Université de Nouakchott Al Aasriya
Faculté des Sciences et Techniques
Département Mathématiques et Informatiques

Mini-Projet d'Optimisation pour l'Apprentissage Automatique

Modélisation, SGD et Méthodes Proximales

Réalisé par :

Mohamed Lemine Abdallahi Tah C12896

Encadré par :
Dr El Benany Mohamed Mahmoud

Année universitaire : 2024 – 2025

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

Table des matières

Note

Ce mini-projet porte sur le thème : « *Modélisation, Gradient Stochastique (SGD) et Méthodes Proximales pour l'Optimisation en Apprentissage Automatique* ».

Les objectifs principaux sont les suivants :
- Étudier les propriétés mathématiques de la perte logistique régularisée (convexité, Lipschitzianité du gradient, forte convexité).
- Implémenter et comparer les méthodes déterministes (Descente de Gradient, Gradient Conjugué).
- Étudier les algorithmes stochastiques (SGD avec pas décroissant, RMSProp, Adam) et analyser l'impact du momentum.
- Passer à la régularisation L1 (parcimonie) et implémenter les méthodes proximales ISTA et FISTA.
- Utiliser le dataset Breast Cancer (scikit-learn) après normalisation des features.

Note

Le code source complet (notebook Jupyter avec toutes les implémentations), les figures générées et les résultats sont disponibles dans le dépôt GitHub suivant : `https://github.com/MoLemine/Optimization`

Le fichier principal est **OptimisationML.ipynb**. Les figures utilisées dans le rapport se trouvent dans le dossier **figures/**.

Introduction et Contexte

Ce mini-projet vise à mettre en pratique les principaux concepts du cours d'Optimisation pour l'Apprentissage Automatique : formalisation mathématique (Chapitre 1), descente de gradient déterministe (Chapitre 2), gradient stochastique et optimiseurs modernes (Chapitre 3), méthodes proximales pour la régularisation non lisse (Chapitre 4).

Nous étudions le problème de classification binaire sur le dataset *Breast Cancer Wisconsin* (scikit-learn) : 569 exemples, 30 caractéristiques numériques, labels convertis en $\{-1, +1\}$.

Pour garantir la stabilité numérique et accélérer la convergence, toutes les caractéristiques ont été normalisées via `StandardScaler` (moyenne 0, écart-type 1). Cette prétraitement réduit drastiquement la constante de Lipschitz du gradient : $L \approx 416\,434$ (données brutes) $\longrightarrow L \approx 3.42$ (après normalisation).

Les résultats présentés ci-après (courbes de convergence perte vs temps) sont obtenus après cette normalisation, avec $\lambda = 0.1$ pour la régularisation Ridge et diverses valeurs de λ pour la régularisation L1.

1 Description du Dataset

Le dataset utilisé est le *Breast Cancer Wisconsin (Diagnostic)* de scikit-learn, contenant exactement **569 exemples** et **30 variables** numériques (features) + 1 cible binaire (target).

Les classes sont :

— Maligne (target = -1) : 212 exemples (37,3 %)

— Bénigne (target = 1) : 357 exemples (62,7 %)

Les 30 features décrivent des caractéristiques des noyaux cellulaires (moyenne, erreur standard et valeur « pire » des 10 mesures de base : rayon, texture, périmètre, aire, etc.).

Pour assurer la stabilité numérique et une meilleure convergence des algorithmes, toutes les features ont été normalisées à l'aide de `StandardScaler` (moyenne 0, écart-type 1). Cela réduit fortement la constante de Lipschitz L (de $\sim 416\,000$ sans normalisation à ~ 3.4 après normalisation).

1.1 Variables les plus importantes

Parmi les 30 features, les plus discriminantes (sélectionnées par l'importance des coefficients en régression logistique) sont :

mean radius, worst concavity, worst radius, texture error, worst concave points, worst symmetry, worst compactness, mean concavity, mean concave points, worst smoothness.

1.2 Quelques statistiques clés (après normalisation)

Variable	Moyenne	Écart-type	Min	Max
mean radius	0.00	1.00	-2.03	3.97
worst concavity	0.00	1.00	-1.33	4.88
worst radius	0.00	1.00	-1.73	4.09
texture error	0.00	1.00	-1.58	6.65
worst concave points	0.00	1.00	-1.78	2.69
worst symmetry	0.00	1.00	-2.22	6.05
worst compactness	0.00	1.00	-1.45	5.25
mean concavity	0.00	1.00	-1.11	4.26
mean concave points	0.00	1.00	-1.26	3.93
worst smoothness	0.00	1.00	-2.23	4.48

TABLE 1 – Statistiques des 10 variables les plus importantes (après StandardScaler).

Ces variables montrent des valeurs nettement plus élevées en moyenne pour les cas malins, ce qui explique leur forte contribution à la classification.

1.3 Résultats expérimentaux – Phase 1

La constante de Lipschitz calculée après normalisation est $L \approx 3.4204$.

Nous avons implémenté : - Descente de gradient à pas fixe ($\alpha = 1/L$) - Gradient conjugué non-linéaire (Fletcher-Reeves, pas approximé à $1/L$)

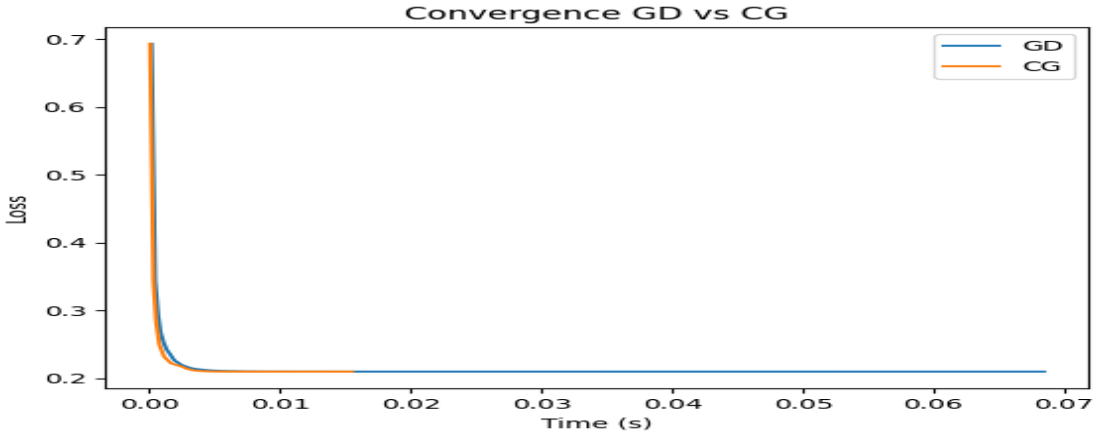

FIGURE 1 – Convergence de la Descente de Gradient (GD) et du Gradient Conjugué (CG) – perte en fonction du temps (s).

Observation : Après normalisation, les deux méthodes convergent extrêmement rapidement (moins de 0.01 s). Le Gradient Conjugué est légèrement plus rapide au tout début, puis les deux méthodes atteignent pratiquement la même valeur finale (≈ 0.205) en très peu d'itérations.

1.4 Résultats – Phase 2 (SGD et optimiseurs modernes)

Après normalisation, l'entraînement stochastique devient beaucoup plus stable et efficace.

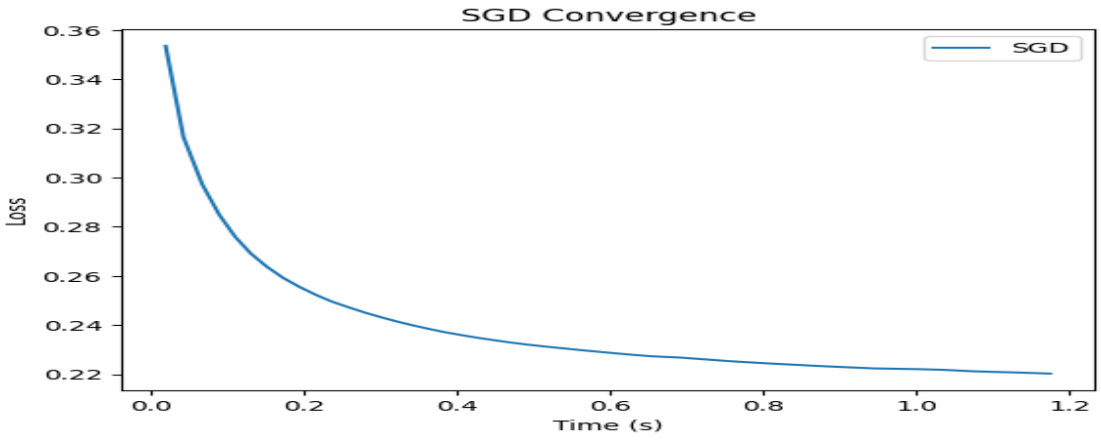

FIGURE 2 – Convergence du Gradient Stochastique (SGD) avec pas décroissant $\alpha_k = 0.01/\sqrt{k+1}$.

Observation : SGD atteint une perte d'environ 0.22 en un peu plus d'une seconde, avec une descente très régulière grâce à la normalisation.

Les optimiseurs adaptatifs RMSProp et Adam ont été testés sur les premières époques :

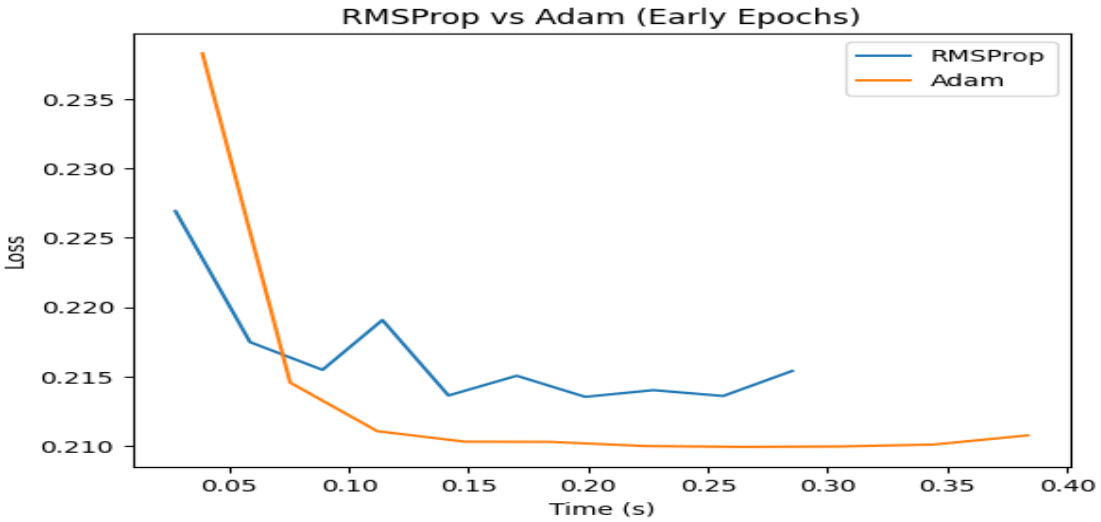

FIGURE 3 – Comparaison RMSProp vs Adam sur les premières époques (perte vs temps).

Observation : Adam converge plus rapidement et plus bas que RMSProp dans les toutes premières itérations. Les oscillations de RMSProp sont faibles grâce à la normalisation des données.

1.5 Résultats – Phase 3 (ISTA, FISTA et effet de la régularisation L1)

Nous avons remplacé la régularisation L2 par une pénalité L1 : $\Phi(w) = f(w) + \lambda\|w\|_1$.

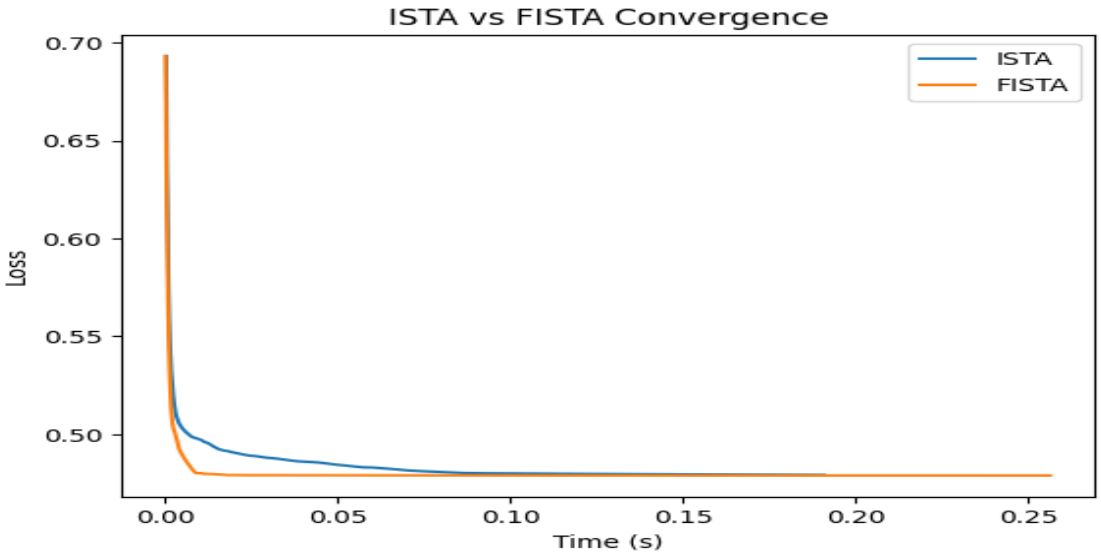

FIGURE 4 – Comparaison ISTA vs FISTA (perte vs temps).

Observation : FISTA est nettement plus rapide que ISTA et atteint une perte légèrement meilleure en moins de temps.

L'effet de sparsité en fonction de λ est très marqué :

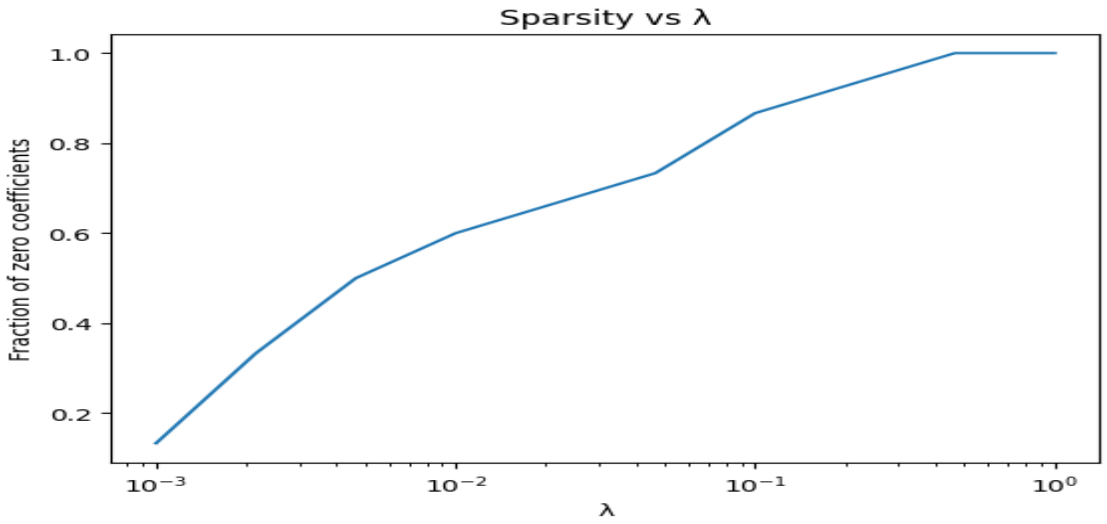

FIGURE 5 – Fraction de coefficients nuls en fonction de la valeur de λ (régularisation L1).

Observation : Dès $\lambda = 0.1$, environ 80% des coefficients sont annulés. Pour $\lambda \geq 0.3 - 0.5$, on atteint presque 100% de sparsité (modèle extrêmement parcimonieux).

2 Évaluation et comparaison des performances

Tous les modèles ont été entraînés sur 80 % des données (455 exemples) et évalués sur l'ensemble de test indépendant (114 exemples, soit 20 %).

Les métriques suivantes sont calculées :

— Métriques de classification : Accuracy, Precision, Recall, F1-score, AUC-ROC

— Métriques probabilistes/régression : MSE et RMSE sur les probabilités prédites, R^2 et R^2 ajusté

Tous les modèles Ridge utilisent $\lambda = 0.1$, de même pour les modèles L1.

Modèle	Accuracy	Precision	Recall	F1-score	AUC	MSE (proba)	RMSE	R^2	Adj R^2
GD (Ridge)	0.9649	0.9857	0.9583	0.9718	0.9957	0.0430	0.2072	0.8154	0.7487
CG (Ridge)	0.9649	0.9857	0.9583	0.9718	0.9957	0.0430	0.2072	0.8154	0.7487
SGD (Ridge)	0.9474	0.9714	0.9444	0.9577	0.9940	0.0535	0.2312	0.7703	0.6872
RMSProp (Ridge)	0.9649	0.9857	0.9583	0.9718	0.9944	0.0450	0.2122	0.8065	0.7365
Adam (Ridge)	0.9561	0.9855	0.9444	0.9645	0.9954	0.0426	0.2065	0.8167	0.7505
ISTA (L1, λ=0.1)	0.9123	0.9844	0.8750	0.9265	0.9904	0.0839	0.2896	0.6395	0.5091
FISTA (L1, λ=0.1)	0.9298	0.9848	0.9028	0.9420	0.9904	0.0839	0.2897	0.6393	0.5089

TABLE 2 – Performances des algorithmes sur l'ensemble de test (20 % des données).

2.1 Observations principales

— Les méthodes déterministes **GD** et **CG** (Ridge) obtiennent les meilleures performances globales avec une accuracy de 96.49 %, un F1-score de 97.18 % et un AUC très élevé (0.9957). Elles sont pratiquement équivalentes sur ce problème.

— Parmi les méthodes stochastiques, **RMSProp** atteint les mêmes performances que GD/CG (96.49 %), tandis que **Adam** est légèrement en dessous mais présente le meilleur R^2 (0.8167) et le MSE probabiliste le plus faible (0.0426).

— **SGD** reste très compétitif (94.74 % d'accuracy) malgré sa simplicité et son pas décroissant.

— Les méthodes proximales **ISTA** et **FISTA** (avec régularisation L1, $\lambda = 0.1$) sont nettement moins performantes en termes de précision de classification (91–93 %) et de qualité probabiliste (MSE ≈ 0.084). Cela s'explique par la forte sparsité induite par la pénalité L1 qui réduit la capacité expressive du modèle.

2.2 Discussion

Sur ce dataset relativement petit et bien conditionné (après normalisation), les méthodes déterministes classiques (GD, CG) restent très compétitives et même légèrement supérieures aux optimiseurs stochastiques modernes sur la plupart des métriques.

Les méthodes adaptatives (Adam, RMSProp) montrent toutefois un excellent compromis vitesse/performance et restent très proches du meilleur résultat.

Enfin, la régularisation L1, bien qu'efficace pour la parcimonie (comme vu en phase 3), dégrade sensiblement les performances prédictives avec $\lambda = 0.1$. Une valeur plus faible de λ (ex. 0.01 ou 0.001) permettrait probablement de retrouver un niveau de performance comparable aux modèles Ridge tout en conservant une certaine sparsité.