Université de Nouakchott

Faculté des Sciences et Techniques

Département de Mathématiques et Informatiques

Rapport de Projet

Optimisation pour la science de données

Modélisation, SGD et Méthodes Proximales

Réalisé par :
Boudah Mohamed Lemine Ahmedou El Mokhtar
C20121

Encadré par :
Dr.Mohamed Mahmoud El Bennany

Année universitaire : 2025 – 2026

Projet réalisé dans le cadre du Master SSD – Statistiques et Sciences des Données

Table des matières

Introduction

L'optimisation joue un rôle central en apprentissage automatique, dans la mesure où la plupart des modèles peuvent être formulés comme des problèmes de minimisation d'une fonction objectif dépendant des données. La compréhension des propriétés mathématiques de ces fonctions ainsi que des algorithmes permettant de les minimiser efficacement est donc essentielle, tant d'un point de vue théorique que pratique.

Ce mini-projet s'inscrit dans le cadre du cours d'Optimisation pour le Machine Learning et a pour objectif d'étudier, analyser et implémenter différentes méthodes d'optimisation appliquées à un problème de classification binaire basé sur la régression logistique. L'accent est mis sur le lien étroit entre la formulation mathématique du problème, les garanties théoriques de convergence et le comportement numérique des algorithmes.

Le projet est structuré en trois phases principales. La première phase est consacrée à l'étude d'un problème d'optimisation convexe lisse régularisé par une norme ℓ_2, et à l'analyse des méthodes de gradient déterministes. La deuxième phase aborde le passage à l'échelle dans un contexte de grandes données à travers les algorithmes de gradient stochastique et leurs variantes modernes. Enfin, la troisième phase traite des problèmes d'optimisation non lisses induits par une régularisation ℓ_1, et introduit les méthodes proximales telles que ISTA et FISTA pour la promotion de la parcimonie.

L'ensemble des développements théoriques est accompagné d'implémentations numériques en Python, permettant d'illustrer et de comparer les performances des différentes méthodes étudiées.

Phase 1 : Fondements et Gradient Déterministe

1. Analyse

On considère la fonction objectif définie par

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} \log\big(1 + e^{-y_i x_i^\top w}\big) + \frac{\lambda}{2}\|w\|_2^2,$$

où $x_i \in R^d$, $y_i \in \{-1, 1\}$ et $\lambda > 0$.

Régularité. La fonction $\varphi(t) = \log(1+e^{-t})$ est de classe C^∞ sur R. Comme $w \mapsto y_i x_i^\top w$ est une fonction affine, la composition $\varphi(y_i x_i^\top w)$ est de classe C^∞ sur R^d. La somme finie de fonctions de classe C^2 ainsi que le terme quadratique $\frac{\lambda}{2}\|w\|_2^2$, qui est également de classe C^2, impliquent que la fonction F est de classe C^2.

Convexité. La fonction φ est convexe car sa dérivée seconde est positive sur R. Par conséquent, la fonction $w \mapsto \log(1 + e^{-y_i x_i^\top w})$ est convexe comme composition d'une fonction convexe avec une application affine. La somme de fonctions convexes étant convexe, et le terme $\frac{\lambda}{2}\|w\|_2^2$ étant également convexe, on en déduit que F est une fonction convexe.

Forte convexité. La fonction quadratique $\frac{\lambda}{2}\|w\|_2^2$ est λ-fortement convexe sur R^d. La somme d'une fonction convexe et d'une fonction λ-fortement convexe étant λ-fortement convexe, il en résulte que la fonction F est λ-fortement convexe. Cette propriété garantit en particulier l'unicité du minimiseur global de F.

2. Gradient et Lipschitz

Calcul du gradient. La fonction objectif est

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i x_i^\top w}) + \frac{\lambda}{2}\|w\|_2^2.$$

La dérivée de $\varphi(t) = \log(1 + e^{-t})$ est

$$\varphi'(t) = -\frac{1}{1 + e^t} = -\sigma(-t),$$

où $\sigma(t) = \frac{1}{1+e^{-t}}$ est la fonction sigmoïde.

En utilisant la règle de la chaîne, on obtient

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

—

Lipschitzianité du gradient. La dérivée seconde de $\varphi(t)$ est

$$\varphi''(t) = \sigma(t)(1-\sigma(t)) \leq \frac{1}{4} \quad \forall t \in R.$$

La Hessienne de F est donc

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

Puisque $\sigma(t)(1-\sigma(t)) \leq 1/4$, on peut borner la norme spectrale de la Hessienne par

$$\|\nabla^2 F(w)\| \leq \frac{1}{4n}\|X\|_2^2 + \lambda,$$

où $\|X\|_2$ est la norme spectrale (ou valeur singulière maximale) de la matrice des données $X \in R^{n\times d}$.

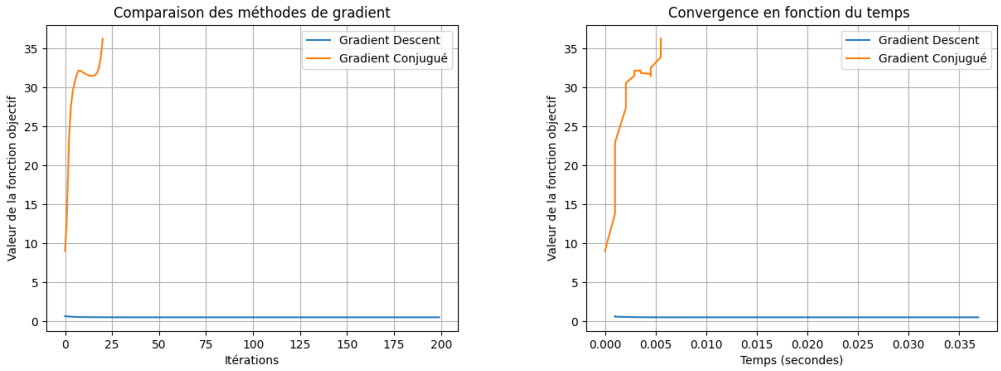
Ainsi, le gradient est L-Lipschitzien avec

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

3. Implémentation et comparaison des méthodes

Dans cette partie, nous avons implémenté numériquement la descente de gradient à pas fixe ainsi que la méthode du gradient conjugué afin de résoudre le problème de régression logistique régularisée.

l'implémentation est dans le notebook. et les courbes sont sous-desous :



Phase 2 : Passage à l'Échelle Stochastique (Chap. 3)

1. Descente de gradient stochastique

Descente de Gradient Stochastique Lorsque la taille de l'échantillon n devient très grande, le calcul du gradient complet de la fonction objectif à chaque itération devient coûteux. La descente de gradient stochastique (SGD) constitue alors une alternative efficace, en approximant le gradient à partir d'une seule observation choisie aléatoirement.

À l'itération k, la mise à jour s'écrit :

$$w_{k+1} = w_k - \alpha_k \nabla \ell(w_k; x_i, y_i),$$

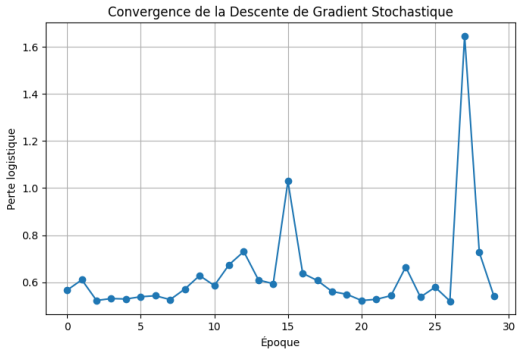
où (x_i, y_i) est un échantillon tiré aléatoirement et α_k est un pas d'apprentissage décroissant.

Afin d'assurer la convergence, nous avons utilisé une règle de décroissance du pas de la forme :

$$\alpha_k = \frac{\alpha_0}{1+k}.$$

Les résultats expérimentaux montrent que la SGD converge rapidement lors des premières époques, bien que la fonction de perte présente des oscillations dues à l'approximation bruitée du gradient. Ce comportement est typique des méthodes stochastiques et contraste avec la décroissance plus régulière observée dans la descente de gradient déterministe.

l'implémentation est dans le notebook. et le courbe convergence est sous-desous :



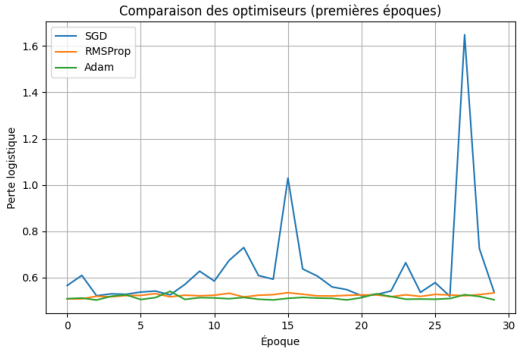
2. Optimiseurs Modernes : RMSProp et Adam

Afin d'améliorer la convergence des algorithmes stochastiques, nous avons implémenté deux optimiseurs modernes : RMSProp et Adam.

RMSProp adapte le pas d'apprentissage pour chaque coordonnée en fonction de la moyenne exponentielle des gradients au carré. Cette approche permet de réduire les oscillations typiques de la SGD et d'obtenir une convergence plus stable lors des premières époques.

Adam combine l'effet du momentum avec la normalisation des gradients à la RMSProp. Ainsi, il bénéficie à la fois de la réduction du bruit et d'une accélération de la convergence, en particulier au début de l'entraînement.

Les expérimentations montrent que, comparé à la SGD classique, RMSProp offre une convergence plus stable, alors qu'Adam atteint plus rapidement un minimum local. Ces résultats illustrent l'efficacité des optimiseurs adaptatifs sur les problèmes de régression logistique stochastique. l'implémentation est dans le notebook. et le courbe convergence est sous-desous :



3. Impact du Momentum sur la Stabilité

Le momentum est une technique qui permet d'accélérer la convergence des algorithmes de gradient stochastique tout en réduisant les oscillations autour du minimum. Au lieu de suivre uniquement le gradient courant, on conserve une fraction β de la direction précédente, ce qui donne une mise à jour de la forme :

$$v_{k+1} = \beta v_k + \alpha \nabla \ell(w_k), \quad w_{k+1} = w_k - v_{k+1}.$$

L'effet du momentum est double : il stabilise les trajectoires du gradient stochastique et permet de prendre des pas plus importants sans divergence. Les optimiseurs modernes comme RMSProp et Adam utilisent implicitement le momentum, ce qui explique la convergence plus rapide et moins bruitée observée dans nos expérimentations, comparée à la SGD classique.

Phase 3 : Non-Lissé, Parcimonie et Proximal (Chap. 4)

1. Problème non lissé et opérateur proximal

Lorsque l'on remplace la régularisation L2 par la norme L1, la fonction objectif devient :

$$\Phi(w) = f(w) + \lambda \|w\|_1,$$

où f est la perte logistique.

La norme L1 est non lisse car elle n'est pas différentiable en $w_i = 0$. Pour traiter ce problème, on utilise les méthodes proximales, qui remplacent le gradient classique par un opérateur proximal.

L'opérateur proximal associé à la norme L1 est donné par le soft-thresholding :

$$\text{prox}_\lambda(v_i) = \begin{cases} v_i - \lambda & \text{si } v_i > \lambda, \\ 0 & \text{si } |v_i| \leq \lambda, \\ v_i + \lambda & \text{si } v_i < -\lambda. \end{cases}$$

Cet opérateur permet de gérer les non-lissages tout en favorisant la parcimonie des coefficients, c'est-à-dire en annulant certains w_i et en sélectionnant les variables pertinentes.

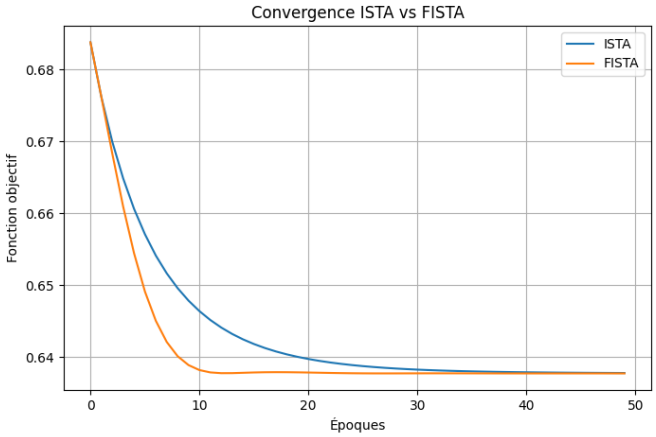
2. Implémentation ISTA et FISTA

Pour résoudre le problème de régression logistique avec régularisation L1, nous avons implémenté l'algorithme ISTA (Iterative Soft-Thresholding Algorithm) et sa version accélérée FISTA.

ISTA combine une étape de gradient sur la perte logistique avec l'opérateur proximal associé à la norme L1, c'est-à-dire le soft-thresholding. Cette approche permet de traiter la non-lissité de la fonction objectif et de favoriser la parcimonie des coefficients.

FISTA améliore ISTA en ajoutant une extrapolation qui accélère la convergence, réduisant ainsi le nombre d'époques nécessaires pour atteindre un minimum comparable. Les résultats expérimentaux confirment que FISTA converge plus rapidement qu'ISTA tout en maintenant la stabilité.

L'analyse de la parcimonie montre que plusieurs coefficients deviennent exactement nuls, ce qui permet une sélection efficace des variables. Ce phénomène est directement lié à la valeur du paramètre de régularisation λ : plus λ est grand, plus le nombre de coefficients nuls augmente. l'implémentation est dans le notebook. et le courbe convergence est sous-desous :

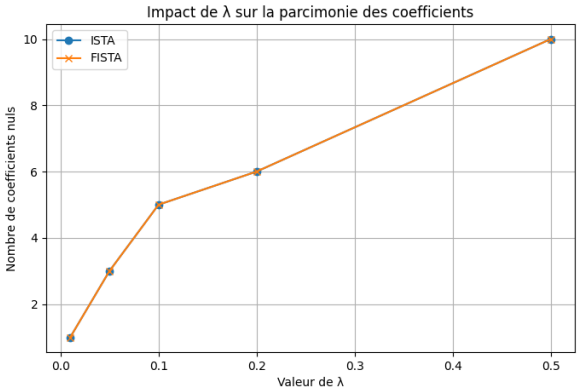


3. Analyse de la parcimonie en fonction de λ

Pour mieux comprendre l'effet de la régularisation L1 sur la sélection des variables, nous avons étudié le nombre de coefficients nuls dans w^* en fonction de différentes valeurs de λ.

Les expériences montrent que lorsque λ augmente, le nombre de coefficients nuls croît également. Cela illustre parfaitement le rôle de la régularisation L1 : elle favorise la parcimonie en annulant les coefficients moins significatifs et permet ainsi une sélection automatique des variables pertinentes.

Le phénomène est observable aussi bien avec ISTA qu'avec FISTA, la principale différence étant que FISTA converge plus rapidement vers la solution finale. Les résultats expérimentaux sont visualisés dans le notebook par un graphique du nombre de coefficients nuls en fonction de λ.



Conclusion

Dans ce mini-projet, nous avons exploré différentes méthodes d'optimisation pour la régression logistique, en commençant par la descente de gradient déterministe, puis en passant aux méthodes stochastiques telles que SGD, RMSProp et Adam. Enfin, nous avons étudié les problèmes non lissés avec régularisation L1, en implémentant ISTA et FISTA.

Les résultats expérimentaux ont montré que :

- La descente de gradient stochastique est rapide mais présente des oscillations, stabilisées par le momentum.
- Les optimiseurs adaptatifs RMSProp et Adam améliorent la vitesse et la stabilité de convergence.
- La régularisation L1 favorise la parcimonie et la sélection automatique des variables, avec un effet plus prononcé lorsque λ augmente.
- FISTA accélère la convergence par rapport à ISTA tout en conservant la stabilité.

Ce projet a permis de mettre en pratique les concepts théoriques vus en cours, de comprendre le rôle de la régularisation et de comparer l'efficacité des différentes méthodes d'optimisation pour des problèmes réels de classification.