geq f(w) + \nabla f(w)^T(v-w) + \frac{\mu}{2}\|v-w\|^2$$

⇒ Les deux inégalités sont utiles pour montrer que la descente de gradient vérifie $\|w_k - w^*\| < \varepsilon$ en $O\left(\frac{L}{\mu}\ln(\varepsilon^{-1})\right)$ itérations

Propriété importante : Si f C^1 et μ-fortement convexe, alors elle possède un unique minimum global, qui est l'unique point $w^* \in \mathbb{R}^d$ tel que $\|\nabla f(w^*)\| = 0$

On note alors $w^* = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} f(w)$

Fortement convexe

Convexe par fortement convexe

Différence entre convexe et fortement convexe

- Toute fonction fortement convexe est convexe.
- L'inverse n'est pas vraie (prendre f constante)
- Une fonction convexe vérifie

$$f(v) \geq f(w) + \nabla f(w)^T(v-w) \quad \forall (v,w)$$

($\approx \mu = 0$)

- Une fonction convexe peut avoir plusieurs minima globaux