Mini-Projet d'Optimisation ML

Modélisation, SGD et Méthodes Proximales

Auteur :
DIENG BABAKAR

Date :
January 11, 2026

Faculté des sciences et techniques (FST)

PHASE 1

Justifier que F est de classe C2 , convexe, et -fortement convexe

1) F est de classe C^2 - La fonction scalaire

$$\phi(t) = \log\left(1 + e^{-t}\right)$$

est de classe C^∞ sur R (composition de fonctions exponentielle et logarithme, toutes deux C^∞). - Pour chaque i, la fonction

$$w \mapsto -y_i x_i^\top w$$

est affine donc C^∞. - Par composition,

$$w \mapsto \log\left(1 + e^{-y_i x_i^\top w}\right)$$

est de classe C^∞ sur R^d. - Le terme de régularisation $\frac{\lambda}{2}\|w\|^2$ est un polynôme quadratique, donc C^∞.

La somme et la moyenne de fonctions C^2 étant encore C^2, on conclut que

$$F \in C^2\left(R^d\right).$$

2) Convexité de F (a) Convexité de la perte logistique

On calcule les dérivées de ϕ :

$$\phi'(t) = -\frac{1}{1+e^t}, \quad \phi''(t) = \frac{e^t}{(1+e^t)^2} \geq 0 \quad \forall t \in R$$

Donc ϕ est convexe sur R.

La composition d'une fonction convexe avec une application affine reste convexe, donc

$$w \mapsto \log\left(1 + e^{-y_i x_i^\top w}\right)$$

est convexe pour chaque i.

La moyenne de fonctions convexes est convexe, donc

$$w \mapsto \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i x_i^\top w}\right)$$

est convexe.

(b) Convexité du terme Ridge

La fonction $w \mapsto \frac{\lambda}{2}\|w\|^2$ est convexe (même strictement convexe) pour $\lambda > 0$. Somme de fonctions convexes F est convexe.

3) λ-forte convexité

Calculons la hessienne de F. - Pour la perte logistique:

$$\nabla^2\left(\log\left(1 + e^{-y_i x_i^\top w}\right)\right) = \sigma_i(w)\left(1 - \sigma_i(w)\right) x_i x_i^\top,$$

où

$$\sigma_i(w) = \frac{1}{1 + e^{y_i x_i^\top w}} \in (0, 1).$$

Donc cette hessienne est semi-définie positive. - Pour la régularisation Ridge :

$$\nabla^2\left(\frac{\lambda}{2}\|w\|^2\right) = \lambda I_d.$$

Ainsi,

$$\nabla^2 F(w) = \frac{1}{n}\sum_{i=1}^{n} \sigma_i(w)\left(1 - \sigma_i(w)\right) x_i x_i^\top + \lambda I_d \succeq \lambda I_d \quad \forall w.$$

Par définition, cela signifie que F est λ-fortement convexe.

F est λ-fortement convexe.

Calcul du gradient $\nabla F(w)$

Posons, pour chaque i,

$$\ell_i(w) = \log\left(1 + e^{-y_i x_i^\top w}\right).$$

On a

$$\nabla \ell_i(w) = \frac{e^{-y_i x_i^\top w}}{1 + e^{-y_i x_i^\top w}} (-y_i x_i) = -\frac{y_i x_i}{1 + e^{y_i x_i^\top w}}.$$

Donc

$$\nabla F(w) = -\frac{1}{n} \sum_{i=1}^{n} \frac{y_i x_i}{1 + e^{y_i x_i^T w}} + \lambda w.$$

Forme matricielle : soit $X \in R^{n \times d}$ la matrice des données (ligne $i = x_i^\top$), $y = (y_1, \ldots, y_n)^\top$, et

$$s(w)_i = \frac{1}{1 + e^{y_i x_i^T w}}.$$

Alors

$$\nabla F(w) = -\frac{1}{n} X^\top (y \odot s(w)) + \lambda w,$$

où © désigne le produit terme à terme.

2) Hessienne de F

Pour chaque i,

$$\nabla^2 \ell_i(w) = \sigma_i(w)\,(1 - \sigma_i(w))\, x_i x_i^\top, \quad \sigma_i(w) = \frac{1}{1 + e^{y_i x_i^\top w}}$$

Or

$$0 < \sigma_i(w)\,(1 - \sigma_i(w)) \leq \frac{1}{4}$$

Ainsi

$$\nabla^2 F(w) = \frac{1}{n}\sum_{i=1}^{n} \sigma_i(w)\left(1-\sigma_i(w)\right) x_i x_i^\top + \lambda I_d$$

On rappelle qu'une fonction F est à gradient L-Lipschitzien si

$$\|\nabla F(u) - \nabla F(v)\| \leq L\|u-v\|, \quad \forall u, v \in R^d,$$

ce qui est équivalent à

$$\nabla^2 F(w) \preceq L I_d, \quad \forall w \in R^d.$$

La hessienne de F s'écrit :

$$\nabla^2 F(w) = \frac{1}{n}\sum_{i=1}^{n} \sigma_i(w)(1-\sigma_i(w))\, x_i x_i^\top + \lambda I_d,$$

où

$$\sigma_i(w) = \frac{1}{1+e^{y_i x_i^\top w}}.$$

Or, pour tout w et tout i, on a :

$$0 < \sigma_i(w)(1-\sigma_i(w)) \leq \frac{1}{4}.$$

Il vient alors :

$$\nabla^2 F(w) \preceq \frac{1}{4n}\sum_{i=1}^{n} x_i x_i^\top + \lambda I_d = \frac{1}{4n} X^\top X + \lambda I_d.$$

En prenant la plus grande valeur propre, on obtient une constante de Lipschitz pour le gradient :

$$L = \frac{1}{4n}\lambda_{\max}(X^\top X) + \lambda = \frac{1}{4n}\|X\|_2^2 + \lambda.$$

Conclusion

Le gradient de la fonction de perte logistique régularisée est L-Lipschitzien avec

$$L = \frac{1}{4n}\|X\|_2^2 + \lambda.$$

Codage de la Descente de Gradient à pas fixe et comparer avec la Méthode du Gradient Conjugué

voir notebook

PHASE 2

Cette partie est intégralement faite sur notebook.

Voici quelques interprétation:

Sur les premières époques d'entraînement, Adam converge plus rapidement que RMSProp grâce à l'utilisation conjointe du momentum et de l'adaptation du pas d'apprentissage. RMSProp reste néanmoins efficace et stable, mais montre une décroissance de la fonction objectif légèrement plus lente.

Amélioration de la stabilité

1. Réduction des oscillations

   Dans les problèmes mal conditionnés (vallées étroites et allongées), la descente de gradient classique oscille fortement. Le momentum :

   amortit les variations rapides du gradient,

   réduit les zigzags perpendiculaires à la direction optimale.

   Les itérations deviennent plus régulières et plus stables.

2. Filtrage du bruit (cas stochastique)

   En optimisation stochastique (mini-batchs) :

   les gradients sont bruités,

   le momentum agit comme un filtre passe-bas.

   Il réduit la variance des mises à jour, ce qui améliore la stabilité numérique.

Coclusion

Le momentum améliore significativement la stabilité des itérations en réduisant les oscillations et le bruit, mais nécessite un réglage approprié pour éviter l'instabilité.

PHASE 3

Cette également partie est intégralement faite sur notebook
Voici quelques conclusion: Sur les premières itérations, FISTA :

- descend beaucoup plus vite,
- atteint une solution de bonne qualité très tôt,
- conserve le même coût par itération que ISTA.

La version accélérée FISTA améliore significativement la vitesse de convergence de ISTA, sans coût computationnel supplémentaire par itération. Le gain est particulièrement marqué durant les premières itérations, ce qui fait de FISTA une méthode de choix pour les problèmes de grande dimension avec pénalisation 1