REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

UNIVERSITE A. MIRA DE BEJAIA
–B E J A I A–

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

وزارة التعليم العالي والبحث العلمي

جامعة عبد الرحمان ميرة
- بجاية -

Mémoire de fin d'études

Pour l'obtention du diplôme de Master

Filière : Mathématique
Spécialité : Probabilités, Statistiques et Applications

Présenté par : Leila HADDAD

Thème

Les algorithmes ISTA et FISTA dans les problèmes du type Lasso

Soutenu publiquement, le 29/ 06/ 2025 , devant le jury composé de :

Mme N. Rebouh	MAA	Université de Béjaïa	Présidente
M. F. Maouche	MCA	Université de Béjaïa	Encadrant
M. S. Ouazine	MCB	Université de Béjaïa	Examinateur

Année universitaire : 2024 / 2025

Dédicace

À mes parents chéris,
Mokrane et Djamila,
merci pour votre amour inconditionnel, vos prières silencieuses, et votre soutien constant dans chaque étape de ma vie.

À Lydia, ma sœur adorée,
ta tendresse et ta lumière m'ont toujours accompagnée.

À Lamine, Juba et Mourad,
mes frères au cœur généreux,
merci pour votre force tranquille et votre présence indéfectible.

À mes amis précieux : Zouzou, Ryma, Zakaria, Amina, Rayane, et Ikram,
merci pour vos encouragements, votre humour, vos conseils et votre fidélité.
Vous avez coloré ce chemin avec bienveillance et amitié.

À tous ceux qui ont cru en moi, de près ou de loin,
ce mémoire vous est dédié avec amour et gratitude.

H. Leila

Remerciements

Au terme de ce parcours riche en apprentissages, en défis et en rencontres,
je tiens à exprimer, du fond du cœur, ma plus sincère reconnaissance.

À Monsieur F. Maouche, mon encadrant,
je vous remercie pour le suivi de ce travail dans le cadre du Master.

À Madame N. Rebouh, présidente du jury,
merci pour l'honneur que vous me faites en évaluant ce travail,
et pour le regard bienveillant que vous portez sur cet effort.

À Monsieur S. Ouazine, examinateur,
merci pour votre disponibilité, votre engagement
et l'attention sincère que vous avez portée à ce mémoire.

À tous mes enseignants du Master PSA,
merci pour la richesse de vos enseignements, vos exigences formatrices
et la passion du savoir que vous nous avez transmise.

À ma chère promotion PSA,
merci pour les rires partagés, les doutes apaisés, les révisions passionnées
et cette belle complicité née au fil des années.

Enfin, à celles et ceux qui, dans l'ombre ou dans la lumière,
ont cru en moi, m'ont soutenue, encouragée et inspirée,
je vous dédie ces lignes avec gratitude, tendresse et respect.

Table des matières

Liste des figures

Liste des tableaux

Liste des Algorithmes

Liste des abréviations et notations

Acronyme	**Signification**
$\mathbb{E}(.)$	Espérance mathématique.
$\mathbb{V}(.)$	Variance mathématique.
$B(.)$	Biais.
SCR	Somme résiduelle des carrés.
MCO	Moindres carrés ordinaires.
MSE	Erreur quadratique moyenne (Mean Squared Error).
Lasso	Least Absolute Shrinkage and Selection Operator.
ISTA	Iterative Soft-Thresholding Algorithm.
FISTA	Fast Iterative Soft-Thresholding Algorithm.
ℓ_1	Norme $\|.\|_1$, utilisée notamment dans le Lasso.
ℓ_2	Norme $\|.\|_2$, utilisée notamment dans le Ridge.
$\hat{\beta}$	Estimateur des coefficients.
λ	Paramètre de régularisation.
L	Constante de Lipschitz du gradient ∇f.
$\mathcal{G}_s$	Opérateur gradient.
tol	Tolérance de convergence dans les algorithmes itératifs.
max_iter	Nombre maximal d'itérations autorisées.

Introduction générale

Dans un environnement où les données servent de moteur aux systèmes décisionnels, que ce soit dans le secteur scientifique ou industriel, l'évolution des méthodes statistiques a été remarquable. Face à l'essor massif des volumes de données et à la complexité croissante des phénomènes à modéliser, les méthodes d'analyse traditionnelles doivent évoluer. La régression linéaire reste, parmi ces dernières, un pilier fondamental de la modélisation statistique [45, 63]. Elle établit une relation linéaire entre une variable dépendante et un ensemble de variables explicatives. Sa simplicité, son interprétabilité et sa capacité d'adaptation en font un outil indispensable en apprentissage supervisé et en modélisation prédictive.

Cependant, l'application de la régression linéaire classique se heurte à des limites bien connues. D'une part, la présence de multicolinéarité c'est-à-dire une forte corrélation entre les variables explicatives peut engendrer une instabilité des coefficients estimés, rendant le modèle peu fiable [54, 29]. D'autre part, dans des contextes de haute dimension où le nombre de variables dépasse celui des observations ($p > n$), le modèle devient non identifiable, et les solutions classiques par moindres carrés ne sont plus applicables.

Pour surmonter ces difficultés, des techniques de régularisation ont été introduites à partir des années 1970, notamment la régression Ridge, introduite par Hoerl et Kennard en 1970 [32], puis analysée plus en détail dans des travaux ultérieurs tels que ceux de Marquardt et Snee [43].Cette méthode repose sur l'ajout d'un terme de pénalisation quadratique (ℓ_2) dans la fonction objectif, ce qui permet de réduire la variance des estimateurs, au prix d'un léger biais, mais sans sélection des variables.

C'est dans ce cadre qu'intervient la méthode Lasso (Least Absolute Shrinkage and Selection Operator), introduite en 1996 par Tibshirani [61]. Contrairement à Ridge, elle utilise une pénalisation de type ℓ_1, favorisant ainsi la parcimonie du modèle en annulant certains coefficients. Cette propriété en fait un outil privilégié pour la sélection de variables, notamment dans les contextes de haute dimension[12, 27, 7, 65].

Néanmoins, la pénalisation ℓ_1 introduit une non-différentiabilité dans la fonction objectif, ce qui empêche l'utilisation directe de méthodes classiques d'optimisation différentiables. Pour contourner cette difficulté, des algorithmes spécifiques, dits « proximaux », ont été développés. L'un des plus célèbres est l'algorithme ISTA (Ite-

rative Shrinkage-Thresholding Algorithm), initialement formalisé dans le cadre des problèmes inverses par Daubechies, Defrise et De Mol [23]. Il repose sur une itération entre une descente de gradient et un opérateur de seuillage doux.

Plus récemment, Beck et Teboulle ont proposé une version accélérée de ISTA, nommée FISTA (Fast ISTA) [5], s'inspirant des travaux de Nesterov sur l'accélération des descentes de gradient [47]. FISTA bénéficie d'une vitesse de convergence améliorée ($O(1/k^2)$ contre $O(1/k)$ pour ISTA où k est le nombre d'itérations) tout en conservant la simplicité de mise en œuvre du schéma initial. Ces méthodes ont été largement étudiées sur le plan théorique, notamment en ce qui concerne leur convergence [4, 60, 39] et se sont révélées performantes dans des applications pratiques variées telles que la détection de signaux faibles[8, 55, 6], la compression d'images ou encore la reconstruction d'IRM [42, 22, 55, 6].

Dans le présent mémoire, nous explorons une application concrète de ces techniques dans un contexte industriel réel : la prédiction de la production de plaques de carton au sein de l'entreprise Général Emballage. Cette entreprise, implantée en Algérie, nous a fourni un jeu de données couvrant les années 2020 à 2024, relatif aux quantités produites pour différents types de produits transformés. L'objectif est de construire un modèle prédictif robuste et parcimonieux permettant d'estimer la production future à partir des productions passées.

Pour ce faire, nous mettons en œuvre les algorithmes ISTA et FISTA en combinaison avec le modèle Lasso. L'accent est mis sur la sélection et la justification des paramètres d'implémentation, tels que le taux d'apprentissage, la pénalisation λ, la tolérance de convergence et le nombre maximal d'itérations. Une comparaison empirique est menée afin d'évaluer les performances des deux méthodes à travers l'erreur quadratique moyenne (MSE), en analysant leur comportement selon différents réglages d'hyperparamètres. L'objectif est de démontrer l'efficacité de ces techniques dans la modélisation de séries industrielles et d'en valider l'utilité opérationnelle dans la planification de la production.

Ce travail est structuré en quatre chapitres :

Le premier chapitre est consacré à la régression linéaire, avec un accent particulier sur les méthodes de régularisation Ridge et Lasso. Il en présente les fondements théoriques, les motivations et les formulations mathématiques.

Le deuxième chapitre introduit les bases de l'optimisation convexe nécessaires à la compréhension de l'algorithme ISTA (Iterative Shrinkage-Thresholding Algorithm). Les concepts fondamentaux tels que la convexité, la sous-différentielle et la continuité de Lipschitz sont présentés, avant de développer le principe du gradient proximal. Ce cadre théorique permet d'introduire l'algorithme ISTA, d'analyser sa convergence et de mettre en évidence ses limites en termes de vitesse.

Le troisième chapitre est dédié à l'étude approfondie de l'algorithme FISTA, considéré dans un cadre général. Son principe, sa convergence accélérée et sa structure algorithmique sont analysés.

Enfin, le quatrième chapitre présente une étude de cas appliquée à l'entreprise Général Emballage. Après une description des données industrielles de production, les algorithmes ISTA et FISTA sont implémentés sous Python afin de modéliser et prédire la production future. Cette partie justifie les choix des paramètres tels que le taux d'apprentissage, la régularisation λ, la tolérance et le nombre d'itérations. Une étude comparative des performances est ensuite menée en se basant sur l'erreur quadratique moyenne (MSE) et différents réglages des hyperparamètres.

CHAPITRE 1

Les méthodes de régression en ℓ_2 (Ridge) et ℓ_1(Lasso)

1.1 Introduction

En apprentissage statistique et en économétrie, la régression linéaire est une méthode couramment utilisée pour modéliser la relation entre une variable dépendante et un ensemble de variables explicatives. Toutefois, lorsque les données présentent une forte multicolinéarité ou un grand nombre de variables, la régression linéaire classique peut souffrir d'instabilité et de surapprentissage (overfitting) [29].

Pour pallier ces problèmes, des techniques de régularisation ont été introduites, notamment la régression Ridge (ℓ_2) et la régression Lasso (ℓ_1). Ces deux méthodes ajoutent une pénalité aux coefficients de régression pour améliorer la robustesse du modèle et éviter le surajustement [32, 61] .

La régression Ridge (ℓ_2) applique une pénalité quadratique sur les coefficients, ce qui réduit leur variance et permet d'obtenir des estimations plus stables en présence de colinéarité [32].

La régression Lasso (ℓ_1) impose une pénalité basée sur la somme des valeurs absolues des coefficients, ce qui conduit non seulement à une réduction de leur magnitude, mais aussi à une sélection automatique des variables les plus pertinentes [61].

Ces deux techniques sont largement utilisées en machine learning et en statistiques, notamment pour l'analyse de données à haute dimension, où la sélection de variables et la gestion de la complexité du modèle sont essentielles [30].

Dans ce chapitre, nous allons explorer en détail les régressions Ridge (ℓ_2) et Lasso (ℓ_1), deux techniques de régularisation utilisées pour améliorer la stabilité et la performance des modèles de régression linéaire. Nous commencerons par introduire les définitions et les démonstrations mathématiques associées à ces méthodes, en mettant en évidence leurs propriétés statistiques et la manière dont elles influencent l'estimation des coefficients. Ensuite, nous examinerons les interprétations géométriques de ces approches, qui permettent de mieux comprendre leur impact sur la structure des modèles et la sélection des variables. Enfin, nous discuterons des avantages et inconvénients de chacune de ces méthodes, notamment en termes de

biais-variance, de sélection de variables et de sensibilité aux données. Cette analyse approfondie nous permettra de mieux cerner dans quels contextes l'une ou l'autre de ces méthodes est la plus appropriée.

1.2 Modèle et estimation

Ayant diagnostiqué un problème mal conditionné mais désirant conserver toutes les variables explicatives pour des raisons d'interprétation, il est possible d'améliorer les propriétés numériques et la variance des estimations en considérant un estimateur biaisé des paramètres par une procédure de régularisation. Soit le modèle linéaire [7] :

$$y = X\beta + \epsilon \tag{1.1}$$

Où

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

- y : est le vecteur a expliquer de taille n ;
- X : est la matrice de taille $n(p+1)$, qui contient l'ensemble des observations des variables explicatives, avec une première colonne formée par la valeur 1 ;
- β : vecteur des paramètres inconnus à estimer ;
- ϵ : vecteur d'erreur de longueur n.

1.3 Régression pénalisée

L'estimateur des moindres carrés est sans biais et présente la plus faible variance parmi tous les estimateurs linéaires sans biais de β. Cependant, pour améliorer la précision des prédictions, il devient nécessaire de réduire la variance, même si cela entraîne une augmentation du biais. L'objectif est que cette réduction de la variance compense la perte due à l'introduction d'un biais, permettant ainsi un gain net en précision prédictive. Hoerl et Kennard [32] ont proposé des méthodes de régression pénalisée qui contraignent les coefficients de β à adopter une certaine forme afin de minimiser l'erreur de prédiction. Cette approche consiste à ajouter une pénalité à la fonction objectif à minimiser.

Proposition 1.3.1. [12] Le coefficient de regression pénalisée est défini comme l'estimateur $\hat{\beta}^{(p)}$ qui minimise

$$\hat{\beta}^{(p)} = \arg\min_{\beta \in \mathbb{R}^p} \{\|y - X\beta\|_2^2\}; \tag{1.2}$$

Sous la contrainte

$$\|\beta\|_q^r \leq t,$$

Où la constante t est appelé "borne absolue" de la contrainte.
• Pour q=r=2 : Regression Ridge.
• Pour q=r=1 : Regression Lasso.
• Regression Elastic Net : Combine les propriétés de Ridge et Lasso.

1.3.1 Pénalisation de la fonction de coût

La régression régularisée consiste à intégrer une pénalisation dans la fonction de coût d'un modèle de régression afin de prévenir le surajustement et d'améliorer sa capacité de généralisation. Les approches les plus répandues sont la régression Ridge et la régression Lasso[27].

1.4 Colinéarité

La régression Ridge est utilisée lorsque les variables explicatives sont fortement corrélées entre elles, un phénomène appelé colinéarité ou multi-colinéarité. Une variable est mathématiquement colinéaire si elle est une combinaison linéaire exacte des autres, rendant impossible l'estimation des coefficients par la méthode des moindres carrés. En revanche, une variable est statistiquement colinéaire si elle est approximativement une combinaison linéaire des autres, ce qui entraîne des estimateurs instables avec des variances potentiellement très élevées. La régression Ridge permet de pallier ce problème en introduisant une pénalisation qui stabilise les estimations. Elle ajoute une pénalité $p(\beta)$ de norme ℓ_2 aux coefficients du modèle afin de réduire leur amplitude et éviter ainsi un surapprentissage[32, 63].

1.4.1 Détection de la multicolinéarité

En général, deux tests sont présenter pour permettre de détecter la présence d'une multicolinéarité[54] :

Test de Klein

Ce test n'est pas un test statistique au sens d'hypothéses mais simplement un critère de présemption de multicolinéarité.
Il est basé sur la comparaison du coefficient R^2 calculé sur le modèle à k variables suivantes suivant :

$$y_i = \hat{a}_0 + \hat{a}_1 x_{1i} + \hat{a}_2 x_{2i} + \ldots + \hat{a}_k x_{ki} + e_i, \quad i = 1, 2, \ldots, n$$

et les coefficients de corrélations simple $r^2_{x_i,x_j}$ entre les variables explicatives pour $i \neq j$.
- Si $R^2 < r^2_{x_i,x_j}$, il y a présomption de multicolinéarité.
- Si $R^2 > r^2_{x_i,x_j}$, il y a absence de multicolinéarité.

Test de Farrar et Glauber

Ce test comporte deux étapes :
- 1ére étape : Elle consiste a calculer le déterminant D de la matrice des coefficients de corrélation entre les variables explicatives :

$$D = \begin{vmatrix} 1 & r_{x_1,x_2} & r_{x_1,x_3} & \cdots & r_{x_1,x_k} \\ r_{x_2,x_1} & 1 & r_{x_2,x_3} & \cdots & r_{x_2,x_k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{x_k,x_1} & r_{x_k,x_2} & r_{x_k,x_3} & \cdots & 1 \end{vmatrix}$$

- 2éme étape : Dans cette étape, effectuons le test de Khi-deux en posant les hypothéses suivantes :
H_0 : D = 1 (les variables sont orthogonales) ;
H_1 : D < 1 (les variables sont dépendantes).
On calcule la statistique

$$^*\chi^2 = -[n - 1 - \frac{1}{6}(2K + 5)]ln(D)$$

avec K=k+1, où k : nombres de variables explicatives et n : taille de l'échantillon.
- Si $^*\chi^2 \geq \chi^2$ (lu dans la table de Khi-deux à $\frac{1}{2}k(k+1)$ degré de liberté au seuil de signification α), alors on rejette H_0, il y'a présence de multicolinéarité.
- Si $^*\chi^2 < \chi^2$, alors on accepte l'hypothése H_0 d'orthogonalité.

1.5 Régression en Ridge ℓ_2

Fut introduite en 1970 par A.E. Hoerl[32]. Il avait constaté que l'existence de corrélation entre les variables explicatives pouvait entrainer des erreurs dans l'estimation des paramètres lors de l'application de la méthode des moindres carrées. Comme alternative à cette méthode traditionnelle, il développa la méthode de la regression Ridge. Cette méthode permet le calcul et l'utilisation d'estimateurs biaisé mais de variance plus faible que les estimateurs des moindres carrées est de fournir des estimateurs de variances minimales parmi les estimateurs non biaisés, le but de la méthode Ridge est de minimiser l'erreur quadratique moyenne des estimateurs, c'est-à-dire de choisir un compromis entre le biais et la variance[63].

En présence de multi-colinéarité, la matrice (X^TX) est presque singulière, rendant son inversion problématique, tout comme l'inversion d'un nombre très proche de zéro. Cela entraîne des éléments très grands dans la matrice inversée, augmentant ainsi les variances dans la matrice de variance-covariance. Par conséquent, les estimateurs de la régression deviennent instables et très sensibles aux variations des données, rendant les coefficients peu fiables.

Pour résoudre ce problème, la régression Ridge est utilisée. Elle modifie la matrice X^TX en ajoutant un terme de régularisation (λI), où (λ) est un réel positif et (I) est la matrice identité. Cela éloigne la matrice de la quasi-singularité, stabilisant ainsi les estimateurs. En résumé, la régression Ridge remplace l'estimateur des moindres carrés en régularisant la solution pour gérer la multi colinéarité [54].

1.5.1 Expression de l'estimateur des MCO

Définition 1.5.1. [7] (Estimateur des MCO)
L'estimateur des Moindres Carrés Ordinaires $\hat{\beta}$ est défini comme suit :

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \left\| y - X\beta \right\|_2^2$$

Proposition 1.5.2. [7] (Expression de $\hat{\beta}$)
L'estimateur $\hat{\beta}$ des Moindres Carrés Ordinaires a pour expression :

$$\hat{\beta} = \left(X^TX\right)^{-1} X^Ty$$

et la matrice de projection H s'écrit : $H = X(X^TX)^{-1}X^T$

Par l'estimateur Ridge :

$$\hat{\beta}^{\text{Ridge}} = \left(X^TX + \lambda I\right)^{-1} X^Ty$$

Proposition 1.5.3. [38] Cette méthode a comme pénalité :

$$p(\beta) = \left\|\beta\right\|_2^2 \tag{1.3}$$

Avec :

$$\left\|\beta\right\|_2^2 = \sum_{i=1}^p \beta_i^2 = \beta^T\beta = l_2$$

Définition 1.5.4. [29] L'estimateur Ridge de $\hat{\beta}^{\text{Ridge}}$ est défini par :

$$\hat{\beta}^{\text{Ridge}} = \arg\min_{\beta} \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j \right)^2 \right\}, \tag{1.4}$$

Tel que :

$$\sum_{j=1}^p \beta_j^2 \leq t.$$

On peut écrire le problème de la méthode Ridge sous une forme équivalente avec le lagrangien :

$$\hat{\beta}_{\text{Ridge}} = \arg\min_{\beta} \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\} \tag{1.5}$$

Où $\lambda \geq 0$ est un paramètre de régularisation.

Démonstration
Résoudre ce problème, est équivalent a résoudre le problème de minimisation sous contrainte suivant [48] :

$$\hat{\beta}^{\text{Ridge}} = \arg\min_{\beta} \left\{ \|y - X\beta\|^2 \right\} \tag{1.6}$$

Ou encore,

$$\hat{\beta}^{\text{Ridge}} = (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta$$
$$\hat{\beta}^{\text{Ridge}} = y^T y - 2\beta^T X^T y + \beta^T X^T X \beta + \lambda \beta^T \beta$$

Si on dérive par rapport à β et qu'on pose les dérivées égales à zéro, on trouve :

$$X^T y = X^T X \beta + \lambda \beta$$
$$X^T y = (X^T X + \lambda I)\beta$$

Résultat. [48] L'estimateur $\hat{\beta}_{\text{Ridge}}$ est la solution de l'équation normal pénalisé :

$$(X^T X + \lambda I)\beta = X^T y \tag{1.7}$$

La matrice symétrique $X^T X$ étant définie semi-défini positive, toutes ses valeurs propres sont positives ou nulles. Par conséquents $X^T X + \lambda I$ est symétrique pour tout $\lambda \geq 0$, donc inversible. Alors, l'estimateur Ridge a une unique solution donnée par :

$$\hat{\beta}^{\text{Ridge}} = (X^T X + \lambda I)^{-1} X^T y \tag{1.8}$$

Dans le cas où X est orthogonale,

$$\hat{\beta}^{\text{Ridge}} = (1 + \lambda)^{-1} X^T y = (1 + \lambda)^{-1} \hat{\beta}^{\text{MCO}} \tag{1.9}$$

et

$$\hat{\beta}_j^{\text{Ridge}} = \frac{\hat{\beta}_j^{\text{MCO}}}{1+\lambda}$$

Avec $\hat{\beta}_j^{\text{MCO}}$ est donné par l'équation suivante :

$$\hat{\beta}_j^{\text{MCO}} = (X^T X)^{-1} X^T y$$

Remarque.

- Il existe une bijection entre t et λ, liée par l'équation [10] :

$$\left\| \hat{\beta}^{\text{Ridge}} \right\|^2 = t^2$$

- Les solutions du problème ne sont pas invariantes par changement d'échelle. Il est donc usuel de standardiser les variables avant l'application de la méthode [27].
- Les variables sont centrées (c'est-à-dire que l'on ne pénalise pas la constante).
- La fonction à minimiser s'écrit sous forme matricielle :

$$\hat{\beta}^{\text{Ridge}} = \arg\min_{\beta} \left\{ \|y - X\beta\|^2 + \lambda \|\beta\|_2^2 \right\}$$

Remarque.

1. L'estimateur Ridge n'est pas invariant par renormalisation des vecteurs X_j , il est préférable de normaliser les vecteurs avant de minimiser le critère.
2. La constante β_0 n'intervient pas dans la pénalité, sinon, le choix de l'origine pour y aurait une influence sur l'estimation de l'ensembles des paramètres. On obtient $\hat{\beta}_0 = \bar{y}$, ajouter une constante à y ne modifie pas les $\hat{\beta}_j^{\text{Ridge}}$ pour $j \geq 1$.
3. $X^T X$ est une matrice symétrique positive, Il en résulte que pour tout $\lambda > 0$, $X^T X + \lambda I$ est nécessairement inversible.
4. On montre que l'estimateur Ridge revient encore à estimer le modèle par les moindres carrés sous la contrainte que la norme du vecteur β des paramètres ne soit pas trop grande :

$$\hat{\beta}^{\text{Ridge}} = \arg\min_{\beta} \left\{ \|y - X\beta\|^2 ; \|\beta\|_2^2 \leq t \right\}. \tag{1.10}$$

1.5.2 Propriétés de l'estimateur Ridge

Espérance [65]

Proposition 1.5.5. Revenons aux définitions de l'estimateur Ridge et des moindres carrées MCO [25] :

$$\hat{\beta}^{\text{Ridge}} = \left(X^T X + \lambda I\right)^{-1} X^T y$$

et

$$\hat{\beta}^{\text{MCO}} = \left(X^T X\right)^{-1} X^T y$$

En multipliant la seconde égalité par $X^T X$, on obtient $X^T X \hat{\beta}^{\text{MCO}} = X^T y$, d'aprés la définition de l'estimateur MCO vient :

$$\hat{\beta}^{\text{Ridge}} = \left(X^T X + \lambda I\right)^{-1} X^T X \hat{\beta}^{\text{MCO}}$$

Cette écriture permet de calculer facilement le biais et la variance de l'estimateur Ridge. En utilisant les propriétés de l'estimateur MCO, l'espérance de l'estimateur Ridge est :

$$\begin{aligned}\mathbb{E}(\hat{\beta}^{\text{Ridge}}) &= \mathbb{E}(X^TX + \lambda I)^{-1}X^TX\mathbb{E}(\hat{\beta}^{\text{MCO}}) \\ &= \left(X^TX + \lambda I\right)^{-1} X^TX\beta^{\text{MCO}} \\ &= \left(X^TX + \lambda I\right)^{-1}\left(X^TX + \lambda I - \lambda I\right)\beta^{\text{MCO}} \\ &= \beta^{\text{MCO}} - \lambda\left(X^TX + \lambda I\right)^{-1}\beta^{\text{MCO}}\end{aligned}$$

Théorème 1.5.6. [27] $\hat{\beta}^{\text{MCO}} = (X^TX)^{-1}X^Ty$ est l'estimateur qui minimise la somme des carrés des résidus (SCR) avec X^T la transposée de X .

Biais [65]

Proposition 1.5.7. Le biais de cet estimateur vaut :

$$\begin{aligned}\text{Biais}(\hat{\beta}^{\text{Ridge}}) &= \mathbb{E}((\hat{\beta}^{\text{Ridge}}) - \beta) \\ &= -\lambda\left(X^TX + \lambda I\right)^{-1}\hat{\beta}^{\text{MCO}}\end{aligned}$$

Ce qui montre que l'estimateur Ridge est un estimateur biaisé (a un biais non nul)[31] contrairement à l'estimateur MCO.

Variance [65]

Proposition 1.5.8. La variance de cet estimateur vaut :

$$\begin{aligned}\text{Var}(\hat{\beta}^{\text{Ridge}}) &= \text{Var}\left[\left(X^TX + \lambda I\right)^{-1}X^Ty\right] \\ &= \left[(X^TX + \lambda I)^{-1}X^T\right]\left[X(X^TX + \lambda I)^{-1}\right]\text{Var}(y)\end{aligned}$$

Où cette hypothèse

$$\text{Var}(y) = \sigma^2 I$$

Résultat. On obtient finalement la variance de l'estimateur Ridge [65] :

$$\text{Var}(\hat{\beta}^{\text{Ridge}}) = \sigma^2\left(X^TX + \lambda I\right)^{-1}\left(X^TX\right)\left(X^TX + \lambda I\right)^{-1} \tag{1.11}$$

Erreur quadratique moyenne MSE

Proposition 1.5.9. Pour mesurer un compromis entre le biais et la variance d'un estimateur, on calcule généralement son erreur quadratique moyenne symbolisé par MSE [63].

L'erreur quadratique moyenne (MSE) de l'estimateur $\hat{\beta}^{\text{Ridge}}$ est définie comme [45] :

$$\text{MSE}(\hat{\beta}^{\text{Ridge}}) = \mathbb{E}\left[(\hat{\beta}^{\text{Ridge}} - \beta)^2\right] = \text{Var}(\hat{\beta}^{\text{Ridge}}) + \left[\mathbb{E}(\hat{\beta}^{\text{Ridge}}) - \beta\right]^2$$

Où

$$\text{MSE}(\hat{\beta}^{\text{Ridge}}) = \text{Var}(\hat{\beta}^{\text{Ridge}}) + \left[\text{Biais}(\hat{\beta}^{\text{Ridge}})\right]^2$$

Démonstration

$$\begin{aligned}
\text{MSE}(\hat{\beta}^{\text{Ridge}}) &= \text{Var}(\hat{\beta}^{\text{Ridge}}) + \left[\text{Biais}(\hat{\beta}^{\text{Ridge}})\right]^2 \\
&= \sigma^2 \text{Tr}\left[\left(X^T X + \lambda I\right)^{-1} (X^T X) \left(X^T X + \lambda I\right)^{-1}\right] + \left[-\lambda \left(X^T X + \lambda I\right)^{-1} B^{\text{MCO}}\right]^2 \\
&= \sigma^2 \sum_{j=1}^{p} \frac{\lambda_j}{(\lambda_j + \lambda)^2} + \lambda^2 \left(X^T X + \lambda I\right)^{-2} \left(\beta^{\text{MCO}}\right)^2
\end{aligned}$$

Où $\lambda_1, \lambda_2, \ldots, \lambda_p$ sont les valeurs propres de la matrice $X^T X$.

Remarque. [43]
L'erreur quadratique moyenne (MSE) est utilisée comme critère pertinent lorsque l'on accepte une légère augmentation du biais en échange d'une réduction significative de la variance. C'est précisément ce que permettent les solutions de régression Ridge et d'inverse généralisée. MSE désigne la distance quadratique moyenne attendue à β. Cette notion entraîne deux corollaires importants.

1. Le terme de variance est une fonction décroissante de λ.
2. Le terme de biais est une fonction croissante de λ.
   Si $\hat{\beta}^{\text{Ridge}}$ est borné, il existe un $\lambda > 0$ tel que :

$$\text{MSE}(\hat{\beta}^{\text{Ridge}}) < \text{MSE}(\hat{\beta})$$

1.5.3 Relation entre le paramètre λ, le biais, la variance et le MSE

Interprétation des courbes

• Variance (courbe bleue décroissante)

D'aprés la figure 1.1, lorsque λ augmente, la variance diminue, cela est attendu car une pénalisation plus forte réduit la complexité du modèle, le rendant moins sensible aux fluctuations des données d'entraînement.

• **Biais (courbe rouge croissante)**

À mesure que λ augmente, le biais augmente également, cela est dû à la contrainte imposée aux coefficients, les rapprochant de zéro et entraînant une sous-estimation des relations dans les données (figure1.1).

• **MSE (courbe noire en forme de U)**

La courbe de l'erreur quadratique moyenne montre une forme en U (figure1.1), un compromis optimal se trouve autour du minimum de cette courbe, où la combinaison entre biais et variance donne la meilleure prédiction possible.

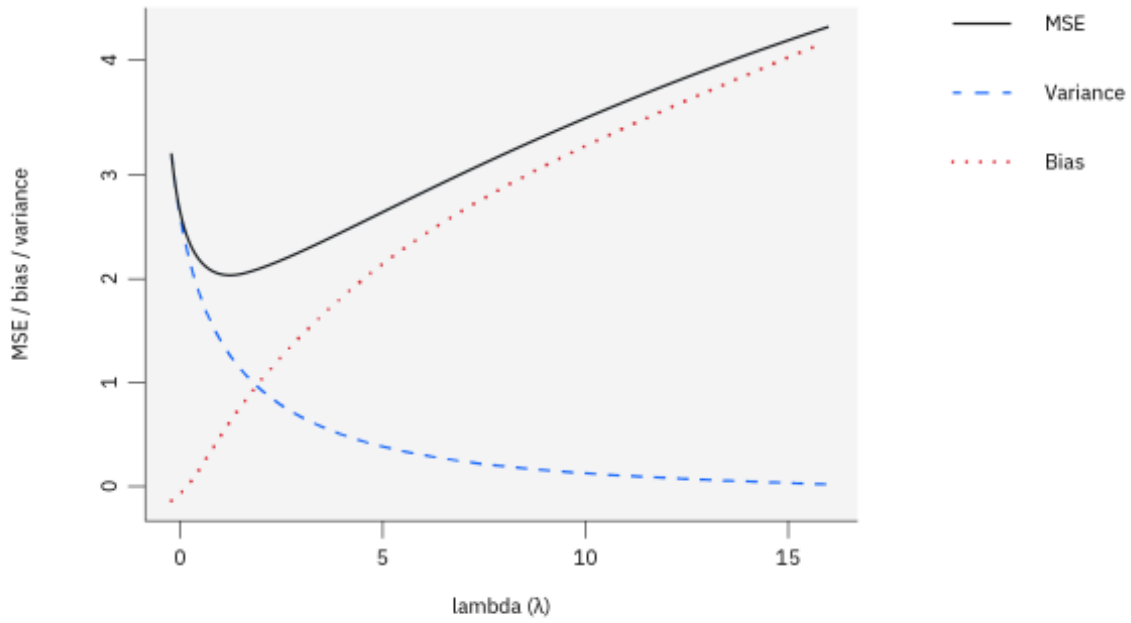

FIGURE 1.1 – Relation entre le paramètre λ, le biais, la variance et le MSE

Le graphique (figure1.1) illustre clairement le compromis biais-variance en régression Ridge. Il montre qu'un choix judicieux du paramètre λ permet de minimiser l'erreur totale en équilibrant la réduction de la variance et l'augmentation du biais.

Corollaire 1.5.10.

1. Si $\lambda \to 0$, alors $\mathbb{E}(\hat{\beta}^{\text{Ridge}}) \to \hat{\beta}$, $\text{Biais}(\hat{\beta}^{\text{Ridge}}) \to 0, \text{Var}(\hat{\beta}^{\text{Ridge}}) \to \infty$
2. Si $\lambda \to \infty$, alors $\mathbb{E}(\hat{\beta}^{\text{Ridge}}) \to 0$, $\text{Biais}(\hat{\beta}^{\text{Ridge}}) \to \infty$, $\text{Var}(\hat{\beta}^{\text{Ridge}}) \to 0$

Ainsi, la méthode Ridge introduit toutes les variables prédictives dans le modèle final, ce qui complique l'interprétation du modèle.

Théorème 1.5.11. [32]
• La variance totale est une fonction continue et monotone décroissante de λ.
• Le biais au carré est une fonction continue et monotone croissante de λ.

1.5.4 Interprétation géométrique de la régression Ridge

En régression Ridge, les contours des sommes des carrés des résidus (SCR) sont représentés par des ellipses (figure 1.2). Le minimum de ces ellipses correspond aux estimations obtenues par les moindres carrés ordinaires (MCO). En régression Ridge, une contrainte est ajoutée sous la forme d'un cercle de rayon $\|\beta\|_2^2 \leq t$. L'estimation Ridge se situe au point de contact entre l'ellipse des SCR et le cercle, ce qui impose un compromis entre la minimisation de la SCR et la régularisation des coefficients[12]. L'objectif est de minimiser :

$$\text{SCR} = \sum_{i=1}^{n} (y_i - X\beta)^2$$

Sous la contrainte :

$$\|\beta\|_2^2 \leq t$$

Cela équivaut à minimiser la fonction suivante :

$$\text{SCR} + \lambda\|\beta\|_2^2$$

Où λ contrôle le compromis entre l'ajustement du modèle et la régularisation des coefficients. Une valeur élevée de λ favorise des coefficients plus petits (réduction de la variance) mais peut augmenter la SCR, tandis qu'une petite valeur de λ a permet une meilleure adaptation aux données mais avec un risque de surapprentissage.
Il existe une correspondance entre les paramètres λ et t, car choisir un λ équivaut à fixer une contrainte t sur la norme ℓ_2 des coefficients.
Lorsque λ=0, la régression Ridge devient une régression linéaire classique.
À l'autre extrême, si λ tend vers l'infini, tous les coefficients β sont fixés à zéro.

Ce compromis illustre l'équilibre entre précision de la prédiction et régularisation dans la régression Ridge.

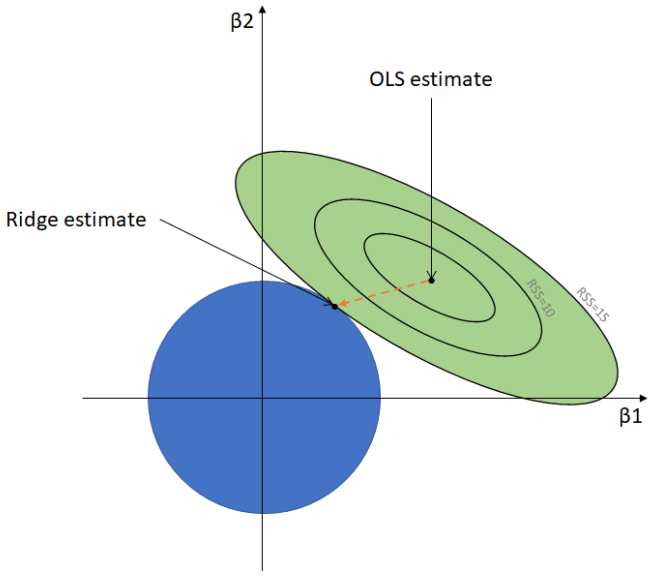

FIGURE 1.2 – La forme géométrique de la fonction de régularisation de Ridge

La régression Ridge est une méthode intuitive qui introduit un paramètre d'ajustement λ dans l'équation des estimations des coefficients β. Cette approche permet, en acceptant une certaine augmentation du biais, de réduire la magnitude des coefficients $\hat{\beta}^{\text{Ridge}} \to 0$ et de diminuer la variance, contribuant ainsi à résoudre le problème de multi-colinéarité. Cependant, malgré ses avantages, la régression Ridge ne permet pas une sélection de variables, obligeant à conserver l'intégralité du modèle.
La régression Ridge contracte les coefficients des moindres carrés vers l'origine, ce qui explique pourquoi les estimateurs Ridge sont parfois appelés estimateurs à contraction. Hocking (1976) a souligné que cet effet de contraction est relatif aux contours de la matrice $X^T X$ [45].

1.5.5 Avantages de la méthode Ridge

D'après les éléments exposés dans [34], la méthode Ridge présente les avantages suivants :

- Rétrécit les coefficients β ;
- Très performante en présence de corrélation entre les colonnes de X ;
- Améliore l'erreur de prédiction en réduisant la variance des estimateurs.

1.5.6 Inconvénients de la méthode Ridge

Bien que performante, la méthode Ridge présente néanmoins quelques inconvénients notables, identifiés notamment dans [34] :

- Cette méthode n'est pas adaptée à la sélection de variables, car lorsque des prédicteurs sont fortement corrélés, leurs coefficients ont tendance à avoir des valeurs très similaires.

- Elle n'induit pas de parcimonie dans le modèle, c'est-à-dire qu'elle ne pénalise pas les variables non pertinentes en leur attribuant des coefficients exactement nuls.

1.6 Régression en Lasso ℓ_1

Le Lasso, introduit par Tibshirani (1996), est un acronyme anglais pour Least Absolute Shrinkage and Selection Operator. La pénalité $p(\beta)$ de cette méthode nous permet non seulement de rétrécir les coefficients $\hat{\beta}_j$ comme avec l'estimateur Ridge, mais elle nous permet également de mettre certains coefficients $\hat{\beta}_j$ à 0. De ce fait, le Lasso conserve les avantages du Ridge en rétrécissant les coefficients, tout en nous permettant de faire de la sélection de modèle par le biais d'un choix de paramètre λ [65].

Définition 1.6.1. [27] L'estimateur classique des moindres carrés pour le couple (β_0, β) repose sur la minimisation de la perte quadratique (ou de l'erreur quadratique).

$$\hat{\beta}^{\text{Lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \right\}, \tag{1.12}$$

sous la contrainte :

$$\sum_{j=1}^{p} |\beta_j| \leq t.$$

Noter qu'on peut écrire le problème Lasso en un problème équivalent avec le lagrangien [34]

$$\hat{\beta}^{\text{Lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{1.13}$$

Définition 1.6.2. [65] (Fonction objectif du Lasso).
Le coefficient de régression Lasso est défini comme l'estimateur $\hat{\beta}^{Lasso}$ qui minimise une fonction objectif pénalisée par la norme ℓ_1 :

$$\hat{\beta}^{Lasso} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{1.14}$$

Cette pénalisation ℓ_1 empêche l'obtention d'une solution analytique pour l'estimateur $\hat{\beta}^{Lasso}$. Par conséquent, il est nécessaire d'utiliser des algorithmes itératifs pour le calculer. Afin d'accélérer ces calculs pour les différents algorithmes, il est recommandé d'ajouter la constante $\frac{1}{2n}$ devant la fonction objectif.

Propriétés [7]

- Si $\lambda = 0$, le Lasso correspond à une régression linéaire classique. On retrouve alors l'estimateur des Moindres Carrés, c'est-à-dire $\hat{\beta}^{Lasso} = \hat{\beta}^{MCO}$. Dans ce cas, la méthode du Lasso sélectionne toutes les variables sans distinction.
- Si $\lambda \to +\infty$, tous les coefficients $\hat{\beta}^{Lasso}$ sont réduits à zéro, soit $\hat{\beta}_j^{Lasso} = 0$ pour tout j, ce qui signifie que le Lasso ne sélectionne aucune variable explicative.
- Si $\lambda \in]0, +\infty[$, le nombre de variables sélectionnées par le Lasso diminue à mesure que λ augmente. Autrement dit, une contrainte plus forte sur le vecteur β entraîne la réduction progressive de certains coefficients $\hat{\beta}^{Lasso}$ jusqu'à ce qu'ils deviennent exactement nuls.

1.6.1 Calcul analytique de la solution de la méthode Lasso

L'estimateur Lasso ne possède pas de formule analytique dans le cas général. Toutefois, lorsque $X^tX = I$, une solution explicite peut être obtenue. Dans cette situation, l'estimateur Lasso correspond à une version de la solution des Moindres Carrés appliquant un seuillage doux aux coefficients [12].

Cas orthonormal

Proposition 1.6.3. [61] Un aperçu de la nature de la réduction des coefficients peut être obtenu dans le cas d'un plan orthonormal. Soit X la matrice de conception de dimension n × p, dont l'élément à la i-ème ligne et j-ème colonne est X_{ij}, et supposons que $X^tX = I$, la matrice identité, qui veut dire que X est une matrice orthonormale. Il est alors possible d'obtenir une solution fermée pour l'estimateur Lasso, soit :

$$\hat{\beta}_j^{Lasso} = \text{sign}\left(\hat{\beta}_j^{(MCO)}\right)\left(\left|\hat{\beta}_j^{(MCO)}\right| - n\lambda\right)^+, \tag{1.15}$$

où $n\lambda$ est déterminé par la condition $|\hat{\beta}_j| = t$.
Pour $j = 1 \dots p$, où $\hat{\beta}_j^{(MCO)}$ sont les coefficients des moindres carrés ordinaires, $(x)^+ = max(0, x)$ est l'opérateur qui prend uniquement des valeurs non-négatives et la fonction sign(·) est définie comme suit [65] :

$$\text{sign}\left(\hat{\beta}_j^{(MCO)}\right) = \begin{cases} -1, & \text{si } \hat{\beta}_j^{(MCO)} < 0, \\ 0, & \text{si } \hat{\beta}_j^{(MCO)} = 0, \\ 1, & \text{si } \hat{\beta}_j^{(MCO)} > 0. \end{cases}$$

Remarque.

- Si $\hat{\beta}_j^{(MCO)}$ est grand, alors le terme $n\lambda$ ne s'anulle pas complètement, donc la variable est retenue dans le modèle.
- Si $\hat{\beta}_j^{(MCO)}$ est petit, alors il est possible que $(|\hat{\beta}_j^{(MCO)}| - n\lambda)$ devienne négatif, donc max(0,x) le force a zéro, la variable est éliminée du modèle.

Démonstration [65] Étant donné que les variables explicatives sont orthonormées, l'estimateur des Moindres Carrés s'écrit $\hat{\beta}^{(MCO)} = X^t y$. Il suffit alors de développer la fonction objectif et de déterminer son minimum. Cette fonction vérifie :

$$\begin{aligned} S(\beta) &= \frac{1}{2n}(y - X\beta)^\top (y - X\beta) + \lambda \|\beta\|_1 \\ &= \frac{1}{2n} y^\top y - \frac{1}{n} y^\top X\beta + \frac{1}{2n}\beta^\top X^\top X\beta + \lambda\|\beta\|_1. \end{aligned}$$

Exprimée à l'aide de sommes, celle-ci devient

$$S(\beta) = \frac{1}{2n}\sum_{i=1}^{n} y_i^2 - \frac{1}{n}\sum_{j=1}^{p} \hat{\beta}_j^{(MCO)}\beta_j + \frac{1}{2n}\sum_{j=1}^{p}\beta_j^2 + \lambda\sum_{j=1}^{p}|\beta_j|. \tag{1.16}$$

pour j = 1, . . ., p.

Maintenant, en examinant cette équation, il est évident que les $\hat{\beta}_j$ qui la minimiseront seront de mêmes signes que les $\hat{\beta}_j^{(MCO)}$ correspondants.

En effet, pour minimiser cette équation, il est capital de faire en sorte que chacun des termes dans la somme $-\frac{1}{n}\sum_{j=1}^{p}\hat{\beta}_j^{(MCO)}\beta_j$ soit négatif.
En dérivant par rapport à β_j, nous trouvons donc :

$$\begin{aligned} \frac{\partial S(\beta_j)}{\partial \beta_j} &= -\frac{1}{n}\hat{\beta}_j^{(MCO)} + \frac{1}{n}\beta_j + \lambda\,\mathrm{sign}(\beta_j) \\ &= -\frac{1}{n}\hat{\beta}_j^{(MCO)} + \frac{1}{n}\beta_j + \lambda\,\mathrm{sign}\left(\hat{\beta}_j^{(MCO)}\right). \end{aligned}$$

L'estimateur $\hat{\beta}_j^{Lasso}$ satisfait alors l'équation :

$$-\frac{1}{n}\hat{\beta}_j^{(MCO)} + \frac{1}{n}\hat{\beta}_j^{Lasso} + \lambda\,\mathrm{sign}\left(\hat{\beta}_j^{(MCO)}\right) = 0.$$

En isolant $\hat{\beta}_j^{LASSO}$, nous trouvons

$$\begin{aligned} \hat{\beta}_j^{Lasso} &= \hat{\beta}_j^{(MCO)} - n\lambda\,\mathrm{sign}\left(\hat{\beta}_j^{(MCO)}\right) \\ &= \mathrm{sign}\left(\hat{\beta}_j^{(MCO)}\right)\left(\left|\hat{\beta}_j^{(MCO)}\right| - n\lambda\right). \end{aligned}$$

Sachant que $\hat{\beta}_j^{Lasso}$ et $\hat{\beta}_j^{(MCO)}$ doivent être de même signe, il est important que $\left(\left|\hat{\beta}_j^{(MCO)}\right| - n\lambda\right)$ soit positif. De ce fait, on utilise l'opérateur $(.)^+$, qui retourne uniquement des valeurs non-négatives, soit :

$$\hat{\beta}_j^{Lasso} = \mathrm{sign}\left(\hat{\beta}_j^{(MCO)}\right)\left(\left|\hat{\beta}_j^{(MCO)}\right| - n\lambda\right)^+. \tag{1.17}$$

En travaillant avec des variables explicatives orthonormées, il est également possible

d'obtenir une solution réduite pour l'estimateur Ridge, soit :

$$\hat{\beta}^{Ridge} = (1+\lambda)^{-1} X^\top y = (1+\lambda)^{-1} \hat{\beta}^{(MCO)}. \tag{1.18}$$

Remarque.

- Il est important de préciser que le rétrécissement de l'estimateur Ridge s'applique simultanément à l'ensemble des prédicteurs, empêchant ainsi l'annulation de certaines composantes. En revanche, dans un cadre orthonormé, le Lasso annule tous les coefficients lorsque $\max_j |\hat{\beta}_j^{MCO}| \leq n\lambda$, ce qui illustre sa supériorité par rapport à l'estimateur Ridge. Cependant, en pratique, les variables explicatives sont souvent corrélées entre elles [65].
- Dans le cas général, il n'existe pas de solution analytique pour l'estimateur Lasso. Sa résolution nécessite donc l'utilisation d'algorithmes itératifs, tels que la descente de gradient proximal, avec des variantes comme ISTA et FISTA, qui permettent d'accélérer la convergence [12].
- Les sommes de carrés des résidus $SRC(\beta)$ peuvent s'écrire sous la forme [34]

$$SRC(\beta) = (\beta - \hat{\beta}^{(MCO)})^T X^T X (\beta - \hat{\beta}^{(MCO)}) + \text{constante}.$$

1.6.2 Cas particulier simple pour Lasso

Considérons un cas spécial simple où n = p et où X est une matrice diagonale avec des 1 sur la diagonale et des 0 sur tous les éléments hors diagonale. Pour rendre le problème plus simple, ajoutons l'hypothèse que nous effectuons la régression sans interception. Dans ce contexte, le problème des moindres carrés devient la recherche des coefficients $\beta_1, \ldots, \beta_p$ qui minimisent [25] :

$$\sum_{j=1}^{p} (y_j - \beta_j)^2 \tag{1.19}$$

Dans ce cas, la solution des moindres carrés est donnée par

$$\hat{\beta}_j = y_i.$$

Et dans ce cadre, Le lasso revient à trouver les coefficients tels que

$$\sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

les estimations du Lasso prennent la forme suivante :

$$\hat{\beta}_j^{\,Lasso} = \begin{cases} y_j - \frac{\lambda}{2}, & \text{si } y_j > \frac{\lambda}{2}, \\ y_j + \frac{\lambda}{2}, & \text{si } y_j < -\frac{\lambda}{2}, \\ 0, & \text{si } |y_j| \leq \frac{\lambda}{2}. \end{cases} \tag{1.20}$$

Ce cas particulier illustre plus clairement le fonctionnement du Lasso dans un contexte simplifié, facilitant ainsi une meilleure compréhension de cette technique de régression régularisée.

1.6.3 Paramètre de régularisation

Le choix des paramètres de régularisation est crucial pour la performance des méthodes de régularisation. Il est donc essentiel de disposer d'une technique efficace pour les sélectionner. Dans la littérature, la validation croisée est la méthode la plus couramment utilisée à cet effet. Elle offre une approche simple pour estimer les paramètres de régularisation λ [34]. La régression pénalisée repose sur un paramètre d'ajustement λ, que l'utilisateur doit sélectionner avec soin. Ce choix est crucial, car il influence directement l'estimation des coefficients.

- Dans le cas de la régression Ridge, un λ trop grand entraîne un rétrécissement excessif des coefficients, réduisant ainsi la variance mais augmentant le biais.
- Pour le Lasso, un λ trop petit risque d'inclure trop de variables dans le modèle, tandis qu'un λ trop grand le rendra trop simplifié.

Il est donc essentiel de trouver un équilibre optimal pour ce paramètre, ce qui peut être réalisé grâce à la validation croisée[65].

1.6.4 La validation croisée avec k groupes

La validation croisée en k groupes consiste à diviser l'échantillon en k sous-ensembles de taille égale. À chaque itération, l'un de ces groupes est réservé pour la validation, tandis que les $k-1$ autres servent à l'apprentissage du modèle.

Par exemple, on utilise d'abord le premier groupe pour la validation, on ajuste le modèle sur les $k-1$ groupes restants, puis on calcule l'erreur MSE_1. Ce processus est ensuite répété pour chaque groupe, produisant ainsi les erreurs $\text{MSE}_2, \text{MSE}_3, \ldots, \text{MSE}_k$.

L'estimation finale, notée $CV_{(k)}$, est obtenue en prenant la moyenne des k erreurs calculées[34].

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} MSE_i \tag{1.21}$$

produisant une courbe d'erreur de validation croisée. On identifie le paramètre λ menant à la plus petite erreur de prédiction, soit [65]

$$\lambda_{\min} = \arg \min_{\lambda \in \{\lambda_1, \ldots, \lambda_m\}} \text{MSE}(\lambda)$$

et on ajuste un nouveau modèle à l'aide de ce paramètre λ_{min}, mais cette fois en utilisant toutes les observations.

Remarque. Cette approche de validation croisée nous permet d'évaluer efficacement la performance du modèle pour différentes valeurs de λ et de sélectionner celle qui minimise l'erreur de prédiction[12].

1.6.5 Interprétation géométrique de la régression Lasso

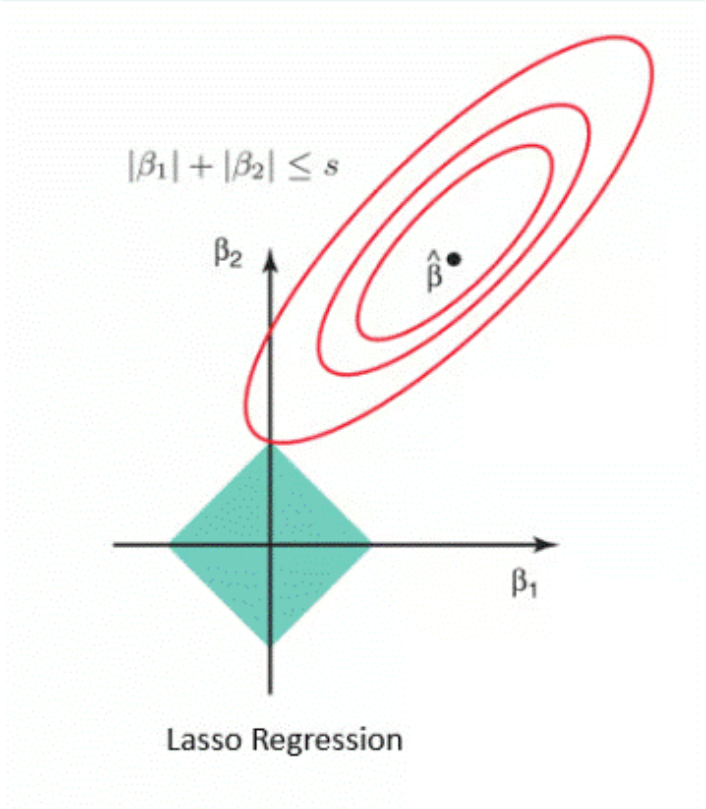

FIGURE 1.3 – Interprétation géométrique de la régression Lasso.

La figure 1.3 illustre la représentation géométrique de l'algorithme de descente du gradient. Elle considère uniquement deux variables explicatives, ce qui entraîne une contrainte de la forme $|\beta_1| + |\beta_2| < t$, représentée sous forme de losange pour le Lasso. Si la solution des moindres carrés se trouve à l'extérieur de ces régions, alors l'estimateur Lasso est obtenu au premier point de contact entre une ellipse de niveau constant du SRC et la région de contrainte. En dimension supérieure ($p > 2$), la contrainte de Lasso prend la forme d'un polytope, conservant sa capacité à sélectionner des variables en annulant certains coefficients [36].

À mesure que λ augmente, la région bleue du graphique se réduit. Lorsque l'un des coefficients atteint un sommet du losange, il est annulé, ce qui favorise la sélection de variables [65].

Remarque. La régression Ridge et le Lasso diffèrent dans leur manière d'imposer des contraintes aux coefficients. Ridge utilise une contrainte sphérique, ce qui entraîne des coefficients non nuls, tandis que le Lasso, avec ses sommets anguleux, favorise la mise à zéro de certains coefficients, facilitant ainsi la sélection de variables. Ces propriétés se généralisent aux dimensions supérieures[36].

1.6.6 Avantages de la méthode Lasso

La méthode Lasso, telle que présentée dans [34], présente plusieurs avantages notables :

- Elle favorise la parcimonie en éliminant les variables non pertinentes du modèle en attribuant une valeur nulle à leurs coefficients.
- C'est une méthode efficace pour sélectionner les variables ayant la plus forte contribution au modèle.
- Elle rétrécit les coefficients β vers zéro.

1.6.7 Inconvénients de la méthode Lasso

Les inconvénients présentés dans cette section s'appuient sur l'analyse développée dans [34].

- C'est une méthode inappropriée pour la sélection des groupes de prédicteurs. En effet, lorsque les prédicteurs sont fortement corrélés entre eux, la méthode Lasso sélectionne un seul prédicteur et pénalise les autres en leur attribuant des coefficients nuls.
- Dans le cas où $p > n$, l'approche Lasso sélectionne au maximum n variables.

1.7 Conclusion

Les régressions Ridge (ℓ_2) et Lasso (ℓ_1) sont des outils puissants de régularisation qui permettent d'améliorer la performance des modèles de régression linéaire, en particulier lorsque les données présentent une forte multicolinéarité ou un grand nombre de variables explicatives. La régression Ridge réduit la variance des estimations en appliquant une pénalité quadratique sur les coefficients, tandis que la régression Lasso favorise la sélection de variables en imposant une contrainte sur la somme des valeurs absolues des coefficients .

L'approche géométrique, quant à elle, a mis en évidence la différence entre les deux méthodes en termes de contraintes imposées sur les coefficients, expliquant pourquoi le Lasso est capable d'éliminer certaines variables tandis que Ridge tend à conserver tous les prédicteurs avec des valeurs réduites .

En termes d'application, Ridge est préférable lorsque toutes les variables ont une influence sur la réponse et que la multicolinéarité est un problème majeur, tandis que Lasso est plus adapté lorsque l'objectif est d'obtenir un modèle parcimonieux en éliminant les variables non pertinentes .

Ainsi, le choix entre Ridge et Lasso dépend du contexte des données et des objectifs de l'analyse. Une compréhension approfondie de leurs propriétés, de leurs interprétations et de leurs performances est essentielle pour les appliquer efficacement dans des problématiques réelles d'apprentissage statistique et de modélisation.

CHAPITRE 2

L'ALGORITHME ISTA POUR LA RÉGULARISATION EN ℓ_1

2.1 Introduction

L'algorithme ISTA (Iterative Shrinkage-Thresholding Algorithm) est une méthode itérative couramment utilisée pour traiter les problèmes inverses tout en adhérant à la contrainte de parcimonie. Il s'agit d'une méthode qui allie une descente de gradient à un opérateur de seuillage doux. Grâce à sa simplicité d'application et sa capacité à incorporer efficacement la régularisation en norme ℓ_1, cet instrument est indispensable dans le secteur de l'optimisation convexe, spécialement pour le traitement des signaux [55, 6, 22] et des images [42, 20, 35, 6].

L'algorithme ISTA a été théorisé par Daubechies, Defrise et De Mol dans leur article pionnier de 2004 [23]. Dans cette publication, ils présentent une technique itérative révolutionnaire pour traiter des problèmes inverses linéaires en respectant le principe de parcimonie. Ils basent leur méthode sur la réduction d'un critère de type

$$\min_x f(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1,$$

où A représente l'opérateur linéaire sous forme de matrice, b symbolise les données observées, tandis que x correspond à un vecteur de coefficients à déterminer dans une base orthonormée. Au lieu d'employer des pénalités quadratiques standard, nous introduisons ici une forme de sanction de type ℓ_1, privilégiant les solutions économes. L'algorithme utilise une itération de Landweber combinée à une opération de seuillage non linéaire (ou shrinkage), qui est appliquée à chaque étape.

Beck et Teboulle [5] ont été des acteurs majeurs en 2009 dans la formalisation et la propagation de l'algorithme ISTA, grâce à leur démonstration de sa convergence. Ils ont ensuite présenté FISTA (Fast Iterative Shrinkage-Thresholding Algorithm), une version optimisée conçue pour aborder les problèmes convexes non lisses.

ISTA (Iterative Shrinkage-Thresholding Algorithm) repose sur une itération simple combinant une descente de gradient sur la partie lisse de la fonction objectif et un opérateur proximal appliqué à la partie non différentiable [5]. Il est particulièrement

adapté aux problèmes d'optimisation de type Lasso, où l'on cherche des solutions parcimonieuses [61]. Grâce à sa structure élémentaire et à sa facilité de mise en œuvre, ISTA constitue une méthode robuste et largement utilisée, notamment en traitement d'image et en apprentissage statistique [50]. Cependant, malgré sa convergence garantie sous des conditions standard, sa vitesse de convergence reste sub-linéaire, de l'ordre de $O(1/k)$ [5], ce qui peut devenir un inconvénient dans les applications à grande échelle ou nécessitant une haute précision.

Gregor et LeCun (2010)[26] ont présenté *LISTA* (*Learned ISTA*), une variante apprise de l'algorithme ISTA, qui exploite les réseaux de neurones pour régler automatiquement les paramètres de l'optimisation. Cette méthode fait appel au *codage parcimonieux*, servant à exprimer les données sous forme de combinaisons linéaires dispersées. Ce modèle, qui s'inspire de la méthode de descente de coordonnées, fournit des résultats similaires tout en nécessitant dix fois moins de calculs. Il demeure entièrement différentiable, ce qui assure son intégration optimale aux architectures récentes d'apprentissage profond.

Par la suite, un certain nombre de chercheurs ont mené des études approfondies sur la convergence de l'ISTA et de ses variantes. Chambolle et Pock [18] ont présenté une étude approfondie des algorithmes de type primal-dual, dont ISTA est un exemple spécifique, en fixant des limites optimales de convergence dans un contexte plus large.

Plus récemment, des recherches telles que celles de O'Donoghue et Candes [49] ont exploré des stratégies d'arrêt adaptatif pour ISTA et FISTA, visant à combiner la rigueur théorique avec une meilleure efficacité numérique en pratique. Ces études ont permis de consolider l'utilisation de l'algorithme dans des domaines variés allant de la reconstruction d'images médicales à l'apprentissage automatique.
Ainsi, l'analyse approfondie de la convergence d'ISTA a joué un rôle fondamental dans l'évolution des méthodes d'optimisation parcimonieuses et continue d'inspirer de nouvelles approches, capables d'allier convergence rapide et robustesse numérique.

L'algorithme ISTA, apprécié pour sa simplicité et sa solidité, demeure un modèle dans le domaine de l'optimisation convexe [23], malgré ses limitations en termes de rapidité qui ont conduit à l'élaboration de versions plus efficaces comme FISTA [5] ou LISTA [26]. Il persiste à progresser pour s'ajuster à des problématiques de plus en plus complexes.

Il est largement utilisé dans diverses applications pratiques, notamment dans :

- L'analyse d'images médicales : notamment pour la reconstruction compressée en imagerie par résonance magnétique (IRM), où il aide à diminuer la durée de l'acquisition tout en maintenant la qualité des images [42, 20, 35, 6].

- Dans le domaine du traitement du signal, il est utilisé pour la séparation de signaux sous-échantillonnés ou leur débruitage, en raison de sa faculté à tirer des représentations parcimonieuses [22, 55, 6].

- En machine learning : l'ISTA est une technique traditionnelle de régression utilisée pour résoudre le Lasso, qui facilite la sélection automatique de variables dans des ensembles de données de haute dimension [61, 26, 19].

- Dans le cadre des réseaux neuronaux : l'algorithme LISTA, qui découle directement de l'ISTA, est utilisé pour acquérir de manière efficace des représentations parcimonieuses, particulièrement dans les domaines de la compression de données et de la super-résolution d'images [26, 58, 62].

Après avoir exposé l'évolution historique et les principales recherches liées au développement et à l'analyse de la convergence de l'algorithme ISTA, il est désormais primordial de détailler son fonctionnement. L'algorithme ISTA, particulièrement utilisé pour résoudre des problèmes inverses sous contraintes de parcimonie, se distingue par sa capacité à intégrer la régularisation en norme ℓ_1 de façon simple et efficace.

Dans le chapitre suivant, nous examinerons en détail la formulation de l'algorithme ISTA dans le contexte de la régularisation ℓ_1. Nous examinerons sa configuration mathématique, l'importance capitale du terme de régularisation ℓ_1 pour favoriser la parcimonie, ainsi que le processus itératif qui fusionne la descente de gradient et le seuillage doux. Cet examen approfondi aidera à saisir pourquoi cet algorithme est aujourd'hui une réponse indispensable dans de multiples domaines d'utilisation, y compris la manipulation d'images, la restitution de signaux et l'apprentissage automatique.

2.2 Préliminaires

Définition 2.2.1. [39] (Fonction convexe)
Une fonction continue $g = g(x)$, définie sur $\mathbb{R}^d$, est dite convexe si, pour tout $\lambda \in [0, 1]$, l'inégalité suivante est vérifiée :

$$g(\lambda x + (1 - \lambda)y) \leq \lambda g(x) + (1 - \lambda)g(y),$$

pour tous $x, y \in \mathbb{R}^d$. On note $\mathcal{F}_0(\mathbb{R}^d)$ la classe des fonctions convexes continues définies sur $\mathbb{R}^d$.

Définition 2.2.2. [39] Soit $g \in \mathcal{F}_0(\mathbb{R}^d)$. Un vecteur $v \in \mathbb{R}^d$ est appelé un sous-gradient de g en un point $x \in \mathbb{R}^d$ si, pour tout $y \in \mathbb{R}^d$, on a :

$$g(y) \geq g(x) + \langle v, y - x \rangle.$$

L'ensemble de tous les sous-gradients de g en x est appelé la sous-différentielle de g en x, et il est noté $\partial g(x)$.

Théorème 2.2.3. [39] (Fonction sous-differentielle) Soient $g_1, g_2 \in \mathcal{F}_0(\mathbb{R}^d)$. Alors, pour tout $x \in \mathbb{R}^d$, les sous-différentielles vérifient la relation suivante :

$$\partial(g_1 + g_2)(x) = \partial g_1(x) + \partial g_2(x),$$

où le membre de droite désigne la somme de Minkowski des ensembles $\partial g_1(x)$ et $\partial g_2(x)$.

Théorème 2.2.4. [39] Soit $g \in \mathcal{F}_0(\mathbb{R}^d)$. Si la fonction g est différentiable en un point $x \in \mathbb{R}^d$, alors le gradient $\nabla g(x)$ est le seul sous-gradient de g en x. En particulier, pour tout $y \in \mathbb{R}^d$, on a :

$$g(y) \geq g(x) + \langle \nabla g(x), y - x \rangle.$$

Définition 2.2.5. [24] (Fonction propre)
Une fonction de E dans $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ est dite propre si elle n'est pas identiquement égale à $+\infty$ et si elle ne prend pas la valeur $-\infty$.

Définition 2.2.6. [3](Semi-continuité inférieure) On dit qu'une fonction g est *semi-continue inférieurement* (s.c.i.) relativement à la topologie faible si pour toute suite $(x_n) \subset X$ telle que $x_n \rightharpoonup x$ (i.e. x_n converge faiblement vers x), on a :

$$\liminf_{n \to +\infty} g(x_n) \geq g(x),$$

c'est-à-dire :

$$\liminf_{n \to +\infty} g(x_n) = \sup_{n \geq 0} \inf_{m \geq n} g(x_n) = \sup\{\inf g(x_m)\}.$$

Remarque. [5] Soit $g : \mathbb{R}^n \to \mathbb{R}$ tel que :

- g est convexe, fermé, propre et possiblement non différentiable ;
- "fermé et propre" est une condition nécessaire pour que l'opérateur proximal soit bien défini ;
- "possiblement non différentiable" signifie que l'on doit remplacer le gradient par le sous-gradient.

2.3 Méthodes proximales

2.3.1 Opérateurs proximaux usuels

Soit F une fonction propre, convexe, définie sur un espace de Hilbert $\mathcal{H}$, à valeurs dans $]-\infty, +\infty[$, donnée par [13] :

$$F(x) = \|x\|_1,$$

et soit $\lambda > 0$, la i-ème composante de l'opérateur proximal de F s'écrit :

$$\mathcal{P}_{sF}(x)_i = \begin{cases} x_i + s & \text{si } x_i \leq -s, \\ 0 & \text{si } |x_i| \leq s, \\ x_i - s & \text{si } x_i \geq s. \end{cases}$$

2.3.2 Optimisation composite

Considérons une fonction convexe F définie sur un espace de Hilbert $\mathcal{H}$, qui se présente comme l'addition de deux fonctions f et g, toutes deux convexes, propres et semi-continues inférieurement. En outre, la fonction f est différentiable selon la définition de Fréchet, et son gradient possède une propriété de Lipschitzien de type L. De plus, il est présumé que pour chaque $\lambda > 0$, l'opérateur proximal de g peut être calculé.

Par la suite, nous allons examiner le problème suivant [17] :

$$\min_{x \in \mathcal{H}} F(x) = \min_{x \in \mathcal{H}} \{f(x) + g(x)\}. \tag{2.1}$$

L'optimisation lisse est un cas particulier d'optimisation composite, où le terme non lisse est identiquement nul. Un exemple particulier d'optimisation composite est le problème linéaire inverse avec représentation éparse, c'est-à-dire le modèle des moindres carrés (ou régression linéaire) avec ℓ_1 régularisation [40]

$$\min_x f(x) = \min_x \frac{1}{2}||Ax - b||_2^2 + \lambda||x||_1 \tag{2.2}$$

où $A \in \mathbb{R}^{m \times d}$ est une matrice de taille $m \times d$, $b \in \mathbb{R}^m$ est un vecteur de dimension m, et $\lambda > 0$ est un paramètre de régularisation permettant de faire le compromis entre la fidélité aux données mesurées et la sensibilité au bruit.

2.3.3 Aperçu de l'algorithme du gradient proximal

On souhaite résoudre le problème d'optimisation suivant [13] :

$$\min_{x \in \mathbb{R}^n} \{f(x) + g(x)\} \tag{2.3}$$

en supposant que :

- la fonction f est convexe, differentiable, definie sur tout $\mathbb{R}^n$,
- la fonction g est convexe, s.c.i., propre, pas forcement differentiable.

Pour cela, la methode du gradient proximal consiste a mettre en œuvre l'algorithme :

$$x_{k+1} = P_s(x_k - s\nabla f(x_k))$$

où $s > 0$ représente la longueur d'un pas, qui sera constante ou déterminée par des techniques de recherche linéaire, et l'opérateur proximal P_s correspond au seuilage

doux (soft-thresholding) appliqué composante par composante. Ainsi, pour chaque composante i, on a :

$$[x_{k+1}]_i = \begin{cases} (x_k - s\nabla f(x_k))_i + s\lambda, & \text{si } (x_k - s\nabla f(x_k))_i \leq -s\lambda \\ 0, & \text{si } |(x_k - s\nabla f(x_k))_i| \leq s\lambda \\ (x_k - s\nabla f(x_k))_i - s\lambda, & \text{si } (x_k - s\nabla f(x_k))_i \geq s\lambda \end{cases}$$

Par définition de l'opérateur proximal, cette méthode s'interprète comme la minimisation de la somme de la fonction G et d'une fonction quadratique représentant F au voisinage de x_k

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ g(x) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2s} \|x - x_k\|_2^2 \right\} \tag{2.4}$$

Proposition 2.3.1. [15] Pour tout $x_0 \in \mathcal{H}$ et pour tout $s \in \left]0, \frac{2}{L}\right[$, la suite définie par x_{k+1} converge faiblement vers un minimiseur x^* de F. De plus, on a :

$$F(x_k) - F(x^*) = O\left(\frac{1}{k}\right)$$

2.3.4 Algorithme du gradient proximal

L'une des principales réussites des techniques proximales est de faciliter la résolution de problèmes d'optimisation où la fonction à minimiser est une addition de deux fonctions, l'une étant lisse (c'est-à-dire différentiable) et l'autre simplement sous-différentiable, tout en disposant d'un opérateur proximal connu [13].

Algorithme 1 : Méthode du gradient proximal

1. Choisir un point initial $x_0 \in \mathbb{R}$. Poser $k = 0$.
2. A l'itération k de l'algorithme,
   - Choisir un pas $s > 0$,
   - Mettre à jour la variable x par :

$$x_{k+1} = P_s(x_k - s\nabla f(x_k)) \tag{2.5}$$

3. Passer de k à $k + 1$, puis retourner à l'étape 2 jusqu'à convergence.

Fin

2.3.5 Convergence de l'algorithme du gradient proximal

On suppose que la constante de Lipschitz L du gradient de la fonction f est connue. On choisit alors un pas constant donné par [52] :

$$s = \frac{1}{L} \tag{2.6}$$

On introduit l'opérateur gradient $\mathcal{G}_s$ défini par :

$$\mathcal{G}_s = \frac{1}{s}\left(x - \mathcal{P}_s\left(x - s\nabla f(x)\right)\right).$$

Cet opérateur permet d'écrire l'itération (2.5) du gradient proximal sous la forme :

$$x_{k+1} = x_k - s\mathcal{G}_s(x_k). \tag{2.7}$$

Théorème 2.3.2. [52] (Convergence) Sous l'hypothèse de convexité et de régularité standard, et avec le choix de pas $s = \frac{1}{L}$, la suite $\{x_k\}$ générée par l'algorithme ISTA définit une suite monotone décroissante $F(x_k)$ qui converge vers la valeur optimale $F^* = F(x^*)$. De plus, l'écart $F(x_k) - F^*$ est borné par une quantité arbitrairement petite en un nombre d'itérations d'ordre $O(1/k)$.

Démonstration [52]

1. **Inégalité fondamentale :** On suppose que la fonction f est différentiable avec un gradient Lipschitzien de constante L. Ainsi, pour tout $x, y \in \mathbb{R}^n$, on a :

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2 \tag{2.8}$$

En posant $y = x - s\mathcal{G}$, cette inégalite s'écrit :

$$f(x - s\mathcal{G}(x)_s) \leq f(x) - s\langle \nabla f(x), \mathcal{G}_s(x)\rangle + \frac{s^2 L}{2}\|\mathcal{G}_s(x)\|^2$$

Ajoutons $g(x - s\mathcal{G}_s(x)$ des deux côtés, on obtient :

$$f(x - s\mathcal{G}_s(x)) \leq f(x) - s\langle \nabla f(x), \mathcal{G}_s(x)\rangle + \frac{s^2 L}{2}\|\mathcal{G}_s(x)\|^2 + g(x - s\mathcal{G}_s(x))$$

Utilisons maintenant la convexité de f et g. Pour tout $y \in \mathbb{R}^n$, on a :

$$F(x) \leq F(y) + \langle \nabla f(x), x - y\rangle$$

d'autre part celle de G pour tout $r \in \partial g(x - s\mathcal{G}_s(x))$, alors :

$$g(x - s\mathcal{G}_s(x)) \leq g(y) + \langle r, x - s\mathcal{G}_s(x) - y\rangle$$

et choisissant $r = \mathcal{G}_s(x) - \nabla f(x)$ qui appartient à $\partial g(x - s\mathcal{G}_s(x))$, on obtient que, pour tout $x \in U$:

$$\begin{aligned} f(x - s\mathcal{G}_s(x)) &\leq f(y) + \langle \nabla f(x), x - y\rangle - s\langle \nabla f(x), \mathcal{G}_s(x)\rangle + \frac{s^2 L}{2}\|\mathcal{G}_s(x)\|^2 + g(y) \\ &\quad + \langle \mathcal{G}_s(x) - \nabla f(x), x - s\mathcal{G}_s(x) - y\rangle \\ &\leq F(y) + \langle \mathcal{G}_s(x), x - y\rangle + \left(\frac{s^2 L}{2} - s\right)\|\mathcal{G}_s(x)\|^2 \end{aligned}$$

Avec un choix de pas conforme à 2.6, on en déduit :

$$F(x - s\mathcal{G}_s(x)) \leq F(y) + \langle g(x), x - y\rangle - \frac{s}{2}\|\mathcal{G}(x)\|^2 \tag{2.9}$$

2. **Variation de la fonction objectif F** : Ecrivant l'inégalité 2.9 pour $x = y = x_k$, on obtient alors :
$$F(x_{(k+1)}) - F(x_{(k)}) \leq -\frac{s}{2}\|\mathcal{G}_s(x_{(k)})\|^2$$
On en deduit que la suite $\{F(x_k)\}_{n\in\mathbb{N}}$ décroissante, minorée par F^*, donc convergente.
3. **Distance à l'optimum :** Ecrivant l'inégalité 2.9 pour $x = x_k$ et $y = x^*$, on obtient :
$$F(x_{k+1}) - F^* \leq \langle \mathcal{G}_s(x_k), x_k - x^*\rangle - \frac{s}{2}\|\mathcal{G}_s(x_k)\|^2$$
$$F(x^*) + \frac{1}{2s}(\|x_k - x^*\|^2 - \|x_k - x^* - s\mathcal{G}_s(x_k)\|^2$$
et donc :
$$F(x_{k+1}) - F^* \leq \frac{1}{2s}(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2). \tag{2.10}$$
On en deduit en particulier que $\|x_{k+1} - x^*\| \leq \|x_k - x^*\|$: la distance de x_k à toute solution optimale x^* décroit au cours des itérations de l'algorithme.
4. **Vitesse de convergence :** Sommant les inégalités 2.10 entre les indices 1 et k, on obtient
$$\sum_{i=0}^{k}(F(x_i) - F^*) \leq \frac{1}{2s}(\|x_0 - x^*\|^2 - \|x_k - x^*\|^2) \leq \frac{1}{2s}\|x_0 - x^*\|^2.$$
La suite $F(x_k)_{n\in\mathbb{N}}$ étant décroissante, $F(x_k) - F^* \leq F(x_i) - F^*$ pour tout $i \leq k$, d'où :
$$F(x_k) - F^* = \frac{1}{2sk}\|x_0 - x^*\|^2$$
On déduit que la suite $F(x_k)_{n\in\mathbb{N}}$ converge vers F^*, et que l'écart $(F(x_k) - F^*)$ est majorée par une quantité de l'ordre $O(\frac{1}{k})$, ce qui signifie qu'il peut devenir arbitrarement petit pour un nombre d'itération k suffisamment grand.

Remarque. [52] Dans le cas où la constante de Lipschitz du gradient de la fonction F n'est pas connue, on peut quand même mettre en œuvre l'algorithme du gradient proximal en ajustant le pas s par une recherche linéaire de type Armijo. Plus précisément, initialisant l'itération k avec un pas s assez grand, on effectue un pas de l'algorithme :

$$x^+ = x_k - s\mathcal{G}_s(x_k)$$

On valide ce résultat, c'est-à-dire on passe à l'itération suivante en fixant $x_{k+1} = x^+$,

si la condition de décroissance :

$$f(x^+) \leq f(x_k) - s\langle \mathcal{G}_s(x), \nabla f(x_k)\rangle + \frac{s}{2}\|\mathcal{G}_s(x_k)\|^2$$

est vérifiée. Dans le cas contraire, on fait décroître le pas s en le multipliant par un coefficient $\beta \in]0,1[$ et on recalcule x^+. Cet algorithme qui incorpore la recherche linéaire dans la méthode du gradient proximal converge dans les mêmes conditions que l'algorithme à pas fixe. On pourra consulter les détails de cet algorithme dans [Beck and Teboulle (2010)][5].

Remarque. [52] On retrouve l'algorithme du point proximal à partir de l'algorithme du gradient proximal en l'appliquant au cas où la partie différentiable f du critère est identiquement nulle, et donc $f = g$. Alors, l'algorithme du point proximal

$$x_{k+1} = P_s(x_k)$$

converge pour le choix de pas s, pourvu que l'on ait $s > 0$.

2.3.6 Le modèle d'approximation de base

Résultats principaux

La méthode de descente de gradient constitue le noyau central de l'algorithme ISTA. Voici le problème d'optimisation sans contrainte [41] :

$$\min_x \{F(x) = f(x)\}$$

où $f(x)$ est la formule $\frac{1}{2}\|Ax - b\|^2$. Supposons que $f(x)$ soit continuellement différentiable. S'il existe une valeur suffisamment petite $s > 0$ telle que $x_{k+1} = x_k - s\nabla F(x_k)$, alors [41, 4] :

$$F(x_k) \geq F(x_{k+1}) \tag{2.11}$$

La valeur x_k correspondant à la valeur minimale de la fonction $f(x)$ peut être obtenue en itérant à travers les étapes suivantes :

$$x_k = x_{k-1} - s\nabla f(x_{k-1}), \quad x_0 \in \mathbb{R}^n$$

L'algorithme ISTA suppose que $f(x)$ satisfait la condition de continuité de Lipschitz, c'est-à-dire que la dérivée de $f(x)$ a une borne inférieure, et la borne inférieure minimale est appelée la constante de Lipschitz $L(f)$. À ce moment-là, pour tout $L \geq L(f)$, il existe [5, 41] :

$$f(x) \leq f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2, \quad x, y \in \mathbb{R}^n \tag{2.12}$$

D'après l'équation 2.12, la valeur de la fonction peut être approximée près du point x_k [41, 39] :

$$\hat{f}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2}\|x - x_k\|^2 \tag{2.13}$$

En adoptant cette même idée de gradient de base pour le problème régularisé ℓ_1 non lisse :

$$\min_x \{F(x) = f(x) + \lambda \|x\|_1\}$$

Après avoir introduit la fonction dans la pénalité, la valeur de la fonction peut être approximée au point x_k [41] :

$$\hat{F}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2}\|x - x_k\|^2 + \lambda \|x\|_1 \tag{2.14}$$

2.3.7 Algorithme ISTA : Iterative Soft-Thresholding Algorithm

La solution du sous-problème proximal est définie composante à composante en fonction du pas de la descente de gradient. Ce résultat est au cœur de la méthode proximale pour les problèmes avec régularisation ℓ_1, appelée ISTA (pour Iterative Soft-Thresholding Algorithm) : une description de cette méthode est donnée par l'algorithme 2 [53].

Algorithme 2 : Algorithme ISTA : Iterative Soft-Thresholding Algorithm

1 **Initialisation :** $x_0 \in \mathbb{R}^d$.

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

1. Calculer le gradient de la partie lisse du problème $\nabla f(x_k)$.
2. Définir une taille de pas $s > 0$.
3. Calculer le nouvel itéré x_{k+1} composante par composante selon la formule :

$$[x_{k+1}]_i = \begin{cases} [x_k - s\nabla f(x_k)]_i + s\lambda & \text{si } [x_k - s\nabla f(x_k)]_i < -s\lambda \\ [x_k - s\nabla f(x_k)]_i - s\lambda & \text{si } [x_k - s\nabla f(x_k)]_i > s\lambda \\ 0 & \text{si } [x_k - s\nabla f(x_k)]_i \in [-s\lambda, s\lambda] \end{cases} \tag{2.15}$$

Fin

La définition de la fonction de " soft-thresholding " force certaines des composantes du nouvel itéré à être nulles, ce qui produira au final une solution plus parcimonieuse que pour un problème non régularisé.

Remarque. [17] La preuve de la convergence faible vers un minimiseur repose sur le fait que l'opérateur $\mathcal{P}_{sF}(x)_i$ est fermement non expansif et que les points fixes de P_s sont les minimiseurs de F.

On peut noter que chaque étape de cet algorithme se réduit à un seuillage doux et à une descente de gradient à pas fixe si la fonction g est une norme ℓ_1. C'est pour cette raison que ce même algorithme est parfois appelé *Iterative Soft Thresholding Algorithm* ou *ISTA*. Cette dénomination est fréquente dans la communauté Image.

2.3.8 Applications de l'algorithme ISTA

L'algorithme ISTA est largement utilisé dans divers domaines où la recherche de solutions parcimonieuses est essentielle. En traitement d'images [42, 20, 35, 6], il est appliqué pour la reconstruction compressée dans des contextes tels que l'IRM (Imagerie par Résonance Magnétique), où il permet de réduire considérablement le temps d'acquisition des données tout en maintenant une qualité d'image acceptable.

Dans le domaine du traitement du signal [22, 55, 6], ISTA est utilisé pour la détection et la séparation de signaux sous-échantillonnés, notamment dans les télécommunications et la radio logicielle.

Enfin, en statistiques et apprentissage automatique, ISTA est exploité pour la résolution de problèmes de régression Lasso [61, 26, 19], favorisant la sélection automatique de variables explicatives pertinentes lorsque les données sont nombreuses et potentiellement corrélées.

2.3.9 Limites de l'algorithme ISTA

Malgré sa simplicité d'implémentation et sa robustesse théorique, l'algorithme ISTA présente certaines limites[5]. L'un de ses inconvénients majeurs est sa convergence relativement lente, notamment lorsque le problème nécessite une grande précision ou que la dimension des données est élevée[61].

De plus, la performance d'ISTA dépend fortement du choix du paramètre L, représentant la constante de Lipschitz, qui peut être difficile à estimer correctement dans la pratique[5].

Ces limitations ont conduit au développement d'algorithmes améliorés tels que FISTA (Fast Iterative Shrinkage-Thresholding Algorithm), qui accélère la convergence[5].

2.4 Conclusion

Cette section a présenté l'algorithme ISTA, une méthode issue de l'optimisation proximale, largement utilisée pour les problèmes de type Lasso. Grâce à sa structure simple, combinant une descente de gradient et seuillage doux, ISTA permet d'optimiser efficacement des fonctions avec une composante lisse et une régularisation ℓ_1.

Nous avons mis en évidence ses atouts majeurs, comme sa facilité de mise en œuvre , sa convergence assurée , et sa capacité à s'adapter à divers types de régularisation . Néanmoins, des limites importantes demeurent, dont une convergence sous-linéaire en $O(1/k)$, une sensibilité au choix du pas , ainsi qu'un handicap de performance comparé à des algorithmes plus rapides tels que FISTA .

Ces constats ont mené à l'émergence de variantes accélérées d'ISTA, dont FISTA (*Fast Iterative Shrinkage-Thresholding Algorithm*) constitue l'une des plus notables. Inspiré des travaux de Nesterov, FISTA intègre un mécanisme d'accélération qui améliore considérablement la vitesse de convergence par rapport à ISTA.

Dans le prochain chapitre, nous allons nous pencher plus en détail sur l'analyse de FISTA, en soulignant ses bases théoriques, sa formulation algorithmique et les améliorations tangibles qu'il offre comparativement à ISTA.

CHAPITRE 3

L'algorithme FISTA pour la régularisation en ℓ_1

3.1 Introduction

L'algorithme FISTA (*Fast Iterative Shrinkage-Thresholding Algorithm*) fait partie des techniques de premier ordre destinées à résoudre les problèmes d'optimisation convexe non lisse. Il constitue une percée significative dans le secteur de l'optimisation parcimonieuse, en proposant une convergence plus rapide par rapport à son prédécesseur, l'algorithme ISTA (*Iterative Shrinkage-Thresholding Algorithm*).

Les problèmes d'optimisation régularisés par des sanctions de type ℓ_1, tels que le Lasso, sont largement présents dans des domaines tels que l'apprentissage automatique, le traitement du signal, la statistique et même l'imagerie médicale. Les méthodes proximales se sont établies comme des instruments standard dans la résolution efficace de ces problématiques. ISTA a été un tournant grâce à sa facilité d'implémentation, mais sa convergence lente est vite devenue un obstacle pour des applications à grande échelle [61, 23].

Pour surmonter cette contrainte, FISTA a été suggéré comme une réponse raffinée, alliant les bénéfices de l'optimisation proximale à une approche d'accélération tirée des recherches de Nesterov [47]. Cela aboutit à une méthode qui peut atteindre une convergence théorique en $O(1/k^2)$, tout en étant facile à appliquer. FISTA conserve l'approche d'ISTA qui consiste en une descente de gradient $x_{k+1} = x_k - s\nabla f(x_k)$ sur f suivie d'une opération proximale (seuillage) sur g.

L'origine théorique de FISTA repose sur les recherches de Nesterov (1983) [47], ayant présenté une technique d'accélération du gradient avec un taux de convergence optimal pour les fonctions convexes. Beck et Teboulle (2009)[5] ont par la suite modifié cette idée en l'appliquant à l'optimisation non lisse grâce à la mise en place de FISTA [47].

Leur travail a consisté à intégrer l'accélération de Nesterov dans le cadre des problèmes de la forme suivante :

$$\{\min_{x\in\mathbb{R}^n}\{F(x) = f(x) + g(x)\} \tag{3.1}$$

où f représente une fonction convexe et lisse dotée d'un gradient Lipschitzien, tandis que g est une fonction convexe propre, possiblement non différentiable. FISTA se distingue par l'introduction d'une extrapolation entre les itérations, suivie d'une phase de minimisation proximale, assurant une accélération sans nuire à la stabilité.

Beck et Teboulle [5] ont prouvé que FISTA atteint une vitesse de convergence en $O(1/k^2)$ concernant la valeur de la fonction objectif, ce qui représente une performance optimale pour les méthodes de premier ordre. Par la suite, d'autres chercheurs ont poursuivi cette analyse de manière plus approfondie. Par exemple, Chambolle et Dossal (2015) [17] ont démontré que dans certaines situations, la série d'itérations produite par FISTA converge réellement vers une solution optimale, ce qui n'était pas assuré dans la version initiale [5]. Ces améliorations ont renforcé les garanties de convergence tout en maintenant l'accélération, suivies par d'autres variantes telles que mFISTA (FISTA monotone).

De plus, O'Donoghue et Candès (2015)[49] ont suggéré l'algorithme Adaptive Restart FISTA qui adapte de manière dynamique l'inertie dans le but d'optimiser la convergence pratique, notamment lorsque le problème devient fortement convexe.

Grâce à sa convergence rapide et sa facilité d'implémentation, l'algorithme FISTA trouve une multitude d'applications dans divers domaines. Dans le domaine du traitement d'image, il est utilisé pour des opérations complexes comme la restauration, l'élimination du bruit ou la déconvolution, facilitant ainsi l'obtention d'images claires à partir de données altérées ou manquantes [47]. Dans le contexte de l'IRM compressée, FISTA occupe une position centrale dans la reconstitution d'images médicales de haute définition à partir d'un ensemble limité de mesures, en accord avec les principes de l'échantillonnage compressé [42, 20, 35, 6].

Dans le domaine de l'apprentissage automatique[61, 26, 19], il est employé dans la régression Lasso pour le choix des variables, mais également pour la classification parcimonieuse et la réduction de dimension, grâce à son aptitude à privilégier les solutions dispersées [23]. Pour finir, en ce qui concerne le traitement du son, FISTA joue un rôle dans la séparation des sources de son ou le codage parcimonieux, contribuant ainsi à une analyse précise des signaux acoustiques [22, 55, 6].

Sa performance, sa rapidité de convergence et sa faculté à s'ajuster à diverses pénalités convexes (telles que $\ell_{1,2}$ ou $\ell_{1,\infty}$) font de lui un algorithme indispensable dans les problématiques d'optimisation parcimonieuse.

S'appuyant sur les bases établies par l'algorithme ISTA, il est logique d'explorer ses extensions destinées à pallier ses restrictions, notamment sa lenteur de convergence. L'algorithme FISTA (Fast Iterative Shrinkage-Thresholding Algorithm), qui intègre une technique d'accélération dérivée des travaux de Nesterov, s'est distingué comme une solution efficace parmi ces améliorations. Ce chapitre se focalise donc sur une analyse approfondie de FISTA : nous exposerons son fondement théorique, les résultats essentiels liés à sa convergence, et ses multiples utilisations dans des situations pratiques comme le traitement d'images, l'IRM compressée ou encore l'apprentissage automatique.

3.2 FISTA : Fast Iterative Shrinkage-Thresholding Algorithm

L'algorithme FISTA (*Fast Iterative Shrinkage-Thresholding Algorithm*) repose sur une amélioration majeure d'ISTA en y introduisant une composante d'inertie pour accélérer la convergence. L'idée fondamentale est d'appliquer l'opérateur

$$T = P_s(x - s\nabla f(x)), \tag{3.2}$$

où P_s désigne l'opérateur proximal associé à la fonction g, tout en exploitant une extrapolation des itérés précédents à la manière des méthodes de Nesterov. Cette approche permet d'améliorer les bornes sur la vitesse de convergence, passant de $O(1/k)$ pour ISTA à $O(1/k^2)$ pour FISTA [24]. Grâce à cette accélération, FISTA s'avère particulièrement efficace pour résoudre des problèmes d'optimisation parcimonieuse de grande dimension.

Définition 3.2.1. [40] Peut-être que la méthode du premier ordre la plus simple est la descente de gradient classique, qui remonte à Cauchy[14][1847]. En prenant une taille de pas fixe $s > 0$, la descente de gradient classique est mise en œuvre selon la règle récursive suivante :

$$x_k = x_{k-1} - s\nabla f(x_{k-1})$$

Étant donné un point initial $x_0 \in \mathbb{R}^d$, la méthode possède un taux de convergence global donné par :

$$F(x_k) - F(x^\star) \leq O\left(\frac{1}{k}\right) \tag{3.3}$$

L'idée moderne d'accélération commence avec Polyak (1964)[51], qui propose sa méthode du boulet de canon (ou méthode du moment) afin d'accélérer localement la vitesse de convergence. Cette approche repose sur le théorème de la variété invariante issu du domaine des systèmes dynamiques. L'une des étapes marquantes dans ce domaine est l'introduction de la méthode de descente de gradient accélérée de Nesterov.

$$\begin{cases} x_k = y_{k-1} - s\nabla f(y_{k-1}), \\ y_k = x_k + \dfrac{k-1}{k+r}(x_k - x_{k-1}), \end{cases}$$

Avec un point initial $x_0 = y_0 \in \mathbb{R}^d$, le cas particulier $r = 2$ a été initialement proposé par Nesterov [46]. Dans ce cas, le taux de convergence global est également accéléré, atteignant :

$$F(x_k) - F(x^*) = O\left(\frac{1}{k^2}\right),$$

où x^* est une solution optimale du problème.

3.2.1 Accéleration de l'algorithme du gradient proximal

Il existe une version accélérée de l'algorithme du gradient proximal, appelée *FISTA* (Fast Iterative Shrinkage-Thresholding Algorithm). Cet algorithme, dont l'analyse complète est présentée dans [5] et [6], est brièvement exposé ici. Il s'applique au même problème, et sous les mêmes hypothèses que l'algorithme standard du gradient proximal[13].

Algorithme 3 : Version accélérée de la méthode du gradient proximal

1. Choisir un point initial $x_0 = y_0$.
2. A l'itération k de l'algorithme,
   - Choisir un pas $s > 0$,
   - Mettre à jour le couple (x,y) par :

$$\begin{cases} x_{k+1} = P_s(y_k - s\nabla F(y_k)), \\ y_{k+1} = x_{k+1} + \dfrac{k-1}{k+2}(x_{k+1} - x_k), \end{cases}$$

3. Passer de k à $k+1$, puis retourner à l'étape 2 jusqu'à convergence.

Fin

3.2.2 Convergence de la suite des itéres

Définition 3.2.2. [17, 24, 16] FISTA est défini par une suite $(t_k)_{k\in\mathbb{N}}$ de réels strictement supérieurs à 1, et par un point initial $x_0 \in \mathbb{R}^d$. Soit $(t_k)_{k\geq 1}$ une suite de réels strictement positifs, et soit $x_0 \in \mathbb{R}^d$. Les suites $(x_k)_{k\in\mathbb{N}}, (y_k)_{k\in\mathbb{N}}$ et $(u_k)_{k\in\mathbb{N}}$ sont définies par : $y_0 = u_0 = x_0$, et pour tout $k \geq 1$,

$$x_k = T(y_{k-1}),$$

$$u_k = x_{k-1} + t_k(x_k - x_{k-1}),$$

$$y_k = \left(1 - \frac{1}{t_{k+1}}\right)x_k + \frac{1}{t_{k+1}}u_k.$$

Le point y_k peut aussi être défini à partir des points x_k et x_{k-1} par :

$$y_k = x_k + \alpha_k(x_k - x_{k-1}) \quad \text{avec} \quad \alpha_k = \frac{t_k - 1}{t_{k+1}} \tag{3.4}$$

Pour des choix adéquats de $(t_k)_{k\geq 1}$, la suite $(F(x_k))_{k\in\mathbb{N}}$ converge vers $F(x^*)$, c'est-à-dire que la suite $(w_k)_{k\in\mathbb{N}}$, définie de la manière suivante :

$$w_k = F(x_k) - F(x^*) \geq 0$$

vérifie $w_k \to 0$ lorsque $k \to \infty$. Plusieurs preuves utilisent la variation locale de la suite $(x_k)_{k\in\mathbb{N}}$, que nous noterons δ_k :

$$\delta_k = \frac{1}{2}\|x_k - x_{k-1}\|^2$$

La suite $(v_k)_{k\in\mathbb{N}}$ décrivant la distance entre x_k et un minimiseur x^* de F sera aussi utile :

$$v_k = \frac{1}{2}\|x_k - x^*\|^2$$

Pour compléter ces notations, nous définissons également la suite $(\delta_k)_{k\geq 2}$, associée à $(t_k)_{k\geq 1}$, dont la positivité assure la vitesse de convergence de FISTA :

$$\rho_k = t_{k-1}^2 - t_k^2 + t_k$$

Lemme 3.2.3. [17][24] Soit $s \in \left]0, \frac{1}{L}\right[$, où L est la constante de Lipschitz de ∇f, et soit $\bar{x} \in \mathbb{E}$ tel que $\hat{x} = T\bar{x}$. Alors :

$$\forall x \in \mathbb{E}, F(\hat{x}) + \frac{\|\hat{x} - x\|^2}{2s} \leq F(x) + \frac{\|x - \bar{x}\|^2}{2s} \tag{3.5}$$

En appliquant ce lemme aux points $x = y_k$, $x = x_{k+1}$ et $x = x_n$, on peut conclure que :

$$F(x_{k+1}) + \frac{\|x_{k+1} - x_k\|^2}{2s} \leq F(x_k) + \frac{\alpha_k\|x_k - x_{k-1}\|^2}{2s} \tag{3.6}$$

Cette inégalité implique que, si les α_k sont des éléments de $[0, 1]$, alors pour tout $k \geq 1$, on a :

$$w_{k+1} + \delta_{k+1} \leq w_k + \delta_k \tag{3.7}$$

C'est-à-dire que la suite $(w_k + \delta_k)_{k\in\mathbb{N}}$ est décroissante.

3.2.3 Résultats de base et complémentaires sur FISTA

Théorème 3.2.4. [24, 17, 16] Pour tout $x_0 \in \mathbb{E}$, si la suite $(t_k)_{k\in\mathbb{N}}$ vérifie

$$\forall k \geq 2, \quad t_k^2 - t_k \leq t_{k-1}^2 \tag{3.8}$$

Et si $t_1 = 1$ et $s \leq \frac{1}{L}$, alors pour tout $N \geq 2$, on a :

$$t_{N+1}^2 w_{N+1} + \sum_{k=1}^{N} \rho_{k+1} w_k \leq \frac{v_0 - k v_{N+1}}{s}. \tag{3.9}$$

Démonstration [24, 16, 17] En appliquant le lemme 3.2.3 avec $\bar{x} = y_k$, $\hat{x} = x_{k+1}$, et $x = \left(1 - \frac{1}{t_{k+1}}\right) x_k + \frac{1}{t_{k+1}} x^*$, on obtient :

$$F(x_{k+1}) + \frac{\left\| \frac{1}{t_{k+1}} u_{k+1} - \frac{1}{t_{k+1}} x^* \right\|_2^2}{2s} \leq F\left(\left(1 - \frac{1}{t_{k+1}}\right) x_k + \frac{1}{t_{k+1}} x^*\right) + \frac{\left\| \frac{1}{t_{k+1}} x^* - \frac{1}{t_{k+1}} u_k \right\|_2^2}{2s}.$$

En utilisant la convexité de F, on a

$$F(x_{k+1}) - F(x^*) - \left(1 - \frac{1}{t_{k+1}}\right)(F(x_k) - F(x^*)) \leq \left(\frac{\|u_k - x^*\|_2^2}{2st_{k+1}^2} - \frac{\|u_{k+1} - x^*\|_2^2}{2st_{k+1}^2}\right)$$

En utilisant les définitions de w_k et v_k, ces inégalités peuvent être reformulées de la manière suivante :

$$t_{k+1}^2 w_{k+1} - (t_{k+1}^2 - t_{k+1}) w_k \leq \frac{v_k - v_{k+1}}{s} \tag{3.10}$$

En sommant ces inégalités de $k = 1$ à $k = N$, on obtient :

$$t_{N+1}^2 w_{N+1} + \sum_{k=1}^{N} \rho_{k+1} w_k \leq \frac{v_0 - v_{N+1}}{s}, \tag{3.11}$$

Une des consequences directes de ce théorème est ainsi l'inégalité suivante valable pour tout $k \geq 1$ [17][24] :

$$F(x_k) - F(x^*) \leq \frac{\|x_0 - x^*\|^2}{2t_k^2 s}. \tag{3.12}$$

Remarque. [24][17]

De ce théorème 3.2.4, on peut déduire différents résultats selon les choix de la suite $(t_k)_{k \in \mathbb{N}}$.

1. Si $(t_k)_{k \in \mathbb{N}}$ est la suite constante égale à 1, la suite $(\alpha_k)_{k \in \mathbb{N}}$ est la suite nulle, et l'algorithme se réduit à l'algorithme de ISTA. La suite $(\rho_k)_{k \geq 2}$ est alors la suite constante égale à 1. On déduit que la série de terme général $(w_k)_{k \in \mathbb{N}}$ est convergente. Comme on sait de plus que la suite $(w_k)_{k \in \mathbb{N}}$ est décroissante, on en déduit que $w_k = O\left(\frac{1}{k}\right)$.
2. Le choix original de Nesterov[47] ainsi que celui de Beck et Teboulle[5] est celui qui optimise la borne dans 3.12, c'est-à-dire le choix de la suite $(t_k)_{k \in \mathbb{N}}$ qui maximise t_k à chaque étape en respectant la condition 3.8. Autrement dit, c'est le choix :

$$t_{k+1} = \sqrt{t_k^2 + \frac{1}{2}} + \frac{1}{2}. \tag{3.13}$$

On peut alors montrer que $t_k \geq \frac{k+1}{2}$, et que $\lim_{k \to \infty} \frac{k}{2t_k} = 1$, ce qui signifie que t_k est

de l'ordre de $\frac{k}{2}$. Dans ce cas, l'inégalité 3.12 devient :

$$F(x_k) - F(x^*) \leq \frac{2\|x_0 - x^*\|^2}{(k+1)^2 s}. \tag{3.14}$$

3. On a la même vitesse de décroissance et la même borne si l'on utilise $t_k = \frac{k+1}{2}$, ce qui correspond au choix $\alpha_k = \frac{k-1}{k+2}$.
4. Si l'on choisit $t_k = \frac{k+a-1}{a}$ avec $a > 2$, l'inégalité 3.12 devient :

$$F(x_k) - F(x^*) \leq \frac{a^2\|x_0 - x^*\|^2}{2(k+a-1)^2 s}. \tag{3.15}$$

Ce qui est localement moins bon que le choix de Nesterov[47] ou celui $t_k = \frac{k+1}{2}$, en revanche, dans ce cas, la suite ρ_k est de l'ordre de k, et l'on déduit du théorème que la série de terme général kw_k est sommable, ce qui permet d'obtenir un comportement asymptotique de la suite meilleur que celui induit par la borne obtenue via (3.14) ou (3.15) pour un k donné.

Théorème 3.2.5. [24] Si $t_k = \frac{k+a-1}{a}$ avec $a > 2$, alors

1. La suite de terme général kw_k est sommable, et : $w_k = o\left(\frac{1}{k^2}\right)$.
2. La suite de terme général $k\delta_k$ est sommable, et $\delta_k = o\left(\frac{1}{k^2}\right)$.
3. La suite $(x_k)_{k\in\mathbb{N}}$ converge vers un minimiseur x^* de la fonction F.

Démonstration [24, 16, 17]
Il n'existe pas de preuve directe de convergence de la suite $(x_k)_{k\in\mathbb{N}}$ pour le choix original de Nesterov. La preuve de convergence repose en réalité sur la sommabilité des suites de terme général kw_k et $k\delta_k$. La sommabilité de la série de terme général kw_k est une conséquence directe du Théorème 3.2.4, car pour ce choix de t_k, on a :

$$\rho_k = \frac{1}{a^2}\left((a-2)k + a^2 - 3a + 3\right)$$

qui est de l'ordre de k dès que $a > 2$.
Pour démontrer la convergence de la série de terme général $k\delta_k$, on reprend l'inégalité (3.6), qui peut se réécrire de la manière suivante :

$$\delta_{k+1} - \alpha_k^2\delta_k \leq s(w_k - w_{k+1})$$

Si $t_k = \frac{k+a-1}{a}, \alpha_k = \frac{t_k - 1}{t_{k+1}} = \frac{k-1}{k+a}$

En multipliant les inégalités par $(k+a)^2$ et en sommant de $k = 1$ à $k = N$, on obtient :

$$\sum_{k=1}^{N}(k+a)^2(\delta_{k+1} - \alpha_k^2\delta_k) \leq s\sum_{k=1}^{N}(k+a)^2(w_k - w_{k+1}),$$

ce qui donne

$$(N+a)^2\delta_{N+1} + \sum_{k=2}^{N}((k+a-1)^2 - (k+a)^2\alpha_k^2)\delta_k \leq$$

$$s\left((a+1)^2 w_1 - (N+a)^2 w_{N+1} + \sum_{k=2}^{N}((k+a)^2 - (k+a-1)^2)w_k\right)$$

C'est-à-dire

$$(N+a)^2\delta_{N+1} + \sum_{k=2}^{N} a(2k-2+a)\delta_k \leq s\left((a+1)^2 w_1 - (N+a)^2 w_{N+1} + \sum_{k=2}^{N}(2k+2a-1)w_k\right)$$

D'après le premier point, le membre de droite de cette inégalité est borné indépendamment de N, ce qui assure que la suite $(k\delta_k)_{k\in\mathbb{N}}$ appartient à $\ell_1(\mathbb{N})$. On en déduit également que f, $N^2\delta_{N+1}$ est borné, mais en fait on a mieux que cela.

En effet, d'après (3.7), la suite de terme général $w_k + \delta_k$ est décroissante, or la série de terme général $n(w_k + \delta_k)$ est convergente. Ceci implique que : $w_k = o\left(\frac{1}{k^2}\right)$ et $\delta_k = o\left(\frac{1}{k^2}\right)$, ce qui conclut la preuve du théorème.

3.2.4 Algorithme FISTA : Fast Iterative Soft-Thresholding Algorithm

Dans cette section, nous étendons la méthode proposée dans [47] au modèle général (3.1) et nous établissons un résultat amélioré en termes de complexité. Nous commençons par présenter l'algorithme avec un pas constant[6].

Algorithme 4 : FISTA avec pas constant

1 **Entrée :** $L = L(f)$ — constante de Lipschitz du gradient ∇f.

2 **Étape 0 :** Prendre $y_1 = x_0 \in \mathbb{R}^n$, $t_1 = 1$.

3 **Étape k :** $k \geq 1$, **faire** :

$$x_k = \text{prox}_s^{(g)}\left(y_k - \frac{1}{L}\nabla f(y_k)\right), \tag{3.16}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \tag{3.17}$$

$$y_{k+1} = x_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_k - x_{k-1}). \tag{3.18}$$

La principale différence entre l'algorithme ci-dessus et ISTA réside dans le fait que l'étape de seuillage itératif (équation (3.16)) n'est pas appliquée au point précédent x_{k-1}, mais plutôt au point y_k, lequel est obtenu à partir d'une combinaison linéaire très spécifique des deux points précédents $\{x_{k-1}, x_k\}$. Il est évident que l'effort de calcul principal dans ISTA et FISTA demeure similaire. Les calculs supplémentaires requis dans FISTA aux étapes (3.17) et (3.18) restent clairement marginaux.

3.2.5 Comparaison entre ISTA et FISTA

Le tableau 3.1 est inspiré des travaux de Beck et Teboulle (2009)[5] et de Parikh et Boyd (2014)[50], qui présentent les différences fondamentales entre les algorithmes ISTA et FISTA en termes de vitesse de convergence, complexité, stabilité et applications.

Critère	**ISTA**	**FISTA**
Vitesse de convergence	$O(1/k)$	$O(1/k^2)$
Principe de base	Descente de gradient suivie d'un seuillage	Terme d'inertie de type Nesterov ajouté
Stabilité	Plus stable	Moins stable, peut osciller
Complexité par itération	Faible	Faible (quasi identique à ISTA)
Facilité de mise en œuvre	Très simple	Un peu plus complexe (mise à jour de t_k et y_k)
Applications typiques	Lasso, traitement du signal, parcimonie	Pareil, avec meilleures performances
Paramètres requis	Constante de Lipschitz L	Même exigence, plus sensible
Avantages	Simplicité, stabilité	Convergence accélérée
Inconvénients	Lenteur pour grands problèmes	Instabilité possible, plus de réglages

TABLE 3.1 – Comparaison entre ISTA et FISTA

une nouvelle expérience numérique a été réalisée en ajoutant l'algorithme FISTA à des fins de comparaison, comme illustré dans la Figure 3.1. De manière similaire, nous avons également ajouté une ligne noire en pointillés servant de référence pour caractériser le pire taux de convergence de la norme du gradient s-proximal au carré en utilisant FISTA. En comparant les deux lignes pointillées de la Figure 3.1, on observe que le phénomène d'accélération existe bel et bien pour la norme du sous-gradient s-proximal au carré de FISTA par rapport à ISTA [39].

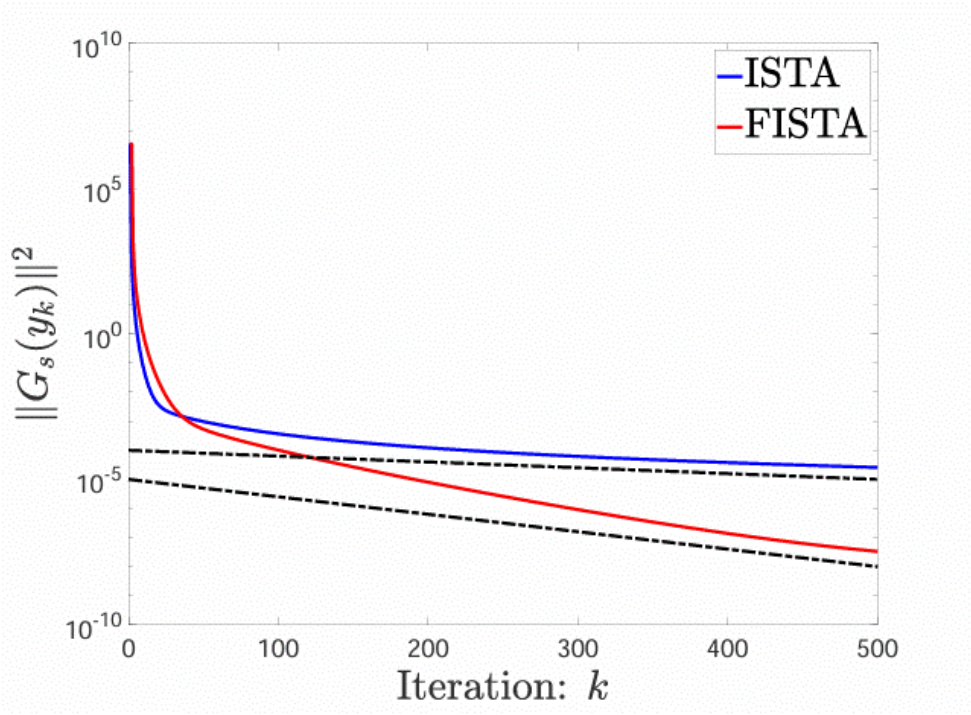

FIGURE 3.1 – Comparaison de la vitesse de convergence entre ISTA et FISTA

3.2.6 Domaines d'application de FISTA

L'algorithme FISTA s'applique aux mêmes types de problèmes que ISTA, notamment dans les domaines du traitement d'image[42, 20, 35, 6] , du traitement du signal [22, 55, 6] et de la régression LASSO[61, 26, 19]. Sa structure algorithmique repose sur les mêmes hypothèses que l'algorithme proximal de base, mais il introduit une accélération qui le rend particulièrement efficace dans les contextes de grande dimension ou lorsque la rapidité de convergence est cruciale. FISTA est ainsi souvent préféré à ISTA lorsqu'une meilleure efficacité numérique est recherchée.

3.2.7 Limites de l'algorithme FISTA

Bien que l'algorithme FISTA offre une amélioration significative en termes de vitesse de convergence par rapport à ISTA, avec un taux de convergence théorique de $O(1/k^2)$ [5], il présente également certaines limites.

Tout d'abord, la stabilité de FISTA peut être plus fragile que celle d'ISTA. En raison de l'extrapolation inertielle inspirée des méthodes de Nesterov [47], les itérés peuvent osciller, surtout lorsque les conditions de convexité stricte ne sont pas pleinement satisfaites ou que les paramètres sont mal calibrés [15]. Cette instabilité peut ralentir la convergence effective ou compliquer l'analyse de l'algorithme dans les cas pratiques.

Ensuite, FISTA nécessite une gestion plus fine des paramètres. En plus de la constante de Lipschitz L, il faut gérer correctement la suite (t_k), ce qui alourdit légè-

rement la complexité de mise en œuvre. Dans certains contextes, cette dépendance aux choix de paramètres peut compromettre les performances, en particulier si les fonctions ne sont pas bien conditionnées [2].

Enfin, bien que FISTA converge plus rapidement en théorie, il ne garantit pas toujours une meilleure efficacité numérique pour un petit nombre d'itérations. En effet, les premières oscillations dues à l'inertie peuvent annuler les gains de convergence dans les premières phases de l'algorithme [15].

3.3 Conclusion

Ce chapitre est consacré à l'analyse approfondie de l'algorithme FISTA (*Fast Iterative Shrinkage-Thresholding Algorithm*), une variante optimisée de l'algorithme ISTA, conçue pour accélérer la convergence dans le cadre des problèmes d'optimisation convexe non différentiable, en particulier ceux régularisés par la norme ℓ_1.

Nous avons constaté que FISTA s'appuie sur une combinaison ingénieuse des deux itérations les plus récentes en utilisant une mise à jour de type Nesterov, ce qui offre un taux de convergence optimal de $O(1/k^2)$ pour les fonctions convexes (Beck et Teboulle, 2009), contrairement à ISTA qui présente un taux de convergence de $O(1/k)$. De plus, bien que l'algorithme FISTA introduise une étape supplémentaire de calcul (mise à jour de la séquence s et combinaison linéaire des points), la complexité par itération reste comparable à celle d'ISTA.

Les performances numériques et les études théoriques attestent que FISTA procure une amélioration notable tout en maintenant la facilité de mise en œuvre d'ISTA. Cette approche est spécifiquement conçue pour les problèmes de haute dimension où la parcimonie est privilégiée, tels que le traitement d'image, l'IRM compressée ou la sélection de variables dans le cadre de la régression.

Ce chapitre offre donc un fondement robuste pour l'application de FISTA dans des contextes pratiques, tout en préparant le terrain pour d'autres variantes perfectionnées telles que FISTA adaptatif, FISTA avec redémarrage, ou l'étude dans un cadre non convexe.

Au chapitre suivant, nous mettrons en œuvre les algorithmes ISTA et FISTA pour aborder un problème de régression Lasso avec régularisation ℓ_1. Nous examinerons leurs performances numériques en les jugeant sur divers aspects tels que la vitesse de convergence, la précision et la stabilité, en utilisant des données simulées ainsi que des données concrètes provenant de secteurs appliqués. Ces essais permettront d'évaluer de manière concrète l'efficacité de FISTA dans la résolution de problèmes d'optimisation parcimonieuse.

CHAPITRE 4

Application numérique

4.1 Introduction

Ce chapitre traite de l'application pratique des algorithmes ISTA (Iterative Shrinkage Thresholding Algorithm) et FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) dans le contexte d'un problème de régression linéaire régularisée de type Lasso. La comparaison des performances en termes de précision de prédiction et de rapidité de convergence, lorsque ces deux méthodes sont mises en œuvre sur des données réelles provenant du secteur industriel, est l'objectif primordial.

Les informations sur les volumes de production annuelle de divers types de plaques en carton ondulé pour la période 2020-2024 proviennent de la société Général Emballage. L'année 2024 sera considérée comme la variable cible, tandis que les années antérieures serviront de variables explicatives. Ce genre de problème est clairement lié à la prévision de séries temporelles multivariées, tout en étant soumis à une restriction de parcimonie imposée par la régularisation ℓ_1 du Lasso.

Ainsi, nous appliquerons les algorithmes ISTA et FISTA sur les données normalisées, en analysant les coefficients obtenus, l'évolution des fonctions de coût, et les valeurs prédites. Cette étude permettra d'évaluer la capacité de ces méthodes à identifier les années les plus influentes sur la production future, tout en réduisant le surapprentissage par une sélection automatique des variables. Nous terminerons par une comparaison graphique et numérique des résultats obtenus, accompagnée d'une discussion critique sur leurs implications.

4.2 Introduction au procédé de fabrication des plaques en carton

La production de plaques de carton est une activité clé pour l'entreprise Général Emballage, qui se spécialise dans la production d'emballages en carton ondulé à destination de divers secteurs industriels. Ce procédé englobe diverses phases techniques, depuis la conversion des bobines de papier jusqu'à la production de plaques

prêtes à l'utilisation.

Les plaques de carton sont des supports rigides obtenus par l'assemblage de plusieurs couches de papier (liner et médium) formant des structures à simple, double ou triple cannelure. Elles constituent la base pour la production de caisses, boîtes et diverses sortes d'emballages.

Le processus de fabrication des plaques de carton se déroule en plusieurs phases cruciales : le papier médium est d'abord ondulé, puis collé sur des feuilles planes (appelées liners), l'assemblage se fait en fonction du type de carton (simple, double, triple cannelure), il est Salle découpé aux dimensions désirées, et enfin stocké ou envoyé à la clientèle.

Ce produit est l'un des éléments essentiels pour évaluer la performance industrielle. Sa production est rigoureusement contrôlée et organisée en fonction de la demande, des capacités des équipements et des restrictions concernant les matières premières.

4.3 Présentation de la base de données

La base de données étudiée contient les quantités annuelles de production de plaques de carton, mesurées en kilogrammes (Kg) sur une période de cinq ans (2020–2024). Chaque ligne correspond à un type de produit transformé ou à une étape de production issue d'une ligne de transformation spécifique. On y retrouve 10 produits différents. Les variables explicatives (X) sont les productions des années 2020 à 2023 (soit 4 variables X par produit), tandis que la variable cible (y) correspond à la production de l'année 2024. Ainsi, la base comprend 10 observations, chacune décrite par 4 variables explicatives et 1 variable cible, ce qui permet d'appliquer des méthodes de régression pour la prédiction.

Ces données permettent de modéliser la production industrielle d'un produit cible à partir des autres produits ou étapes, à l'aide d'outils statistiques de régression adaptés à des situations de grande dimension et de sélection de variables, comme le Lasso résolu par les algorithmes ISTA et FISTA.

Description des variables

Les variables de la base de données sont les suivantes :

- **FOS** : Feuilles ondulées simples – production brute en sortie d'ondulation,
- **OND** :Production de modules d'ondulation (papier ondulé brut) ;
- **618** :Produit semi-fini, probablement combiné à des liners ;
- **GTMZ** :Transformation GTMZ : découpe ou collage avancé ;
- **MCUT** :Machine de découpe CUT : découpage des plaques ;

- **MFC** :Module de façonnage et contrecollage ;
- **MRT** :Marquage, rainurage ou traitement thermique ;
- **PTMZ** :Post-traitement machine Z : étapes finales avant emballage ;
- **TMZ** :Transformation machine Z : produit spécifique ou personnalisé ;
- **VISION FOLD** : Plieuse-colleuse automatique – produit fini prêt à être expédié.

Dans la présente étude, nous nous intéressons à l'application de deux algorithmes d'optimisation, à savoir ISTA (Iterative Shrinkage-Thresholding Algorithm) et FISTA (Fast Iterative Shrinkage-Thresholding Algorithm), dans le cadre de la résolution d'un problème de régression Lasso. Le but est de créer un modèle et de prévoir la production annuelle de plaques de carton pour l'année 2024 en se basant sur les données historiques de production collectées entre 2020 et 2023. Cette méthode facilite la sélection automatique des variables explicatives les plus appropriées, tout en instaurant une régularisation qui favorise la simplicité du modèle.

L'étude vise à :

- Évaluer la capacité prédictive de chaque algorithme ;
- Comparer leur vitesse de convergence et la stabilité de leurs coefficients estimés ;
- Vérifier leur efficacité dans le contexte industriel de la société Général Emballage, où les données sont limitées et potentiellement corrélées.

Tableau des données

Produit	2020	2021	2022	2023	2024
FOS	45 508 200	48 527 528	40 901 512	37 710 658	38 233 466
OND	34 675 742	30 273 858	34 407 444	31 280 103	30 030 260
618	10 915 338	13 338 774	10 894 945	11 627 892	10 282 395
GTMZ	1 621 278	2 034 498	1 447 011	1 258 636	1 326 583
MCUT	7 937 425	9 382 533	8 820 320	7 841 035	6 922 261
MFC	7 580 546	8 555 609	9 134 240	7 090 580	6 519 715
MRT	10 708 480	11 965 367	9 851 100	9 391 780	9 356 551
PTMZ	1 714 839	1 666 205	940 119	867 658	1 051 282
TMZ	1 182 416	1 255 652	901 468	375 794	285 944
VISION FOLD	2 016 691	1 513 668	766 909	807 444	990 123

TABLE 4.1 – Production annuelle de plaques de carton (en kg) par type de produit (2020–2024)

Afin de mieux visualiser la dynamique de production au sein de l'entreprise sur la période étudiée, nous présentons ci-dessous 4.1, l'évolution annuelle des quantités produites pour chaque type de produit.

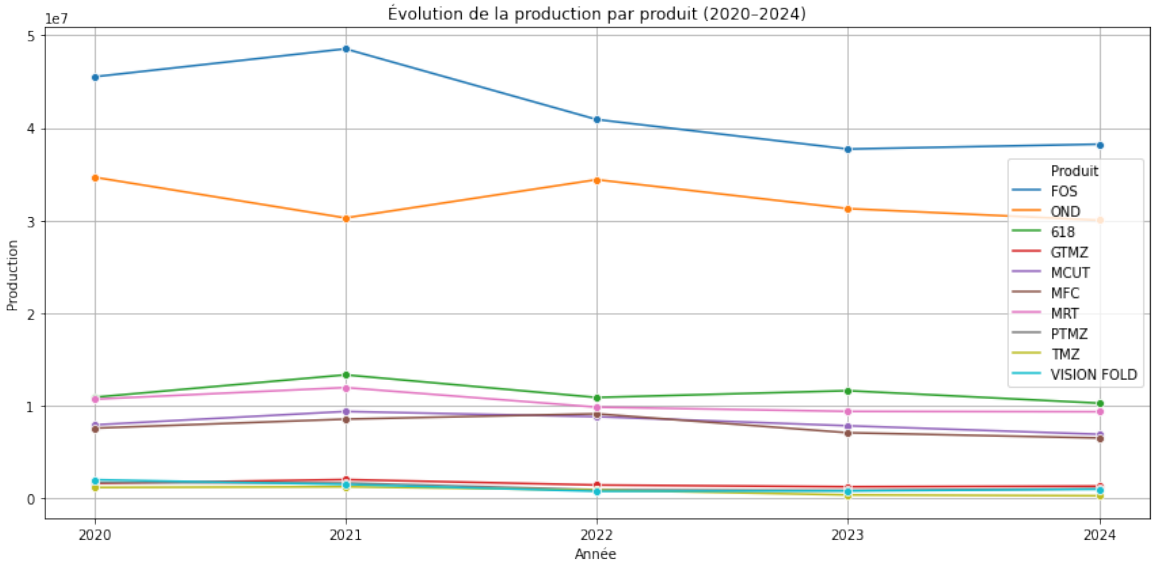

FIGURE 4.1 – Evolution de la production par produit (2020–2024)

Le graphique (4.1) de l'évolution de la production entre 2020 et 2024 montre que les produits FOS et OND sont les plus produits, bien que leur volume tende à diminuer au fil des années. Le produit 618 affiche une production importante avec un pic en 2021, suivi d'une légère baisse. Les produits comme MCUT, MFC et MRT restent relativement stables, tandis que d'autres, comme PTMZ, TMZ, GTMZ ou VISION FOLD, présentent une production plus faible et irrégulière, avec une tendance globalement décroissante.

4.3.1 Modèle de prédiction : Régression Lasso

Dans le cadre de la prédiction de la production de cartons, le problème est formulé comme une régression pénalisée. L'objectif est de construire un modèle linéaire permettant de prédire la production future (année 2024) à partir des productions passées (années 2020 à 2023), tout en sélectionnant les variables les plus pertinentes et en évitant le sur-apprentissage.

Le modèle utilisé repose sur l'optimisation de la fonction objectif du Lasso (Least Absolute Shrinkage and Selection Operator), qui s'écrit :

$$\mathcal{L}(\beta) = \frac{1}{2n}||X\beta - y||^2 + \lambda||\beta||_1$$

En pratique, cette fonction a été implémentée en Python comme suit :

```
 def compute_cost(X, y, beta, lambd)
n = len(y)
residual = X @ beta - y
return (1 / (2 * n)) * np.sum(residual**2) + lambd *
np.sum(np.abs(beta))
```

Cette fonction permet d'évaluer le compromis entre la qualité d'ajustement du modèle et la pénalisation des coefficients, assurant ainsi une meilleure généralisation.

4.3.2 Pourquoi utiliser les algorithmes ISTA et FISTA ?

Nous avons choisi d'utiliser les algorithmes ISTA et FISTA pour résoudre le problème de régression Lasso, en raison de leur capacité à effectuer une sélection automatique des variables tout en garantissant une prédiction performante. Plus précisément, FISTA se distingue par sa rapidité de convergence, rendue possible grâce à un mécanisme d'accélération basé sur l'inertie.

Ces deux méthodes itératives visent à minimiser une fonction objectif composée d'un terme de moindres carrés et d'un terme de régularisation de norme ℓ_1. Les fondements théoriques, ainsi que les définitions précises de ces algorithmes, ont été détaillés dans les chapitres précédents.

Choix et justification des paramètres d'implémentation

Deux fonctions clés sont implémentées :

Le premier bloc de code met en œuvre l'opérateur de seuillage doux, élément clé de la régularisation ℓ_1.

```
soft_thresholding(x, lambd)
```

Le second bloc de code calcule la fonction coût du Lasso :

```
compute_cost(X, y, beta, lambd)
```

Choix de λ

Pour déterminer la meilleure valeur du paramètre de régularisation dans un modèle de régression Lasso, nous utilisons la classe `LassoCV` de la bibliothèque `scikit-learn`. Celle-ci effectue automatiquement une validation croisée et identifie la valeur de `lambda` minimisant l'erreur quadratique moyenne.

Voici le code ci-dessous pour trouver la valeur de λ.

```
lasso_cv = LassoCV(cv=5, random_state=0).fit(X_scaled, y)
```

La validation croisée a permis de déterminer la valeur optimale du paramètre de régularisation, $\lambda = 0.001$, que nous retiendrons pour l'entraînement final du modèle ainsi que pour l'application des algorithmes ISTA et FISTA

```
lambda_lasso = 0.001
```

Taux d'apprentissage (learning rate)

Le taux d'apprentissage, noté *lr*, contrôle la vitesse à laquelle l'algorithme met à jour les coefficients à chaque itération. Pour garantir la convergence de ces algorithmes, le taux doit être choisi de façon à vérifier :

$$lr = \frac{1}{L}, \quad \text{avec} \quad L = \|X^T X\|$$

où L représente la constante de Lipschitz du gradient de la fonction à minimiser.

Nous avons donc utilisé les commandes suivantes pour estimer cette constante et calculer le taux d'apprentissage associé :

```
L = np.linalg.norm(X_scaled.T @ X_scaled, ord=2)
lr = 1 / L
```

Les résultats obtenus sont :

```
L = 39.751060
```

```
lr = 0.025157
```

Tolérance de convergence (`tol`)

Le paramètre de tolérance (`tol`) définit un critère d'arrêt basé sur la variation des coefficients estimés entre deux itérations successives. Lorsque cette variation devient inférieure à un certain seuil, l'algorithme considère que la solution est suffisamment stable et interrompt les calculs.

Dans notre implémentation, ce seuil a été fixé à :

$$\text{tol} = 10^{-6}$$

Ce choix garantit une bonne précision des résultats tout en évitant un nombre excessif d'itérations, ce qui permet de réduire le temps de calcul.

```
beta_ista = ista(X_scaled, y_scaled, lambd=0.001,
      lr=0.025157, max_iter=100, tol=1e-6)
```

Nombre maximum d'itérations (`max_iter`)

Le paramètre `max_iter` fixe un nombre maximal d'itérations pour l'algorithme. Cela permet de limiter le temps de calcul et de garantir l'arrêt du processus, même si la convergence complète n'est pas atteinte.

Dans le cadre de notre étude, deux valeurs ont été utilisées selon les objectifs :

– `max_iter = 1000` : utilisée pour les expérimentations principales afin de garantir une convergence complète ;

```
beta_ista = ista(X_scaled, y_scaled, lambd=0.001,
    lr=0.025157, max_iter=1000, tol=1e-6)
```

– `max_iter = 100` : utilisée pour les tests de convergence afin de visualiser plus clairement la progression des algorithmes.

```
beta_ista_short = ista(X_scaled,
    y_scaled, lambd=0.001,
lr=0.025157,max_iter=100 , tol=1e-6)
```

4.3.3 Estimation des coefficients

Mise en œuvre des algorithmes ISTA et FISTA

L'implémentation se base sur les fonctions ISTA et FISTA codées en Python. Après l'exécution, les vecteurs de coefficients estimés sont récupérés et stockés dans un tableau pour une analyse comparative.

```
beta_ista, cost_ista = ista(X_scaled, y_scaled,
    lambd=0.001, lr=0.025157, max_iter=100)
```

```
beta_fista, cost_fista = fista(X_scaled,
y_scaled, lambd=0.001, lr=0.025157, max_iter=100)
```

Présentation tabulaire des coefficients estimés

Le tableau ci-dessous synthétise les valeurs numériques des coefficients obtenus :

Année	ISTA	FISTA
2020	0.301554	0.492650
2021	0.137027	0.039741
2022	0.249020	0.000000
2023	0.304412	0.458835

TABLE 4.2 – Tableau des coefficients estimés

Visualisation graphique des coefficients estimés

Afin d'illustrer les différences entre les deux méthodes, un graphique est généré à l'aide de `matplotlib`. Il représente les coefficients estimés pour chaque année (de 2020 à 2023) par ISTA et FISTA.

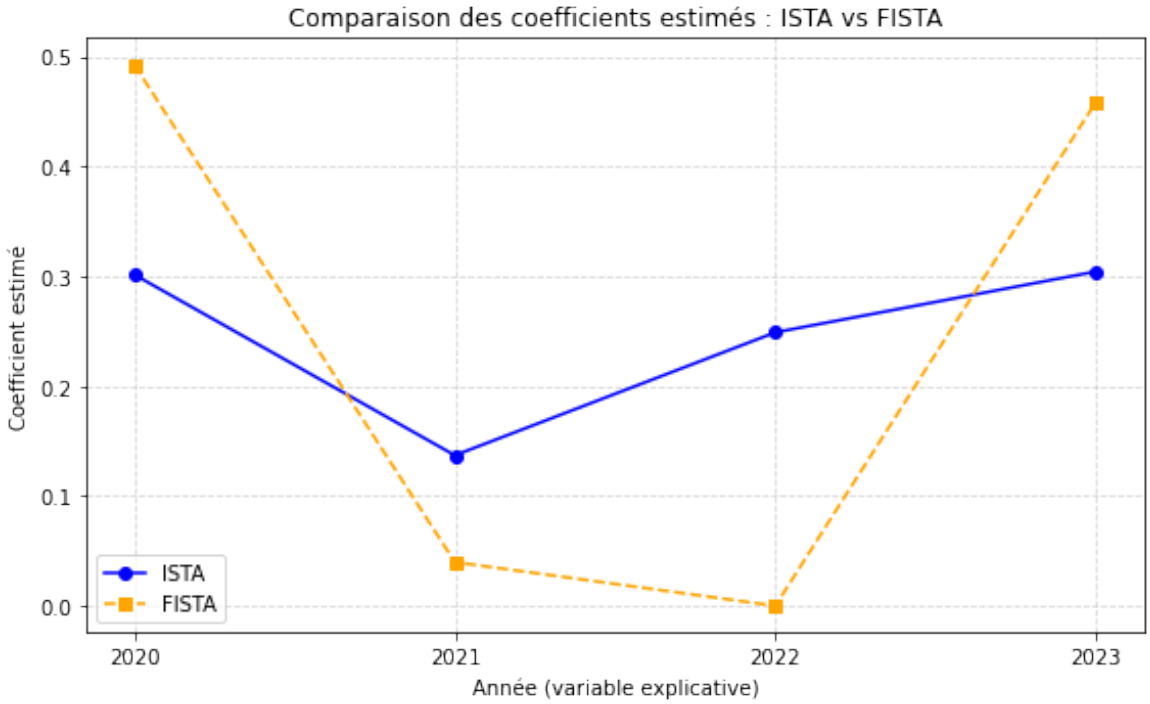

FIGURE 4.2 – Comparaison des coefficients estimés entre ISTA et FISTA

Cette représentation graphique démontre que FISTA a mis à zéro le coefficient lié à l'année 2022, à la différence d'ISTA qui retient des valeurs non nulles pour toutes les années. Ceci illustre la compétence de FISTA à réaliser une sélection automatique de variables.

Interprétation des résultats

Le graphique compare les coefficients estimés par les algorithmes ISTA et FISTA pour prédire la production de 2024 à partir des données des années précédentes. Ces coefficients reflètent l'importance relative de chaque année dans le modèle de prédiction.

ISTA attribue des coefficients non nuls à toutes les années, ce qui indique que chaque année (de 2020 à 2023) contribue à la prédiction. Parmi elles, 2020 (0,301554) et 2023 (0,304412) sont les plus influentes, tandis que 2021 (0,137027) et 2022 (0,249020) ont un impact plus modéré. Cette répartition montre qu'ISTA intègre l'ensemble des variables, même celles ayant une influence réduite.

FISTA, en revanche, adopte une approche plus sélective. Il attribue des coefficients élevés à 2020 (0,492650) et 2023 (0,458835), mais élimine totalement l'influence

de 2022 (coefficient nul), et réduit fortement celle de 2021 (0,039741). Cela signifie que l'algorithme considère que seules certaines années sont réellement informatives pour la prédiction.

Sélection des variables pertinentes

Présentation des coefficients sélectionnés

Les tableaux suivants mettent en parallèle les valeurs non nulles des coefficients estimés par la méthode ISTA et FISTA. Les années dont les coefficients sont nuls sont signalées par un tiret :

	2020	**2021**	**2022**	**2023**
ISTA	0.301554	0.137027	0.249020	0.304412
FISTA	0.492650	0.039741	—	0.458835

TABLE 4.3 – Coefficients sélectionnés (non nuls) par ISTA et FISTA

Interprétation des coefficients sélectionnés

Le tableau indique les coefficients non nuls obtenus grâce à ISTA et FISTA. On observe que ISTA retient toutes les années (2020 à 2023), tandis que FISTA supprime intégralement l'année 2022, suggérant qu'elle ne fournit pas d'information suffisamment discriminante. Pour les deux scénarios, les années 2020 et 2023 se démarquent par leur coefficient élevé, mettant en évidence leur rôle prépondérant dans l'anticipation de la production pour 2024. En intégrant une régularisation renforcée, FISTA réalise une sélection automatique des variables et génère un modèle plus parcimonieux.

4.3.4 Comparaison des performances selon les paramètres d'implémentation

Les deux tableaux suivants présentent les erreurs quadratiques moyennes (MSE) obtenues à l'aide des algorithmes ISTA et FISTA, en fixant le paramètre de régularisation à $\lambda = 0{,}001$, et en faisant varier le taux d'apprentissage `lr`. Les calculs ont été effectués pour deux configurations différentes du nombre maximal d'itérations, à savoir 1000 et 1500, afin de comparer l'impact de ce paramètre sur la performance et la convergence des deux algorithmes. Cette analyse permet d'évaluer à la fois l'influence du taux d'apprentissage et celle du nombre d'itérations sur la précision des estimations.

Taux d'apprentissage (lr)	MSE ISTA	MSE FISTA
0.005	0.000520	0.000129
0.010	0.000434	0.000133
0.020	0.000349	0.000134
0.030	0.000292	0.000134

TABLE 4.4 – MSE pour ISTA et FISTA selon différentes valeurs de `lr`, avec `max_iter`=1000.

Taux d'apprentissage (lr)	MSE ISTA	MSE FISTA
0.005	0.000468	0.000131
0.010	0.000386	0.000134
0.020	0.000292	0.000134
0.030	0.000233	0.000134

TABLE 4.5 – MSE pour ISTA et FISTA selon différentes valeurs de `lr`, avec `max_iter`=1500.

Interprétation des résultats

Les résultats présentés dans les deux tableaux révèlent des tendances contrastées entre les algorithmes ISTA et FISTA face à la variation du taux d'apprentissage (`lr`) pour deux valeurs du nombre maximal d'itérations (1000 et 1500). Pour ISTA, on observe une diminution régulière de l'erreur quadratique moyenne (MSE) lorsque le taux d'apprentissage augmente, aussi bien à 1000 qu'à 1500 itérations. Cette évolution témoigne d'une amélioration progressive de la convergence, la MSE passant de 0.000520 à 0.000292 puis à 0.000233 pour `lr` = 0.030. Cela suggère qu'un taux plus élevé, combiné à un nombre d'itérations plus important, permet à ISTA d'atteindre une solution plus précise. Néanmoins, cette tendance pourrait ne pas se maintenir au-delà de cette plage, car un `lr` trop grand pourrait mener à une instabilité du processus d'optimisation.

En revanche, FISTA se distingue par une grande stabilité des erreurs MSE, indépendamment du taux d'apprentissage utilisé. À la fois pour 1000 et 1500 itérations, les valeurs de MSE restent très proches (comprises entre 0.000129 et 0.000134), et n'affichent pas de tendance claire selon `lr`. La MSE la plus basse est atteinte pour `lr` = 0.005 à 1000 itérations, mais les variations restent négligeables. Cela confirme que FISTA, grâce à sa stratégie d'accélération, est à la fois plus rapide et moins sensible aux variations du taux d'apprentissage, ce qui en fait un algorithme plus robuste dans ce type de configuration.

En somme, ISTA bénéficie davantage d'un ajustement fin du taux d'apprentissage, tandis que FISTA offre une performance stable et fiable même avec un choix de `lr` peu optimisé.

Remarque. Ces résultats illustrent bien la sensibilité d'ISTA au choix du taux d'apprentissage, contrairement à FISTA qui montre une meilleure stabilité. Cela confirme que FISTA, grâce à son accélération, est non seulement plus rapide mais aussi plus robuste face aux variations de paramètres d'apprentissage dans cette configuration.

4.3.5 Analyse de la convergence de ISTA et FISTA

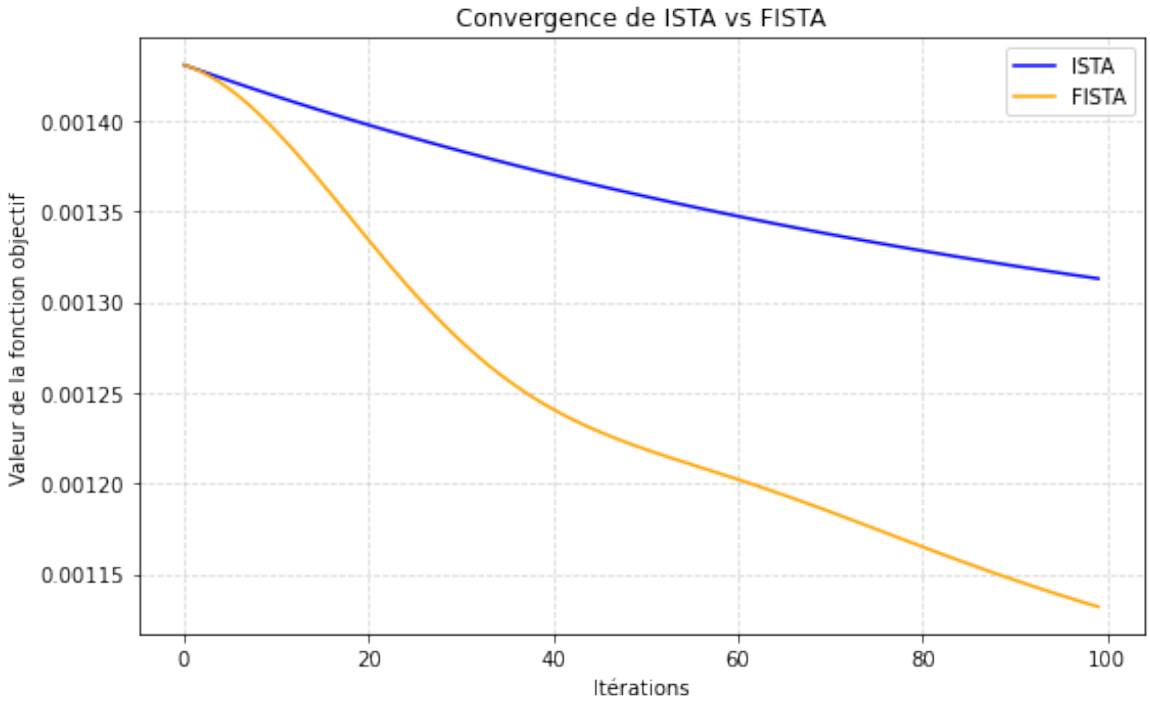

FIGURE 4.3 – Comparaison de la convergence entre ISTA et FISTA

Le diagramme ci-dessus 4.3 représente le changement de la fonction objectif à travers les itérations pour les algorithmes ISTA et FISTA. On note que la courbe de FISTA diminue beaucoup plus vite que celle d'ISTA, ce qui indique une convergence plus rapide vers la solution optimale.

Interprétation

Cette variation de rapidité s'explique par la nature même de FISTA, qui intègre un élément d'accélération issu des techniques de Nesterov. Ce procédé donne à FISTA la possibilité de prévoir le chemin de descente, et ainsi d'abaisser plus rapidement la valeur de la fonction objectif, contrairement à ISTA qui suit une descente plus lente et constante.

FISTA se distingue par sa rapidité de convergence, le rendant ainsi un choix de prédilection pour les problèmes où la durée de calcul est cruciale.

4.3.6 Prédiction de la production de 2024

Le tableau suivant 4.6 présente, pour chaque produit, la valeur réelle de la production en 2024 ainsi que les prédictions générées par ISTA et FISTA :

Produit	Réel 2024	Prévision ISTA	Prévision FISTA
FOS	38 233 466	38 495 874	38 285 401
OND	30 030 260	29 558 535	29 939 558
618	10 282 395	10 249 463	10 189 944
GTMZ	1 326 583	1 095 358	1 109 693
MCUT	6 922 261	7 360 238	7 226 574
MFC	6 519 715	7 006 794	6 819 297
MRT	9 356 551	9 107 290	9 071 631
PTMZ	1 051 282	825 282	901 680
TMZ	285 944	490 053	508 449
VISION FOLD	990 123	809 687	946 347

TABLE 4.6 – Comparaison des prévisions ISTA et FISTA pour l'année 2024

Ce graphique 4.4 permettra de visualiser produit par produit les performances des deux algorithmes comparées aux données réelles.

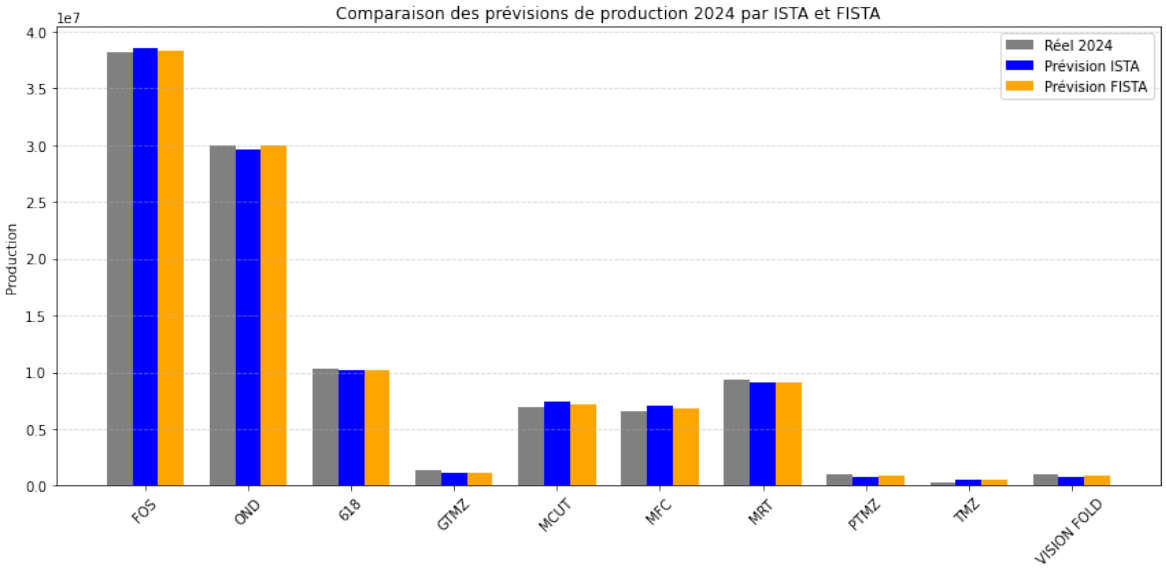

FIGURE 4.4 – Comparaison des prévisions de production 2024 par ISTA et FISTA

interprétation

Pour les produits à fort volume, tels que FOS, OND, 618 et MRT, les deux algorithmes donnent des résultats raisonnablement proches des valeurs observées. Cependant, FISTA se démarque légèrement par une meilleure précision. Par exemple, pour FOS, la prédiction FISTA est très proche de la production réelle, alors qu'ISTA présente une légère surestimation. De même, pour OND, 618 et MRT, FISTA offre des prévisions plus cohérentes avec les données réelles, réduisant ainsi l'écart de manière significative.

Concernant les produits MCUT et MFC, bien qu'ISTA fournisse des prévisions acceptables, celles de FISTA sont systématiquement plus proches de la réalité, confirmant une meilleure adaptation du modèle. Cela est particulièrement visible pour MFC, où l'écart entre la valeur réelle et la prédiction est nettement réduit avec FISTA.

Pour les produits à faible volume, comme GTMZ, PTMZ, TMZ et VISION FOLD, les deux algorithmes présentent une tendance à la surestimation, plus marquée avec ISTA. FISTA, bien que pas parfait, s'ajuste mieux aux faibles quantités et offre des prévisions plus stables et proches des valeurs mesurées, notamment pour VISION FOLD et PTMZ, où il améliore considérablement la précision par rapport à ISTA.

Dans l'ensemble, l'analyse met en évidence la supériorité de l'algorithme FISTA sur ISTA, tant pour les produits à forte production que pour ceux à faibles volumes. Cette performance peut être attribuée à sa meilleure vitesse de convergence et à sa capacité à affiner les coefficients de manière plus efficace, ce qui en fait un outil particulièrement fiable pour la prédiction de production industrielle.

4.4 Discussion des résultats et conclusion

L'analyse comparative des algorithmes ISTA et FISTA, utilisée dans un cadre industriel de prédiction de la production sur un problème du type Lasso, a permis de confirmer leur performance. Les deux méthodes offrent des performances satisfaisantes, avec des prédictions globalement proches des valeurs réelles. Toutefois, FISTA se démarque par sa rapidité de convergence, sa capacité à mieux sélectionner les variables pertinentes, et sa précision accrue sur les produits à forte production. Ces qualités en font un outil particulièrement adapté aux problématiques où la parcimonie et la stabilité du modèle sont essentielles.

Conclusion générale

Ce mémoire a été conçu dans le contexte de l'étude des méthodes de régularisation en régression, en particulier le modèle Lasso, qui est actuellement d'une grande importance dans l'analyse des données modernes. Le Lasso se fait remarquer par sa faculté à conjuguer l'estimation et le choix des variables, le rendant ainsi un instrument performant pour modéliser des données de haute dimension, en générant des modèles simples, compréhensibles et significatifs.

Afin de traiter le problème d'optimisation proposé par le Lasso, nous avons examiné deux approches basées sur l'optimisation proximale : les algorithmes ISTA et FISTA. La première propose une méthode simple basée sur une succession de techniques de descente de gradient et de seuillage doux. Le second, qui repose sur une accélération spécifique, offre un net progrès en termes de performance, notamment en matière de vitesse et d'efficacité numérique. Ces méthodes s'appuient sur des outils mathématiques robustes tels que les sous-gradients, les opérateurs proximaux et les propriétés de convexité.

L'implémentation pratique sur des données du secteur a révélé les avantages concrets de ces approches. FISTA a démontré une efficacité notable en matière de prédiction, tout en préservant les avantages du Lasso pour la sélection automatique des variables.

Ainsi, cette recherche a enrichi notre compréhension des techniques de régularisation et de l'optimisation convexe, aussi bien du point de vue théorique que pratique. Il propose également des possibilités d'examiner des variantes plus avancées, comme les versions adaptatives ou stochastiques, ainsi que leur application sur d'autres genres de données et modèles. De manière générale, cela témoigne de l'importance croissante des méthodes d'optimisation dans la conception d'outils statistiques modernes, utiles dans de nombreux secteurs d'analyse et d'assistance à la prise de décision.

Pour la suite, il serait intéressant d'appliquer ces méthodes à d'autres domaines comme la finance ou la santé, de tester des régularisations alternatives comme l'Elastic Net, et de les comparer à des approches plus récentes, notamment le deep learning et les modèles non linéaires.

Bibliographie

[1] Anbari, M. E. and Mkhadri, A. (2009). Penalized regression with a combination of the ℓ_1 norm and the correlation-based penalty. In *Actes des 41èmes Journées de Statistique de la SFdS*, Bordeaux, France. HAL Open Science.

[2] Attouch, H., Chbani, Z., Peypouquet, J., and Redont, P. (2018). Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Mathematical Programming*, 168 :123–175.

[3] Bachene, A. (2021). Semi-continuité inférieure et convexité en calcul des variations. Diplome de master en Mathématiques , Université Dr. Yahia Farès de Médéa.

[4] Bayram, I. (2015). On the convergence of the iterative shrinkage/thresholding algorithm with a weakly convex penalty. *arXiv preprint arXiv :1510.07821v2*.

[5] Beck, A. and Teboulle, M. (2009a). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1) :183–202.

[6] Beck, A. and Teboulle, M. (2009b). A fast iterative shrinkage thresholding algorithm with application to wavelet-based image deblurring. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '09)*, pages 693–696, United States. IEEE Computer Society.

[7] Bernard, N. and Feti, A. (2018). Le lasso (least absolute shrinkage and selection operator). *Université de Lille 1*.

[8] Bonnefoy, A. (2015). Élimination dynamique : accélération des algorithmes d'optimisation convexe pour les régressions parcimonieuses. *Aix-Marseille Université, Laboratoire d'Informatique Fondamentale de Marseille (LIF)*. Rapport de recherche ou thèse.

[9] Bortoli, V. D. and Durmus, A. (2025). *TP d'optimisation numérique : Minimisation par optimisation proximale*. Master Hadamard, première année. Supports de cours et travaux pratiques.

[10] Boyd, S. and Vandenberghe, L. (2004). Convex optimization. *Cambridge University Press*.

[11] Bradley Efron, Trevor Hastie, I. J. and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2) :407–499.

[12] Brahmi, A. (2024). Introduction au lasso et aux méthodes de régularisation. Mémoire de Master, Université Abderrahmane Mira - Béjaia.

[13] Carpentier P., Chiche A., e. F. P. (2020). Optimisation non-différentiable et introduction aux méthodes proximales. *ENSTA Paris*.

[14] Cauchy, A. (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus de l'Académie des Sciences de Paris*, 25 :536–538.

[15] Chambolle, A. and Dossal, C. (2015a). Convergence des itérés de fista. *prépublication HAL*. Prépublication HAL.

[16] Chambolle, A. and Dossal, C. (2015b). On the convergence of the iterates of fista. *Journal of Optimization Theory and Applications*, 166(3) :968–982.

[17] Chambolle, A. and Dossal, C. H. (2014). On the weak convergence of the iterates of" fista".

[18] Chambolle, A. and Pock, T. (2010). A first-order primal-dual algorithm for convex problems with applications to imaging. *SIAM Journal on Imaging Sciences*, 3(4) :1302–1324.

[19] Chen, X., Liu, J., Wang, Z., and Gu, J. (2018). Theoretical linear convergence of unfolded ista and its practical weights and thresholds. *arXiv preprint arXiv :1806.09228*.

[20] Choudhary, H., Sahoo, K., Orra, A., Kumar, S., Sharma, H., Balachandran, K., Kim, J., and Bansal, J. (2023). Modified iterative shrinkage–thresholding algorithm for image de-blurring in medical imaging. *Springer Nature Singapore*.

[21] Combettes, P. L. and Wajs, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4) :1168–1200.

[22] Daisuke Ito, S. T. and Wadayama, T. (2019). Trainable ista for sparse signal recovery. *IEEE Transactions on Signal Processing*, 67(12) :3099–3113.

[23] Daubechies, I., Defrise, M., and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11) :1413–1457.

[24] Dossal, C. (Janvier 2017). Optimisation convexe.

[25] Gareth, J., Witten, D., Hastie, T., and Tibshirani, R. (2023). *An Introduction to Statistical Learning : with Applications in R*. Springer, New York, NY, 2^{e} éd. edition.

[26] Gregor, K. and LeCun, Y. (2010). Learning fast approximations of sparse coding. *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*.

[27] Guechtouli, A. A. and Rahmoune, I. (2024). Sélection des variables de régression linéaire et régularisée. Mémoire de Master, Université USTHB.

[28] Guillot, D., Rajaratnam, B., Rolfs, B. T., Maleki, A., and Wong, I. (2012). Iterative thresholding algorithm for sparse inverse covariance estimation. *Department of Statistics, Stanford University*.

[29] Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York, NY, 2nd edition.

[30] Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity : The Lasso and Generalizations*. Chapman and Hall/CRC, Boca Raton, FL.

[31] Hoerl, A. E. and Kennard, R. W. (1968). On regression analysis and biased estimation. *Technometrics*, 10(4) :422–423.

[32] Hoerl, E. A. and Kennard, W. R. (1970). Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, 12(1) :55–67.

[33] Kerroum, K. (2015). Mémoire master, régression multiple et ridge.

[34] Kharoubi, R. (2016). Une nouvelle approche pour la sélection des variables dans le cas de modèles de discrimination en grandes dimensions. Mémoire de maîtrise, Université du Québec à Montréal.

[35] Kong, F. (2018). Comparison of reconstruction algorithms for compressive sensing magnetic resonance imaging. *Multimedia Tools and Applications*, 77 :22617–22628.

[36] Kumar, D. (May 2019). Ridge regression and lasso estimators for data analysis. *Missouri State University*.

[37] Kumaresh A K, Kother Mohideen S, B. I. (2015). Comparative study of restoration algorithms ista and iista. *International Journal of Scientific & Engineering Research*, 6(6).

[38] Lafond, M. H. (2017). *Analyse de données de grande dimension à l'aide de méthodes d'apprentissage statistique*. Université du Québec à Montréal.

[39] Li, B., Shi, B., and Yuan, Y. (2022a). Linear convergence of ista and fista. *Mathematical Programming*, 190(1) :121–152.

[40] Li, B., Shi, B., and Yuan, Y. (2022b). Proximal subgradient norm minimization of ista and fista. Technical report, University of Chinese Academy of Sciences, Beijing 100049, China. Technical report.

[41] Lin Yu Wang, Ming Qi He, J. H. X. P. F. Y. (2020). An iterative threshold algorithm based on log-sum norm regularization for magnetic resonance image recovery. *Progress In Electromagnetics Research M*, 88 :121–131.

[42] Lustig, M., Donoho, D. L., and Pauly, J. M. (2007). Sparse mri : The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine*, 58(6) :1182–1195.

[43] Marquardt, W. D. and Snee, R. D. (1975). Ridge regression in practice. *The American Statistician*, 29(1) :3–20.

[44] Martinet, B. (1970). Brève communication : Régularisation d'inéquations variationnelles par approximations successives. *Revue Française d'Informatique et Recherche Opérationnelle, Série Rouge*, 4(R3) :154–158.

[45] Montgomery, D. C., Peck, E. A., and Vining, G. G. (2012). Introduction to linear regression analysis. *Wiley Series in Probability and Statistics*, 5(4) :295–312.

[46] Nesterov, Y. et al. (2018). *Lectures on Convex Optimization*. Springer, Cham, Switzerland.

[47] Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate o(1/k²). *Doklady Akademii Nauk SSSR*, 269 :543–547.

[48] Ounaissi, D. (2016). *Méthodes Quasi-Monte Carlo et Monte Carlo : application au calcul des estimateurs LASSO et LASSO bayésien*. Thèse de doctorat, Université de Lille.

[49] O'Donoghue, B. and Candes, E. (November 27, 2024). Adaptive restart for accelerated gradient schemes.

[50] Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3) :127–239.

[51] Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5) :1–17.

[52] Rondepierre, A. (2017-2018). Méthodes numériques pour l'optimisation non linéaire déterministe. *Département Génie Mathématique et Modélisation*, pages 1168–1200.

[53] Royer, C. W. (2024). Outils d'optimisation pour les sciences des données et de la décision. Technical report, Institut de Statistique Mathématique, Université Paris-Dauphine. Rapport technique.

[54] S. Belkacemi, K. O. (2010). Mémoire de licence, problèmes de multicolinéarité dans les modèles de régression linéaire.

[55] Saluja, R. and Deb, S. (2016). Reconstruction du signal vocal à l'aide de l'algorithme de seuillage de rétrécissement itératif en deux étapes. *Revue Internationale des Applications Informatiques*, 153(11) :1–4.

[56] Shi, B., Du, S. S., Su, W., and Jordan, M. I. (2019). Acceleration via symplectic discretization of high-resolution differential equations. *Advances in Neural Information Processing Systems*, 32.

[57] Su, W., Boyd, S., and Candès, E. J. (2016). A differential equation for modeling nesterov's accelerated gradient method : Theory and insights. *Journal of Machine Learning Research*, 17 :1–43.

[58] Sulam, J., Aberdam, A., Beck, A., and Elad, M. (2019). On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(8) :1968–1980.

[59] Tao, S., Boley, D., and Zhang, S. (2015). Convergence of common proximal methods for ℓ_1-regularized least squares. *University of Minnesota, Minneapolis, MN, USA*.

[60] Tao, S., Boley, D., and Zhang, S. (2016). Local linear convergence of ista and fista on the lasso problem. *SIAM Journal on Optimization*, 26(1) :313–336.

[61] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society : Series B (Methodological)*, 58(1) :267–288.

[62] Xiang, J., Dong, Y., and Yang, Y. (2021). Fista-net : Learning a fast iterative shrinkage thresholding network for inverse problems in imaging. *IEEE Transactions on Medical Imaging*, 40(5) :1329–1339.

[63] Y. Dodge, V. R. (2004). *Analyse de régression appliquée (2ème éd.)*. DUNOD, Paris.

[64] Yuan, Y. and Chen, X. (2020). Interpretable learned lasso. *Journal of Machine Learning Research*.

[65] Y.Watts (2023). *Le Lasso Linéaire : une méthode pour des données de petites et grandes dimensions en régression linéaire*. Université de Montréal.

[66] Zhao, Y. and Huo, X. (2023). A survey of numerical algorithms that can solve the lasso problems. *Biostatistics and Research Decision Sciences Department, Merck & Co., Inc.*

ANNEXE A

Pour la mise en œuvre de l'application numérique liée à ce mémoire, nous avons fait appel à l'environnement Jupyter Notebook et au langage de programmation Python. Ce choix repose sur la flexibilité, l'interaction et la clarté que propose cette plateforme, particulièrement ajustée aux travaux de modélisation statistique et d'optimisation numérique.

Qu'est-ce que Jupyter Notebook

Jupyter Notebook est une plateforme open source interactive qui offre la possibilité d'intégrer du code Python, des commentaires explicatifs, des équations mathématiques (en LaTeX), et des visualisations au sein d'un unique document exécutable. On l'utilise fréquemment dans les domaines de la science des données, de l'apprentissage automatique, du traitement du signal et de la statistique appliquée. Son format en cellules autorise l'exécution du code par étapes, l'affichage immédiat des résultats et l'expérimentation flexible de différents paramètres. On a choisi cet outil pour sa facilité d'utilisation, la clarté de son organisation de code et sa compatibilité avec toutes les bibliothèques Python fréquemment employées dans le domaine des sciences des données, la modélisation numérique et l'analyse exploratoire.

Jupyter Notebook a été lancé en 2014 par Fernando Pérez et Brian Granger, dans le cadre du projet Jupyter, en héritage du projet IPython. Il est aujourd'hui largement utilisé dans la recherche, l'enseignement et l'analyse de données.

Résumé

Ce mémoire explore l'utilisation des méthodes de régularisation ℓ_1, notamment le Lasso, dans le cadre de la régression et de la sélection de variables. Deux algorithmes d'optimisation proximale, ISTA et FISTA sont étudiés, comparés et appliqués à des données industrielles réelles. Les résultats montrent l'efficacité de FISTA en termes de rapidité et de précision, confirmant l'intérêt des méthodes proximales accélérées pour le traitement de données complexes.

Mots-clés : Lasso, régularisation ℓ_1, sélection de variables, optimisation convexe, algorithme proximal, ISTA, FISTA, seuillage doux, régression pénalisée, données industrielles.

Abstract

This thesis explores the use of ℓ_1 regularization methods, particularly Lasso, in regression and variable selection. Two proximal optimization algorithms, ISTA and FISTA are studied, compared, and applied to real industrial data. The results demonstrate the effectiveness of FISTA in terms of speed and accuracy, confirming the value of accelerated proximal methods for processing complex data.

Keywords : Lasso, ℓ_1 regularization, variable selection, convex optimization, proximal algorithm, ISTA, FISTA, soft thresholding, penalized regression, industrial data.