REPUBLIQUE ALGERIENNE DEMOCRATIQUE et POPULAIRE.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.
UNIVERSITE MOULOUD MAMMERI de TIZI-OUZOU

Faculté des Sciences
Département de Mathématiques

Faculté des Sciences
Département de Mathématiques

Mémoire de Master
en
Mathématiques
Option
mathématique appliquée à la gestion

Thème

Méthodes d'optimisation non linéaire et applications

Présenté par

BENHAMNA DIHIA

Devant le jury

M MOHAND OUANES	Professeur	UMMTO	Rapporteur
M CHEBAH MOHAMMED	MCB	UMMTO	President
M GOUBI MOULOUD	MCB	UMMTO	Examinateur

Soutenu le 03/10/2019

Dédicaces

A mes parents :
Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études. Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.
Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.
Que dieu leur accorde santé et prospérité.
A mes frères et soeurs, je leur souhaite un avenir plein de joie, de bonheur, de réussite et sérénité. Je leur exprime à travers ce travail mes sentiments de fraternité et d'amour.
A tous mes amis, à qui je souhaite tout le succès dans leur vie.

Remerciements

Louange à dieu, le miséricordieux, sans lui rien de tout cela n'aurait pu être. Je tiens tout d'abord à témoigner toute ma reconnaissance à mon promoteur Monsieur MOHAND OUANES pour avoir accepté d'encadrer mon travail et pour tous ses précieux conseils. Je remercie également les membres du jury qui ont accepté d'évaluer mon travail.
J'adresse également tous mes remerciements à l'ensemble des enseignants qui m'ont suivi pendant mon cursus.
Enfin, merci à tous ceux qui ont contribué de près ou de loin à ce mémoire, du point de vue scientifique ou administratif

Table des matières

Introduction

L'optimisation et particulièrement la programmation mathématique, vise à résoudre des problèmes où l'on cherche à déterminer parmi un grand nombre de solutions candidates, celle qui donne le meilleur résultat de la fonction objectif. Plus précisément, on cherche à trouver une solution satisfaisant un ensemble de contraintes, et qui minimise ou maximise une fonction donnée, l'application de la programmation mathématique est en expansion croissante et trouve beaucoup d'applications dans plusieurs domaines pratiques. L'optimisation non linéaire s'occupe principalement des problèmes d'optimisation dont les données, i.e., les fonctions définissant ces problèmes, sont non linéaires, mais sont aussi différentiables autant de fois que nécessaire pour l'établissement des outils théoriques, comme les conditions d'optimalités, ou pour la bonne marche des algorithmes de résolution qui y sont introduits et analysés. L'objectif de ce mémoire est l'implémentation dans un langage de programmation, en l'occurence MATLAB, d'algorithmes d'optimisation non linéaires, ce qui nous permettra ensuite de les comparer, d'étudier le comportement en pratique des méthodes au-delà des considérations purement théoriques, et d'établir ainsi plus précisément, à la lumière de leur utilisation pratique, leurs champs d'application et de mettre en évidence leurs qualités et leurs défauts respectifs. Ce travail s'articule autour de trois chapitres : Le premier chapitre où nous abordons les conditions d'optimalité de l'optimisation sans contraintes ainsi que la description des méthodes de résolution de ce type de problèmes d'optimisation. Parmi lesquelles on peut citer :
Méthodes d'optimisation basées sur le gradient.
Méthode de Newton.
Méthodes utilisant des directions conjuguées.
Le deuxième chapitre rappelle, premièrement, les notions de bases de régression linéaire (régression linéaire simple et multiple), Le dernier chapitre est consacré à l'implémentation. Enfin, ce mémoire s'achève par une conclusion générale.

Chapitre 1

OPTIMISATION SANS CONTRAINTES

Définition : Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une fonction continue. On définit un problème d'optimisation sans contraintes par : $\begin{cases} \min f(x) \\ x \in \mathbb{R}^n \end{cases}$ où $\begin{cases} \max f(x) \\ x \in \mathbb{R}^n \end{cases}$ où f est dite fonction objectif.

Remarque : on a $minf(x) = -\max(-f(x))$. Puisque la minimisation de f est équivalente à la maximisation de -f, une méthode pour trouver le minimum suffit à résoudre le problème d'optimisation. Ce qui justifie désormais(et sauf mention contraire)de traiter dans toute la suite de ce chapitre uniquement les problèmes de minimisation.
Définition :
-On dit que x^* est un minimun local si $f(x) \geq f(x^*), \forall x \in V(x^*)$, où $V(x^*)$ est un voisinage de x^*. (x^* est un maximum local si $f(x) \leq f(x^*), \forall x \in V(x^*)$)
-On dit que x^* est un minimum global si $f(x) \geq f(x^*), \forall x \in \mathbb{R}^n$. ($x^*$ est un maximum global si $f(x) \leq f(x^*), \forall x \in \mathbb{R}^n$).

1.1 Condition nécessaire du premier ordre

[1]

Théorème 1.1.1. *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une fonction différenciable en x^*. x^* est un minimum local alors $\nabla f(x^*) = 0$*
Démonstration :
On écrit la formule de Taylor :

$$f(x^* + td) = f(x^*) + \nabla f(x^*)td + o(\|td\|), t > 0, d \in \mathbb{R}^n$$

comme x^* est un minimum local alors $f(x) - f(x^*) \geq 0$, $\forall x \in V(x^*)$. pour x=$x^* + td$ avec $t > 0$ assez petit et pour un quelconque $d \in \mathbb{R}^n$, on aura :

$$0 \leq f(x^* + td) - f(x^*) = \nabla f(x^*)td + o(\|td\|), t > 0, d \in \mathbb{R}^n.$$

Par conséquent :

$$\frac{f(x^* + td) - f(x^*)}{t} = \frac{\nabla f(x^*)td}{t} + \frac{o(\|td\|)}{t}$$

En passant à la limite quand $t \longrightarrow 0^+$, *on obtient :*

$$0 \leq \lim_{t \longrightarrow 0^+} \frac{f(x^* + td) - f(x^*)}{t} = \lim_{t \longrightarrow 0^+} (\nabla f(x^*)d + \frac{o(\|td\|)}{t}).$$

Comme

$$\lim_{t \longrightarrow 0^+} (\frac{o(\|td\|)}{t}) = 0,$$

on aura :

$$\nabla f(x^*)d \geq 0$$

En particulier pour $d = -\nabla f(x^*)$, *on obtient :*$-\|\nabla f(x^*)\|^2 \geq 0$, *par conséquent* $\nabla f(x^*) = 0$

Remarque :

- Cette condition est nécessaire mais non suffisante.
- Les points x qui vérifient $\nabla f(x) = 0$ sont appelés points critiques ou points stationnaires de f. Parmi les points critiques, on peut avoir des minima, des maxima ou bien des points d'inflexion dans $\mathbb{R}$ ou des points selles(cols)dans $\mathbb{R}^n$

Example :
$f(x) = x^2, \nabla f(x) = 0 \Longleftrightarrow 2x = 0 \Longrightarrow x^* = 0$ est un point critique qui est dans ce cas un minimum.
$.f(x) = -x^2, \nabla f(x) = 0 \Longleftrightarrow -2x = 0 \Longrightarrow x^* = 0$ qui est un point critique qui est dans ce cas un maximum.
$f(x) = x^3, \nabla f(x) = 0 \Longleftrightarrow 3x^2 = 0 \Longrightarrow x^* = 0$ est un point critique qui n'est dans ce cas ni un minimum, ni un maximum mais plutôt un point d'inflexion. Le résultat qui suit montre que dans le cas d'une fonction convexe différenciable, la condition nécessaire de premier ordre devient suffisante, et le minimum recherché devient global.

Théorème 1.1.2. *Soit* $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *convexe et différenciable en* x^*. x^* *est un minimum si et seulement si* $\nabla f(x^*) = 0$

Démonstration *:*

1. *Si* x^* *est un minimum global, alors c'est aussi un local, on a* $\nabla f(x^*) = 0$
2. *Pour la réciproque, on utilise l'inégalité de convéxité :*$f(x) \geq f(x^*) + \nabla f(x^*).(x - x^*), \forall x \in \mathbb{R}^n$
   Comme $\nabla f(x^*) = 0$, *donc* $f(x) \geq f(x^*), \forall x \in \mathbb{R}^n$, *qui par définition* x^* *est minimum global*

Example : On considère une fonction de deux variables f définie sur $\mathbb{R}^2, f(x_1, x_2) = x_1^2 + x_2^2$. Cette fonction est convexe car la matrice hessienne de f est définit positive. En effet : $H_{f(x)} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ et d'après le critére de sylvester, les niveaux principaux d'ordre 1 et 2 sont positifs.
$\nabla f(x) = 0 \Longleftrightarrow (2x_1, 2x_2) = (0, 0)$ est un point critique. Comme f est convexe, alors x^* est un minimum global de f.

1.2 Condition nécessaire du second ordre

[1]

Théorème 1.2.1. *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une fonction de classe C^2. x^* est un minimum local, alors*
a)$\nabla f(x^*) = 0$.
b)$H_{f(x^*)}$ est semi-définie positive.

Remarque :
Dans ce théorème, lorsque x^* est un maximum local, alors la condition b)devient $H_{f(x^*)}$ est semi-définie négative.
Démonstration :
a)est déjà démontré (voir le théorème 1.1.2)
Pour démontrer b), on écrit la formule de Taylor à l'ordre 2 pour la fonction f au voisinage de x^*. soit $t \in \mathbb{R}$, $d \in \mathbb{R}^n$, posons $x = x^* + td$

$$f(x^* + td) = f(x^*) + \nabla f(x^*)td + \frac{1}{2}td^T.H_{f(x^*)}.td + o(\|td\|^2),$$

Pour un $t > 0$ assez petit et pour un quelconque $d \in \mathbb{R}^n$. Ce qui donne

$$0 \leq f(x^* + td) - f(x^*) = \frac{1}{2}t^2d^T.H_{f(x^*)}.d + o(\|td\|^2)$$

$$0 \leq \frac{f(x^* + td) - f(x^*)}{t^2} = \frac{1}{2}d^T.H_{f(x^*)}.d + (\frac{O(\|td\|^2)}{t^2})$$

Par passage à la limite, on obtient :

$$0 \leq \lim_{t \longrightarrow 0^+} \frac{f(X^* + td) - f(X^*)}{t^2} = \lim_{t \longrightarrow 0^+} (\frac{1}{2}d^T.H_{f(x^*)}.d + \frac{O(\|td\|^2)}{t^2})$$

Par conséquent :

$$0 \geq \frac{1}{2}d^T H_{f(x^*)}.d, \forall d \in \mathbb{R}^n,$$

car

$$\lim_{t \longrightarrow 0^+} \frac{O(\|td\|^2)}{t^2} = 0.$$

On déduit que $H_{f(x^*)}$ est semi-définie positive.

Example : on considère la fonction :
$f(x_1, x_2) = x_1^2 - x_2^2$. On a $\nabla f(x) = 0 \Longleftrightarrow (2x_1, -2x_2) = (0, 0) \Longrightarrow$

$x^* = (x_1^*, x_2^*) = (0, 0)$ est un point critique. La matrice hessienne de f appliquée au point x^* est donnée par : $H_{f(x^*)} = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$
Cette matrice est non-définie. Donc elle n'est pas semi-définie positive, et d'aprés le théoréme 1.3, la condition nécessaire du 2éme ordre n'est pas vérifiée. On conclut donc que x^* n'est pas un minimum local.
De la même maniére, cette matrice qui est non-définie, n'est également pas semi-définie négative, et toujours d'aprés le théorème 1.3, la condition nécessaire du 2éme ordre n'est pas vérifiée. On conclut donc que x^* n'est pas un maximum local. Comme x^* n'est ni un maximum, ni un minimum, c'est donc un point selle.

1.3 Condition suffissante du second ordre

[1]

Théorème 1.3.1. *soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une fonction de classe C^2, Soit x^* vérifiant*
a)$\nabla f(x^*) = 0$
b)$H_{f(x^*)}$ est définie positive alors x^* est un minimum local stricte .

Remarque :
Dans ce théorème, lorsque la condition b) devient $H_{f(x^*)}$ est définie négative, alors x^* est un maximum local.

Démonstration :
On écrit la formule de Taylor pour f à l'ordre 2 au voisinage de x^* en posant $x = x^* + d, d \in V_0$:

$$f(x^* + d) = f(x^*) + \nabla f(x^*).d + \frac{1}{2} d^T H_{f(x^*)}.d + o(\|d\|)^2, \forall d \in V_0$$

Par conséquent :

$$f(x^* + d) - f(x^*) = \frac{1}{2} d^T H_{f(x^*)} \times d + o(\|d\|)^2, \forall d \in V_0$$

Ce qui signifie que

$$signe(f(x^* + d) - f(x^*)) = signe(d^T.H_{f(x^*)}.d)$$

Comme $H_{f(x^*)}$ est définie positive, alors $d^T.H_{f(x^*)}.d > 0, \forall d \neq 0$. On a le sign$(f(x^*+d)-f(x^*))$ est positif, ce qui implique que X^* est un minimum local.

Example : $f(x_1, x_2) = x_1^3 + 2x_2^3 + x_1^2 + x_2^2$. On a $\nabla f(x) = 0 \Longleftrightarrow (3x_1^2 + 2x_1, 6x_2^2 + 2x_2) = (0,0) \Longrightarrow$

$x^{1*} = (0,0), x^{2*} = (\frac{-2}{3}, \frac{-1}{3}), x^{3*} = (\frac{-2}{3}, 0), x^{4*} = (0, \frac{-1}{3})$

sont les points critiques. La matrice hessienne de f appliquée au point x est donnée par :

$$H_{f(x)} = \begin{pmatrix} 6x_1 + 2 & 0 \\ 0 & 12x_2 + 2 \end{pmatrix}$$

Pour le point $x^{1*}, H_{f(x^{1*})} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ est une matrice définie positive, il s'en suit que $x^{1*} = (0,0)$ est un minimum local. Pour le point $x^{2*}, H_{f(x^{2*})} = \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix}$ est une matrice définie négative, il s'en suit que $x^{2*} = (\frac{-2}{3}, \frac{-1}{3})$ est un maximum local.

Pour les 2 points x_{3*} et x_{4*}, on obtient respectivement les matrices $H_{f(x^{3*})} = \begin{pmatrix} -2 & 0 \\ 0 & 2 \end{pmatrix}$ et $H_{f(x^{4*})} = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$ qui sont des matrices non-définies. La condition nécessaire du 2ème ordre n'est pas vérifiée, et d'aprés le théorème 1.3.1, on conclut que x_{3*} et x_{4*} sont des points selles.

1.4 Méthodes numériques en optimisation sans contraintes

Définition :[5] La dérivée directionnelle de la fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ au point x dans la direction d est donnée par :

$$\lim_{t \longrightarrow 0^+} \frac{f(x+td) - f(x)}{t} = \nabla f(x).d$$

Elle est aussi appelée taux de variation de f dans la direction d. En utilisant l'inégalité de Cauchy-Schwartz, on aura :

$$|\nabla f(x).d| \leq \|\nabla f(x)\|.\|d\|.$$

Lorsque d est une direction normalisée -ie- $\|d\| \leq 1$, alors $|\nabla f(x).d| \leq \|\nabla f(x)\|$. C'està dire que le taux de variation de f dans une direction normalisée est toujours inférieur à la norme du gradient. Lorsque on choisit une direction du gradient normalisée $d = \frac{\nabla f(x)}{\|\nabla f(x)\|}$, alors le taux de variation devient

$$\nabla f(x).d = \nabla f(x).\frac{\nabla f(x)}{\|\nabla f(x)\|} = \frac{\|\nabla f(x)\|^2}{\|\nabla f(x)\|} = \|\nabla f(x)\|.$$

C'està dire que le taux de variation de f dans la direction du gradient est maximal par rapport

à toutes les autres directions.
C'est à dire que la fonction croit le plut rapidement possible dans la direction du gradient. Elle décroît le plut rapidement possible dans la direction du l'anti-gradient -ie-$d = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$.
On appelle $d = \frac{\nabla f(x)}{\|\nabla f(x)\|}$ la direction de montée et $d = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$ la direction de descente.
Dans toute la suite du cours, on ne s'interessera qu'à la direction de descente puisqu'il s'agit de trouver un minimum pour la fonction f.
Définition :[2] Un algorithme est définit par une application $A : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ permettant de générer une suite d'éléments de $\mathbb{R}^n$ par la formule :

$$\begin{cases} x^0 \in \mathbb{R}^n, & \text{étape d'initialisation} \\ x^{k+1} = A(x^k), & \text{itération k=0,1,2..} \end{cases}$$

Étudier la convergence d'un algorithme c'est étudier la convergence de la suite $\{x^k\}$ dans $\mathbb{R}^n$.

Définition : Soit $\{x^k\}$ une suite dans $\mathbb{R}^n$ de limite x^* définit par un algorithme convergent A. on dit que la convergence de A est :

1. linéaire si l'erreur $e_k = \|x^k - x^*\|$ décroit linéairement -ie- $e_{k+1} \leq c.e_k$, avec $0 < c < 1$;
2. super-linéaire si $e_{k+1} \leq \alpha_k.e_k$, avec $\alpha_k \longrightarrow 0$ lorsque $k \longrightarrow +\infty$;
3. d'ordre p si $e_{k+1} \leq c.(e_k)^p$, avec $0 < c < 1$. Pour $p = 2$, on dit que la convergence est quadratique.

Définition : Soit A un algorithme convergent. On dit que A est globalement convergent si quel que soit le point de depart $x^0 \in \mathbb{R}^n$, la suite x^k générée par l'algorithme A converge. Sinon, on dit que l'algorithme est localement convergent.

1.4.1 Méthode du gradient

[2]La méthode(ou l'algorithme)du gradient fait partie d'une classe plus grande de méthode numériques applées méthode de descente. Considérons le problème :

$$(P) \begin{cases} \min f(x) \\ x \in \mathbb{R}^n \end{cases}$$

Soit $x^0 \in \mathbb{R}^n$ un point de départ. On construit un point $x^1 = x^0 - \alpha_0 \nabla f(x^0)$, où α_0 est appelé pas de descente. $-\nabla f(x^0)$ est la direction de descente. α_0 est donné par la résolution du problème unidimensionnel suivant : $\min_{\alpha>0} f(x^0 - \alpha \nabla f(x^0))$.
On adopte l'écriture suivant $\alpha_0 = arg \min_{\alpha>0} f(x^0 - \alpha \nabla f(x^0))$
À l'itération k, connaissant x^k, on calcule $x^{k+1} = x^k - \alpha_k \nabla f(x^k)$. α_k est donné par la résolution du problème d'optimisation $\min_{\alpha>0} f(x^k - \alpha \nabla f(x^k))$
Cette méthode ainsi décrite est appelée méthode du gradient à pas optimal puisque le pas α_k est obtennu par optimisation.
Pour une implémentation sur machine on adopte les tests d'arrêt :

1. $\|x^{k+1} - x^k\| < \epsilon$;
2. $\|\nabla f(x^{k+1})\| < \epsilon$;
3. $|f(x^{k+1}) - f(x^k)| < \epsilon$

où ϵ est fixé à l'avance.

1.4.2 Algorithme du gradient à pas optimal

1. Initialisation : soit $\epsilon > 0$ fixé, choisir $x^0 \in \mathbb{R}^n$.
2. Itération k=0,1,2...

$$\begin{cases} x^{k+1} = x^k - \alpha_k \nabla f(x^k) \\ \alpha_k = Arg\min_{\alpha>0} f(x^k - \alpha \nabla f(x^k)) \end{cases}$$

3. Si l'un des tests d'arrêt est vérifié stop x^{k+1} est la solution recherchée ; sinon aller en 2., en passant à une autre itération en posant $k = k + 1$.

Remarque : La convergence de la méthode du gradient à pas optimal est linéaire, donc assez lente.
Remarque : Le calcul de α_k d'une façon exacte est parfois compliqué et trés coûteux en temps de calculs machine. C'est pour cela qu'en pratique, on se contentera d'une valeur fixe pour α_k (méthode du gradient à pas constant), ou d'une valeur acceptable α_k(méthode D'Armijo)

1.4.3 Algorithme du gradient à pas constant

On prend $\alpha_k = \alpha$, constant, mais pour calculer cette constante, la fonction f doit vérifier les conditions suivantes :

1. f de classe C^1 sur $\mathbb{R}^n$
2. $\|\nabla f(x) - \nabla f(y)\| \leq .\|x - y\|$ c'est à dire que le gradient vérifie la condition de Lipschitz avec Ł comme constante de Lipschitz.
3. $(\nabla f(x) - \nabla f(y)).(x - y) \geq \phi\|x - y\|^2$, $\forall x, y \in \mathbb{R}^n$ pour une constante ϕ positive. On posera alors $\alpha = \frac{\phi}{2}$ et $x^{k+1} = x^k - \frac{\phi}{2}\nabla f(x^k)$

1.4.4 Méthode D'Armijo pour le calcul d'un pas acceptable

On se fixe $\beta \in]0, 1[$, (généralement on prend $\beta = \frac{1}{2}$ et on pose $\alpha_0 = 1$). Pour trouver α_k acceptable à chaque itération, on considère l'ensemble

$$E = \left\{ \ \alpha_i = \frac{\alpha_0}{2^i}, \quad \text{i=0,1,2...} \right.$$

et on choisit le plus grand α_i dans E vérifiant la condition :

$$f(x^k - \alpha_i \nabla f(x^k)) \leq f(x^k) - \beta\alpha_i \|\nabla f(x^k)\|^2$$

. Cette inégalité est appelé inégalité d'Armijo.

1.4.5 Méthode de Newton-Raphson dans $\mathbb{R}$(cas d'une fonction scalaire

Soit f une fonction réelle d'une variable réelle. $f : \mathbb{R} \longrightarrow \mathbb{R}$ de classe C^1 dont on sait qu'elle admet au moins un zéro, c'est à dire $\exists x^* \in \mathbb{R}/f(x^*) = 0$.
Soit x^0 un point initial. Une approximation linéaire de f au $V_{(x^0)}$ est donnée à partir de la formule de Taylor à l'ordre 1.

$$f(x) = f(x^0) + (x - x^0).f'(x^0).$$

On suppose que $f'(x^0) \neq 0$, sinon on prend un autre point initial.
f est linéaire, on peut calculer son unique zéro qu'on notera x^1 vérifiant $f_l(x^1) = 0$.
On obtient :

$$x^1 = x^0 - \frac{f(x^0)}{f'(x^0)}$$

Géométriquement $f(x) = f(x^0) + (x - x^0).f'(x^0)$ est l'équation de la tangente à la courbe de f au point x^0. Cette droite coupe l'axe des abscisses au point x^1. En répétant le procédé, on génére une suite de points(algorithmes de Newton Raphson) $\{x^k\}_{k\in\mathbb{N}}$ telle que,

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}.$$

Remarque :La méthode de Newton-Raphson calcule un zéro de f c'est à dire trouver $x^*/f(x^*) = 0$.
Cette méthode est également appelée méthode de la tangente.

1.4.6 Méthode de Newton dans $\mathbb{R}^n$(cas d'une fonction vectorielle)

Soit $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$
$x = (x_1, ..., x_n) \longrightarrow (f_1(x), f_2(x), ..., f_n(x))$. Une fonction qui possède au moins une racine x^*. On suppose que la matrice jacobienne JF(x^k) est inversible pour tout $x^k \in V(x^*)$. Pour un point initial x^0. On considére l'approximation linéaire de F au $V(x^0)$:

$$F(x) = F(x^0) + JF(x^0).(x - x^0).$$

Un zéro de F est donné par l'expression :

$$x^1 = x^0 - (JF(x^0)^{-1}.F(^{x^0}))$$

En réitérant le procédé, on génére une suite de points $\{x^k_{k\in N}\}$ selon la formule :

$$x^{k+1} = x^k - (JF(x^k)^{-1}.F(^{x^k}))$$

Remarque : La méthode de Newton dans $\mathbb{R}^n$ est une généralisation de la méthode de Newton-Raphson dans $\mathbb{R}$. Elle converge d'une façon quadratique à condition que x^0 soit dans $V_{(x^0)}$.

1.4.7 Méthode de Newton appliquée en optimisation

Soit le problème d'optimisation sans contrainte suivant :

$$\begin{cases} minf(x) \\ x \in \mathbb{R}^n \end{cases}$$

Une condition nécessaire pour un minimum x^* est que $\nabla f(x^*) = 0$.
On pose $F(x) = \nabla f(x)$ avec :
$\nabla f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$
$x \longrightarrow (\frac{\partial f}{\partial x_1}..., \frac{\partial f}{\partial x_n})$
Il est clair que $\nabla f(x^*) = 0 \Longleftrightarrow F(x^*) = 0$ et $JF(\overline{x}) = J(\nabla f(\overline{x})) = H_{f(\overline{x})}, \forall \overline{x} \in \mathbb{R}^n$

1.4.8 Algorithme de Newton en optimisation

1. Initialisation :fixer $\epsilon > 0$, et choisir x^0
2. Itération k=0,1,2...
   $x^{k+1} = x^k - (H_{f(x^k)})^{-1}.\nabla f(x^k)$
3. Si $\|x^{k+1} - x^k\| < \epsilon$ stop, x^{k+1} est la solution recherchée ; sinon poser $k = k + 1$, aller en 2.

Remarque : Pour des problèmes conséquents, le calcul de $(H_{f(x^k)})^{-1}$ est beaucoup trop coûteux en temp de calcul et parfois impossible. c'est pour cela qu'en pratique, on utilise une approximation B_k de l'inverse de la matrice hessienne, on aboutira alors à des méthodes appelées méthodes de quasi-Newton. On trouve notamment deux méthodes de calcul de B_k en fonction de $B_{k-1}, \nabla f(x^k), \nabla f(x^{k-1}), x^k$ et x^{k-1}.
La formule de Davidon-Fletcher-Powell(DFP) et la formule de Broyden-Fletcher-Goldfarb-Shanno(BFGS). Ces formules donnent toujours des matrices définies positives. À l'aide de ces estimations, on établit un algorithme à convergence trés rapide et particulièrement robuste, Son unique inconvénient est la nécessité du stockage en mémoire d'une matrice carrée d'ordre n. Dans la pratique, on utilise en général la méthode BFGS qui est plus efficace.

Remarque : Dans le cas où $H_{f(x^k)}$ n'est pas définie positive, on utilise la méthode de levenberg-Marquard qui consiste à trouver, à chaque itération, un paramétre μ_k qui rend la matrice $H_{f(x^k)} + \mu_k.I_n$ définie positive (I_n étant la matrice identité d'ordre n)
La méthode de levenberg-Marquardt est particulièrement robuste et efficace.

1.4.9 Méthode de gradient conjugué

[7]L'algorithme que nous présentons dans cette section posséde deux intéret. Il permet de résoudre des systémes linéaires à coefficients strictement positif de grande taille et il sert d'algorithme de base pour la résolution des problémes d'optimisation non linéaire. Soit le probléme de minimisation quadratique suivant :
$f(x) = \frac{1}{2}x^tAx + b^tx + c$ avec, $x \in \mathbb{R}^n$
où A est une matrice carrée d'ordre n, symétrique, définie positive et $b \in \mathbb{R}^n$. On note :
$\nabla f(x) = Ax + b$ qui est non nul et $\nabla^2 f(x) = A$
La première idée fondamentale de l'algorithme du gradient conjugué consiste à choisir chaque direction de descente conjugué à la direction de descente précédente par rapport à A. Afin de développer le gradient conjugué, introduisons la notion de direction conjugué.
Définition : Soient $A \in \mathbb{R}^n \times \mathbb{R}^n$ une matrice définie positive. Les vecteurs(ou directions)non nuls de $R^n, d^1, ..., d^k$ sont conjugués par rapport à A(A conjuguées)si : $(d^i)^tAd^j = 0$, $\forall i, j$ tels que :
$i \neq j$ Ceci signifie que ces deux vecteurs sont orthogonaux pour le produit scalaire associé à la matrice A défini par :
$< x, y >_A = x^tAy; \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n$

Théorème 1.4.1. *Soit $d^1, .., d^k$ un ensemble de directions non nulles et conjuguées par rapport à A. Alors les vecteurs $d^1, .., d^k$ sont linéairement indépendants.*

Algorithme du gradient conjugué :

1. Initialisation fixer $\epsilon > 0$, choisir $x^0 \in \mathbb{R}^n$, Poser $\nabla f(x^0) = Ax^0 + b$ et $d^0 = -\nabla f(x^0)$
2. Itération $k = 0, 1, ...$
   Calculer $\alpha_k = -(d^k)^t \nabla f(x^k)/(d_k)^t Ad^k$
   Calculer $x^{k+1} = x^k + \alpha_k d^k$
   Calculer $\nabla f(x^{k+1}) = f(x^{k+1}) + \alpha_k Ad^k$
   Calculer $\beta^{k+1} = \|\nabla f(x^{k+1})\|^2 / \|\nabla f(x^k\|^2$
   Calculer $d^{k+1} = -\nabla f(x^{k+1}) + \beta^{k+1} d^k$
3. Critère d'arrêt
   Si : $\|x^{k+1} - x^k\| < \epsilon$ Stop
   Sinon on pose $k = k + 1$ et aller à 2.

1.4.10 méthode du Quasi-Newton

[6] Le couplage de la méthode de Newton avec la recherche linéaire de Wolfe a permis de construire une méthode globalement convergente. Les méthodes de quasi-Newton ont été développées pour pallier d'autres inconvénients de la méthode de Newton : en particulier le problème du calcul de la matrice hessienne qui n'est pas toujours possible ou conseillé. Ces méthodes se concentrent donc sur la construction itérative de matrices H_K approchant la hessienne, ou de matrices $_K$ approchant l'inverse de la hessienne
Equation de sécante et approximation :
Comment calculer une approximation H_{k+1} de la matrice hessienne $H[f](x_{k+1})$ connaissant

x_k, x_{k+1}, $\nabla f(x_k)$, et $\nabla f(x_{k+1})$?.
Ecrivons le développement limité de ∇f au voisinage de x_{k+1} et appliqué en x_k :

$$\nabla f(x_k) = \nabla f(x_{k+1}) + H_{[F]}(x_{k+1})(x_k - x_{k+1}) + o(x_k - x_{k+1}).$$

D'où :

$$H_{[f]}(x_{k+1})(x_k - x_{k+1}) \simeq \nabla f(x_{k+1}) - \nabla f(x_k).$$

On construit une approximation H_{k+1} de $H_{[f]}(x_{k+1})$ comme solution de l'équation :

$$\nabla f(x_{k+1}) - \nabla f(x_k) = H_{k+1}(x_{k+1} \times x_k). \tag{1.1}$$

appelée **équation de la sécante ou équation de Quasi-Newton**. De façon similaire, on peut construire une approximation B_{k+1} de $H[f](x_{k+1})^{-1}$, a comme solution de l'équation :

$$B_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k. \tag{1.2}$$

Dans les deux cas, les équations de quasi-Newton forment un système sous-déterminé à n équations et n^2 inconnues. Il existe donc une infinité de matrices H_{k+1} pouvant convenir.

Mises à jour DFP et BFGS :

L'idée proposée par Broyden (1965) est de choisir parmi ce nombre infini de modèles linéaires celui qui est le plus proche du modèle établi à l'itération précédente, conservant ainsi le plus possible ce qui a déjà été calculé. On cherche donc à résoudre :

$$\begin{cases} min_H \frac{1}{2}\|H - H_k\|_F^2, \\ s \times t \times \nabla f(x_{k+1}) - \nabla f(x_k) = H(x_{k+1} - x_k), \end{cases}$$

où $\|.\|_F$ désigne la norme de Frobenius. Il s'agit d'un problème quadratique strictement convexe dont l'unique solution est donnée par :

$$H_{K+1} = H_K + \frac{(y_K - H_k\sigma_K)\sigma_K^T}{\sigma_K^T\sigma_K},$$

Où :

$$\begin{cases} \sigma_k = x_{k+1} - x_k & . \\ y_k = \nabla f(x_{k+1}) - \nabla f(x_k) & . \end{cases}$$

Le problème de cette méthode est que la matrice H_k ainsi construite, n'est en général pas symétrique, ni définie positive. La hessienne d'une fonction de classe C^2 étant symétrique, il est naturel d'imposer à son approximation de l'être également. Nous cherchons donc H_{k+1} comme solution du problème :

$$(p) \begin{cases} min_H \frac{1}{2}\|H - H_k\|^2 \\ s.t. y_k = H_{\sigma_k}, H^T = H \end{cases}$$

De nombreuses normes matricielles peuvent être utilisées, et conduisent à des méthodes de Quasi-Newton différentes. Les méthodes les plus utilisées aujourd'hui sont les méthodes DFP et BFGS obtenues en choisissant comme des normes de la forme :

$$\|A\|_W = \|W^{\frac{1}{2}} A W^{\frac{1}{2}}\|_F$$

où W est une matrice symétrique inversible vérifiant :$y_k = W\sigma_k$. La résolution(longue et technique)du problème(P)conduit alors aux formules de mise à jour suivantes :

Méthode DFP(DAVIDSON,FLETCHER,POWELL 1959-63)

$$H_{k+1} = H_k + \frac{y_k y_k^T}{y_k^T \sigma_k} - \frac{H_k \sigma_k \sigma_k^T H_k}{\sigma_k^T H_k \sigma_k}$$

Méthode BFGS(Broyden,Fletcher,Goldfarb,Shannon.1969-70)

$$B_{k+1} = (I - \frac{\sigma_k . y_k^T}{y_k^T . \sigma_k})^T . B_k (I - \frac{\sigma_k . y_k^T}{y_k^T . \sigma_k}) + \frac{\sigma_K \sigma_K^T}{y_k^T \sigma_k}$$

Ces deux formules sont duales l'une de l'autre dans le sens où l'on obtient la formule BFGS en inversant la relation DFP avec : $H_k = B_k^{-1}$. On a alors également : $H_{k+1} = B_{k+1}^{-1}$

1.5 CONCLUSION

Dans ce chapitre, on a discuté sur les conditions d'optimalité en optimisation sans contraintes, ainsi on a abordé quelques méthodes itératives utilsé pour résoudre des problémes d'optimisation sans contraintes, comme la méthoe du gradient conjugué et de quasi Newton .

Chapitre 2

RÉGRESSION LINÉAIRE SIMPLE ET MULTIPLE

2.1 la régresion linéaire simple

Introduction

[4] Commençons par un exemple afin de fixer les idées. Pour des raisons de santé publique, on s'intéresse à la concentration d'ozone O_3 dans l'air(en microgrammes par millilitre). En particulier, on cherche à savoir s'il est possible d'expliquer le taux maximal d'ozone de la journée par la température T_{12} à midi. Les données sont :
D'un point de vue pratique, le but de cette régression est double :

Température à 12h	23.8	16.3	27.2	7.1	25.1	27.5	19.4	19.8	32.2	20.7
O_3 max	115.4	76.8	113.8	81.6	115.4	125	83.6	75.2	136.8	102.8

FIGURE 2.1 – 10 données journalières de température et d'ozone

Ajuster un modèle pour expliquer O_3 en fonction de T_{12}.
prédire les valeurs d'O_3 pour de nouvelles valeurs de T_{12}. Avant toute analyse, il est intéressant de représenter les données, comme sur la figure 2.1.
Pour analyser la relation entre les x_i(température) et les y_i (ozone), nous allons chercher une fonction f telle que :
$y_i \approx f(x_i)$. Pour preciser le sens de $\approx$, il faut se donner un critére quantifiant la qualité de l'ajustement de la fonction f aux données. Il conviendra aussi de se donner une classe de fonctions $\mathcal{F}$ dans laquelle est supposée vivre la vraie fonction inconnue.
Le problème mathématique peut alors s'écrire de la façon suivante :

$$\arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} L(y(i) - f(x_i))$$

Où n réprésente le nombre de données disponibles(taille de l'échantillon) et $L(.)$ est applée fonction de coût ou fonction de perte(Loss en anglais).

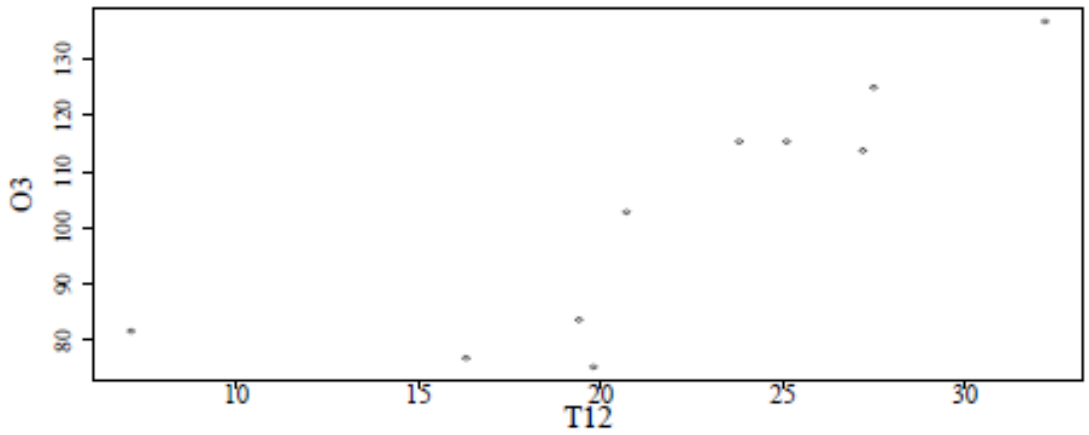

FIGURE 2.2 – 10 données journalières de température et d'ozone

2.1.1 La modélisation

Dans de nombreuses situations, en première approche, une idée naturelle est de supposer que la variable à expliquer y est une fonction affine de la variable explicative x, c'est-à-dire de chercher f dans l'ensemble $\mathcal{F}$ des fonctions affines de $\mathbb{R}$ dans $\mathbb{R}$. C'est le principe de la régression linéaires simple. On suppose dans la suite disposer d'un échantillon de n points (x_i, y_i) du plan.
Définition : (Modéle de régression linéaire simple) :
Un modèle de régression linéaire simple est défini par une équation de la forme :

$$\forall i = 1.., n, y_i = \beta_1 + \beta_2 x_i + \epsilon_i.$$

Les quantités ϵ_i viennent du fait que les points ne sont jamais parfaitement alignés sur une droite. On les appelle les erreurs(ou bruits)et elles sont supposées aléatoires. Pour pouvoir dire des choses pertinentes sur ce modéle, il faut néanmoins imposer des hypothéses les concernant. Voici celles que nous ferons dans un premier temps :

$$\mathcal{H} \begin{cases} (\mathcal{H}_1) : E[\epsilon_i] = 0, & \text{pour tous indice i} \\ (\mathcal{H}_2) = cov[\epsilon_i, \epsilon_j] = \delta_{i,j}\sigma^2, & \text{pour tous couple(i,j)} \end{cases}$$

Les erreurs sont donc supposées centrées, de même variance(homoscedasticité) et non correlées entre elles $(\delta_{i,j})$ est le symbole de Kronecker, i.e.$\delta_{i,j} = 1$ si $i = j, \delta_{i,j} = 0$ si $(i \neq j)$. Notons que le modele de regréssion linéaire simple de la définition 2.1, peut encore s'écrire de façon vectorielle :

$$Y = \beta_1 \times 1 + \beta_2 \times X + \epsilon$$

où :
Le vecteur $Y = [y_1, ..., y_n]'$ est aléatoire de dimension n
Le vecteur $1 = [1, 1, ..., 1]'$ est un vecteur de $\mathbb{R}^n$ dont les n composantes valent toutes 1
Le vecteur $X = [x_1, ..., x_n]'$ est un vecteur de n donné(non aléatoire)

Les coéfficient β_1, β_2 sont les paramètres inconnus(mais non aléatoire)du modéle.
Le vecteur $\epsilon = [\epsilon_1, ..., \epsilon_n]'$ est aléatoire de dimension n
Cette notation vectorielle sera commode notamment pour l'interprétation géométrique du problème. Nous y reviendrons en Section 2.1.3, et elle sera d'usage constant en régression linéaire multiple, c'est pourquoi il convient d'ores est déjà de s'y habituer.

2.1.2 Moindres Carrée Ordinaires

Les points (x_i, y_i) étant donnés, le but est maintenant de trouver une fonction affine f telle que la quantité $\sum_{i=1}^{n} L(y_i - f(x_i))$ soit minimale.
Pour pouvoir déterminer f, encore faut-il préciser la fonction de coût L. Deux fonctions sont classiquement utilisées :
Le coût absolu $L(u) = |u|$;
Le coût quadratique $L(u) = u^2$. Les deux ont leurs vertus, mais on privilégiera dans la suite la fonction de coût quadratique. On parle alors de méthode d'estimation par moindres carrés(terminologie due à Legendre dans un article de 1805 sur la détermination des orbites des comètes).
Définition : (Estimateurs des moindres carrés ordinaires) :
On appelle estimateurs des Moindres Carrés Ordinaires(en abregé MCO) $\hat{\beta}_1$ et $\hat{\beta}_2$ les valeurs minimisant la quantité :

$$S(\beta_1, \beta_2) = \sum_{i=1}^{n} (y_i - \beta_1 - \beta_2 x_i)^2.$$

Autrement dit, la droite des moindres carrés minimise la somme des carrés des distances verticales des points (x_i, y_i) du nuage à la droite ajustée $y = \hat{\beta}_1 + \hat{\beta}_2 x$

Calcul des estimateurs de β_1 et β_2

La fonction de deux variables S est une fonction quadratique et sa minimisation ne pose aucun problème, comme nous allons le voir maintenant.

Proposition 2.1.1. *(Estimateurs $\hat{\beta}_1, \hat{\beta}_2$). Les estimateurs des MCO ont pour expressions :*

$$\hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x}$$

avec :

$$\hat{\beta}_2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x}y_i)}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

Preuve la première méthode consiste à remarquer que la fonction $S(\beta_1, \beta_2)$ est strictement convexe, donc qu'elle admet un minimum en un unique point $(\hat{\beta}_1, \hat{\beta}_2)$, lequel est determiné en annulant les derivées partielles de S. On obtient les "équations normales" :

$$\begin{cases} \frac{\partial S}{\partial \beta_1} = -2\sum_{i=1}^{n}(y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i) = 0, \\ \frac{\partial S}{\partial \beta_2} = -2\sum_{i=1}^{n} x_i(y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i) = 0, \end{cases}$$

La première équation donne :

$$\hat{\beta}_1 n + \hat{\beta}_2 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

d'où l'on déduit immédiatement :

$$\hat{\beta}_1 = \bar{y} - \beta_2 \bar{x} \tag{2.1}$$

où $\bar{x}$ et $\bar{y}$ sont comme d'habitude les moyennes empiriques des x_i et y_i. La seconde équation donne :

$$\hat{\beta}_1 \sum_{i=1}^{n} x_i + \hat{\beta}_2 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

et en remplaçant $\hat{\beta}_1$ par son expréssion (2.1), nous avons :

$$\hat{\beta}_2 = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \bar{y}}{\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \bar{x}} = \frac{\sum_{i=1}^{n} x_i (y_i - \bar{y})}{\sum_{i=1}^{n} x_i (x_i - \bar{x})} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})} \tag{2.2}$$

La seconde méthode consiste à appliquer la technique de Gauss de réduction des formes quadratiques, c'est-à-dire à décomposer S(β_1, β_2)en somme de carrés, carrés qu'il ne restera plus qu'à annuler pour obtenir les estimateurs $\hat{\beta}_1$ et $\hat{\beta}_2$.
Dans notre cas, aprés calculs, ceci s'écrit :

$$S(\beta_1, \beta_2) = n(\beta_1 - (\bar{y} - \beta_2 \bar{x}))^2 + (\sum_{i=1}^{n} (x_i - \bar{x})^2)(\beta_2 - \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x}}))^2)$$

+

$$\sum_{i=1}^{n} (y_i - \bar{y})^2 (1 - \frac{(\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}))^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2})$$

où apparaissent deux carrés et un troisième terme independant de β_1 et β_2 : ce dernier est donc incompressible. Par contre, le second est nul si et seulement si $\beta_2 = \hat{\beta}_2$. Ceci étant fait, le premier est alors nul si et seulement si $\beta_1 = \hat{\beta}_1$.
L'éxpression (2.2)de $\hat{\beta}_2$ suppose que le dénominateur $\sum (x_i - \bar{x})^2$ est non nul. Or ceci ne peut arriver que si tous les x_i sont égaux, situation sans intérêt pour notre problème et que nous excluons donc à priori dans toute la suite.
Remarque :

1. La relation $\hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x}$ montre que la droite des MCO passe par le centre de gravité du nuage $(\bar{x}, \bar{y})$

2. Les expressions obtenues pour $\hat{\beta}_1$ et $\hat{\beta}_2$ montrent que ces deux estimateurs sont linéaires par rapport au vecteur $Y = [y_1, ..., y_n]'$.
3. L'estimateur $\hat{\beta}_2$ peut aussi s'écrire comme suit :

$$\hat{\beta}_2 = \beta_2 + \frac{\sum(x_i - \bar{x})\epsilon_i}{\sum(x_i - \bar{x})^2} \tag{2.3}$$

Si cette décomposition n'est pas intéressante pour le calcul effectif de $\hat{\beta}_2$ puisqu'elle fait intervenir les quantités inconnues β_1 et β_2, elle l'est par contre pour démontrer des proprietés théoriques des estimateurs (biais et variance). Son avantage est en effet de mettre en exergue la seule source d'aléa du modele, à savoir les erreurs ϵ_i. Avant de poursuivre, notons que le calcul des estimateurs des moindres carrés est purement detérministe : il ne fait en rien appel aux hypothèses $(\mathcal{H}_1)$ et $(\mathcal{H}_2)$ sur le modele. Celles-ci vont en fait servir dans la suite à expliciter les proprietés statistiques de ces estimateurs.

Quelques propriétée des estimateurs $\hat{\beta}_1$ et $\hat{\beta}_2$

Sous les seules hypothèses $(\mathcal{H}_1)$ et $(\mathcal{H}_2)$ de centrages, decorrélations et homoscedasticités des erreurs ϵ_i du modele, on peut déja donner certaines propriétés des estimateurs. β_1 et β_2 des moindres carrés.

Théorème 2.1.1. *(Estimateurs sans biais) $\hat{\beta}_1$ et $\hat{\beta}_2$ sont des estimateurs sans biais β_1 et β_1*

Preuve Partons de l'écriture (2.3) pour $\hat{\beta}_2$:

$$\hat{\beta}_2 = \beta_2 + \frac{\sum(x_i - \bar{x})\epsilon_i}{\sum(x_i - \bar{x})^2}$$

Dans cette expression, seuls les bruits ϵ_i sont aléatoires, et puisqu'ils sont centrés, on en déduit bien que $E[\hat{\beta}_2] = \beta_2$.Pour $\hat{\beta}_1$, on part de l'expression :

$$\hat{\beta}_1 = \bar{y} - \hat{\beta}_2\bar{x}$$

d'où l'on tire :

$$E[\hat{\beta}_1] = E[\bar{y}] - \bar{x}E[\hat{\beta}_2] = \beta_1 + \bar{x}\beta_2 - \bar{x}\beta_2 = \beta_1$$

On peut également exprimer variances et covariances de nos estimateurs.

Théorème 2.1.2. *(Variance et covariance). Les variances des estimateurs sont :*

$$Var(\hat{\beta}_1) = \frac{\sigma^2 \sum x_i^2}{n\sum(x_i - \bar{x})^2} = \sigma^2(\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2})$$

et

$$Var(\hat{\beta}_2) = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}$$

tandis que leur covariance vaut :

$$Cov(\hat{\beta}_1, \hat{\beta}_2) = -\frac{\sigma^2 \bar{x}}{\sum(x_i - \bar{x})^2}$$

preuve : On part à nouveau de l'expression de $\hat{\beta}_2$ utilisée dans la preuve du non-biais :

$$\hat{\beta}_2 = \beta_2 + \frac{\sum(x_i - \bar{x})\epsilon_i}{\sum(x_i - \bar{x})^2}$$

Or les erreurs ϵ_i sont decorrelées et de même variance σ^2 donc la variance de la somme est la somme des variances :

$$Var(\hat{\beta}_2) = \frac{\sum(x_i - \bar{x})^2\sigma^2}{(\sum(x_i - \bar{x})^2)^2} = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}$$

Par ailleurs, la covariance entre $\bar{y}$ et $\hat{\beta}_2$ s'écrit :

$$Cov(\bar{y}, \hat{\beta}_2) = Cov(\frac{\sum y_i}{n}, \frac{\sum(x_i - \bar{x})\epsilon_i}{\sum(x_i - \bar{x})^2}) = \frac{\sigma^2 \sum(x_i - \bar{x})}{n\sum(x_i - \bar{x})^2} = 0$$

d'où il vient pour la variance de $\hat{\beta}_1$:

$$Var(\hat{\beta}_1) = Var(\frac{\sum y_i}{n} - \hat{\beta}_2\bar{x}) = \frac{\sigma^2}{n} + \frac{\bar{x}^2\sigma^2}{\sum(x_i - \bar{x})^2} - 2\bar{x}Cov(\bar{y}, \hat{\beta}_2)$$

C'est à dire :

$$Var(\hat{\beta}_1) = \frac{\sigma^2}{n} + \frac{\bar{x}^2\sigma^2}{\sum(x_i - \bar{x})^2} = \frac{\sigma^2 \sum x_i^2}{n\sum(x_i - \bar{x})^2}$$

Enfin, pour la covariance des deux estimateurs :

$$Cov(\hat{\beta}_1, \hat{\beta}_2) = Cov(\bar{y} - \hat{\beta}_2\bar{x}, \hat{\beta}_1) = Cov(\bar{y}, \hat{\beta}_2) - \bar{x}Var(\hat{\beta}_2) = -\frac{\sigma^2 \bar{x}}{\sum(x_i - \bar{x})^2}$$

Remarque :

1. On a vu que la droite des MCO passe par le centre de gravité du nuage $(\bar{x}, \bar{y})$. Supposons celui-ci fixé et $\bar{x}$ positif, alors il est clair que si on augmente la pente, l'ordonnée à l'origine va baisser et vice versa, on retrouve donc bien le signe négatif pour la covariance entre $\hat{\beta}_1$ et $\hat{\beta}_2$
2. En statistique inférentielle, la variance d'un estimateur décroît typiquement de façon inversement proportionnelle à la taille de l'échantillon, c'est-à-dire en $\frac{1}{n}$. En d'autres termes, sa précision est généralement en $1\sqrt{n}$.

par exemple l'expression obtenue pour la variance de β_2 :

$$Var(\hat{\beta}_2) = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}$$

Pour comprendre que tout se passe comme d'habitude, il suffit de considerer que les x_i sont eux-mêmes aléatoires, avec écart-type σ_x. Dans ce cas trés genéral, le dénominateur est d'ordre $n\sigma_x^2$ et l'on retrouve bien une variance en $\frac{1}{n}$
Les estimateurs des moindres carrés sont en fait optimaux en un certain sens, c'est ce que précise le résultat suivant.

Théorème 2.1.3. *(Gauss-Markov).*

Parmi les estimateurs sans biais linéaires en y, les estimateurs $\hat{\beta}_j$ sont de variances minimales

Preuve

L'estimateur des MCO s'écrit :

$$\hat{\beta}_2 = \sum_{i=1}^{n} p_i y_i,$$

avec

$$p_i = \frac{(x_i - \bar{x})}{\sum(x_i - \bar{x})^2}.$$

Considérons un autre estimateur $\tilde{\beta}_2$ linéaire en y_i est sans biais, c'est-à-dire :

$$\tilde{\beta}_2 = \sum_{i=1}^{n} \lambda_i y_i$$

Montrons que :

$$\sum \lambda_i = 0$$

et

$$\sum \lambda_i x_i = 1.$$

L'égalité :

$$E(\hat{\tilde{\beta}}_2) = \beta_1 \sum \lambda_i + \beta_2 \sum \lambda_i x_i + \sum \lambda_i E(\epsilon_i) = \beta_1 \sum \lambda_i + \beta_2 \sum \lambda_i x_i$$

.
est vrai pour tous β_2, l'estimateur $\tilde{\beta_2}$ est sans biais donc $E(\tilde{\beta_2}) = \beta_2$ pour tous β_2, c'est- à-dire que $\sum \lambda_i = 0$, et $\sum \lambda_i x_i = 1$
Montrons que :

$$Var(\tilde{\beta_2}) \geq Var(\hat{\beta_2})$$

$$Var(\tilde{\beta_2}) = Var(\tilde{\beta_2} - \hat{\beta_2} + \hat{\beta_2}) = Var(\tilde{\beta_2} - \hat{\beta_2}) + Var(\hat{\beta_2}) + 2Cov(\tilde{\beta_2} - \hat{\beta_2}, \hat{\beta_2})$$

Or :

$$Cov(\tilde{\beta_2} - \hat{\beta_2}, \hat{\beta}2) = Cov(\tilde{\beta_2}, \hat{\beta_2}) - Var(\hat{\beta_2}) = \frac{\sigma^2 \sum \lambda_i (x_i - \bar{x})}{\sum (x_i - \bar{x})^2} - \frac{\sigma^2}{\sum (x_i - \bar{x})^2} = 0$$

la dernière égalité étant due aux deux relations $\sum \lambda_i = 0$ et $\sum \lambda_i x_i$=1. Ainsi :

$$Var(\tilde{\beta_2}) = Var(\tilde{\beta_2} - \hat{\beta_2}) + Var(\hat{\beta_2})$$

Une variance est toujours positive, donc :

$$Var(\tilde{\beta_2}) \geq Var(\hat{\beta_2})$$

Le résultat est demontré. On obtiendrait la même chose pour $\hat{\beta_1}$:(ref4 de regression)

Remarque : Comme nous le verrons au chapitre suivant, on peut en fait montrer un peu mieux : au sens de la relation d'ordre

sur les matrices symétriques reélles, la matrice de covariance de $\hat{\beta} = [\hat{\beta_1}, \hat{\beta_2}]'$ est inférieure à celle de n'importe quel autre estimateur $\tilde{\beta} = [\tilde{\beta_1}, \tilde{\beta_2}]'$ sans biais et linéaire en y.

Calcul des résidus et de variance résiduelle

Dans $\mathbb{R}^2$(espace des variables x_i et y_i), $\hat{\beta_1}$ est l'ordonnée à l'origine et $\hat{\beta_2}$ la pente de la droite ajustée. Cette droite minimise la somme des carrés des distances verticales des points du nuage à la droite ajustée. Notons :
$\hat{y_i} = \hat{\beta_1} + \hat{\beta_2} x_i$ l'ordonnée du point de la droite des moindres carrés d'abscisse x_i, ou valeur ajustée. les résidus sont définis par(cf.figure 2.2) :

$$\hat{\epsilon_i} = y_i - \hat{y_i} = y_i - \hat{\beta_1} - \hat{\beta_2} x_i = (y_i - \bar{y}) - \hat{\beta_2}(x_i - \bar{x}) \tag{2.4}$$

Par construction, la somme des résidus est nulle :

$$\sum_i \hat{\epsilon_i} == \sum_i (y_i - \bar{y}) - \hat{\beta_2} \sum_i (x_i - \bar{x}) = 0$$

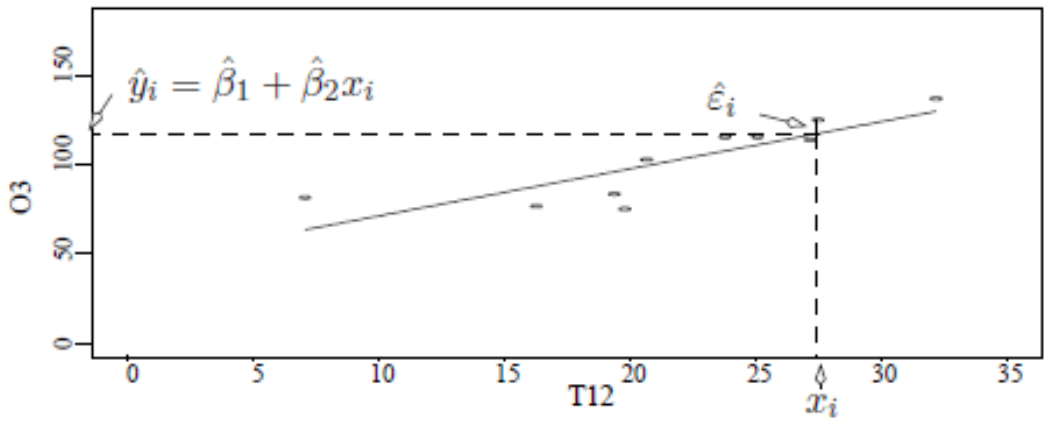

FIGURE 2.3 – Réprésentation des individus

Notons maintenant que les variances et covariance des estimateurs $\hat{\beta}_1$ et $\hat{\beta}_2$ établies en section precédente ne sont pas pratiques car elles font intervenir la variance σ^2 des erreurs, laquelle est en général inconnue. Néanmoins, on peut en donner un estimateur sans biais grâce aux résidus.

Théorème 2.1.4. *(Estimateur non biaisé de σ^2). la statistique : $\hat{\sigma^2} = \sum_{i=1}^{n} \frac{\hat{\epsilon_i^2}}{n-2}$ est un estimateur sans biais de σ^2*

Preuve : Réecrivons les résidus en constatant que $\hat{\beta}_1 = \bar{y} - \hat{\beta}_2\bar{x}$ et $\beta_1 = \bar{y} - \beta_2\bar{x} - \bar{\epsilon}$, ce qui donne :

$$\begin{aligned}\hat{\epsilon}_i &= \beta_1 + \beta_2 x_i + \epsilon_i - \hat{\beta}_1 - \hat{\beta}_2 x_i \\ &= \bar{y} - \beta_2\bar{x} - \bar{\epsilon} + \beta_2 x_i + \epsilon_i - \bar{y} + \hat{\beta}_2\bar{x} - \hat{\beta}_2 x_i \\ &= (\beta_2 - \hat{\beta}_2)(x_i - \bar{x}) + (\epsilon_i - \bar{\epsilon})\end{aligned}$$

En développant et en nous servant de l'écriture vue plus haut :

$$\hat{\beta}_2 = \beta_2 + \frac{\sum(x_i - \bar{x}\epsilon_i)}{\sum(x_i - \bar{x})^2}$$

Nous avons :

$$\sum \hat{\epsilon_i^2} = (\beta_2 - \hat{\beta}_2)^2 \sum (x_i - \bar{x})^2 + \sum (\epsilon_i - \bar{\epsilon})^2 + 2(\beta_2 - \hat{\beta}_2) \sum (x_i - \bar{x})(\epsilon_i - \bar{\epsilon})$$

$$= (\beta_2 - \hat{\beta}_2)^2 \sum (x_i - \bar{x})^2 + \sum (\epsilon_i - \bar{\epsilon})^2 - 2(\beta_2 - \hat{\beta}_2)^2 \sum (x_i - \bar{x})^2$$

prenons en l'espérance :

$$E(\sum \hat{\epsilon_i^2}) = E(\sum(\epsilon_i - \bar{\epsilon})^2) - \sum(x_i - \bar{x})^2 Var(\hat{\beta}_2) = (n-2)\sigma^2$$

Bien sur, lorsque n est grand, cet estimateur différe trés peu de l'estimateur empirique de la variance des résidus, à savoir $\sum_i^n \frac{\hat{\epsilon_i^2}}{n}$

Prévision

Un des buts de la régression est de faire de la prévision, c'est-à-dire de prévoir la variable à expliquer y en présence d'une nouvelle valeur de la variable explicative x. Soit donc x_{n+1} une nouvelle valeur, pour laquelle nous voulons prédire y_{n+1}. Le modèle est toujours le même :

$$y_{n+1} = \beta_1 + \beta_2 x_{n+1} + \epsilon_{n+1}.$$

avec $E[\epsilon_{n+1}] = 0$ et $Var(\epsilon_{n+1}) = \sigma^2$ et $Cov(\epsilon_{n+1}, \epsilon_i) = 0$ pour tous $i = 1, .., n$.
Il est naturel de prédire la valeur corréspondante via le modèle ajusté :

$$\hat{y}_{n+1} = \hat{\beta}_1 + \hat{\beta}_2 x_{n+1}$$

Deux types d'erreurs vont entacher notre prévision : la premiére est due à la non-connaissance de ϵ_{n+1}, la seconde est du à l'incertitude sur les estimateurs $\hat{\beta}_1$ et $\hat{\beta}_2$.

Proposition 2.1.2. *(Erreur de prévision) : L'erreur de prévision* $\hat{\epsilon}_{n+1} = (y_{n+1} - \hat{y}_{n+1})$ *satisfait les propriétés suivantes :*

$$\begin{cases} E[\hat{\epsilon}_{n+1}] = 0, \\ Var(\hat{\epsilon}_{n+1}) = \sigma^2(1 + \frac{1}{n} + \frac{(x_{n+1}-\bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}), \end{cases}$$

Preuve :Pour l'espérance, il suffit d'utiliser le fait que ϵ_{n+1} est centrée et que les estimateurs
$\hat{\beta}_1$ et $\hat{\beta}_2$ sont sans biais :

$$E[\hat{\epsilon}_{n+1}] = E[\beta_1 - \hat{\beta}_1] + E[\beta_2 - \hat{\beta}_2]x_{n+1} + E[\epsilon_{n+1}] = 0$$

Nous obtenons la variance de l'erreur de prévision en nous servant du fait que y_{n+1} est fonction de ϵ_{n+1} seulement tandis que $\hat{y}_{n+1}$ est fonction des autres erreurs $(\epsilon_i)_{1 \leq i \leq n}$:
$Var(\hat{\epsilon}_{n+1}) = Var(y_{n+1} - \hat{y}_{n+1}) = Var(y_{n+1}) + Var(\hat{y}_{n+1}) = \sigma^2 + Var(\hat{y}_{n+1})$

Calculons le second terme :
$Var(\hat{y}_{n+1}) = Var(\hat{\beta}_1 + \hat{\beta}_2 x_{n+1})$

$= Var(\hat{\beta}_1) + x_{n+1}^2 Var(\hat{\beta}_2) + 2x_{n+1} Cov(\hat{\beta}_1, \hat{\beta}_1)$

$= \frac{\sigma^2}{\sum(x_i - \bar{x})^2}(\frac{\sum x_i^2}{n} + x_{n+1}^2 - 2x_{n+1}^2 \bar{x})$

$= \frac{\sigma^2}{\sum(x_i - \bar{x})^2}(\frac{(\sum(x_i - \bar{x})}{n} + \bar{x}^2 + x_{n+1}^2 - 2x_{n+1}\bar{x}))$

$= \sigma^2(\frac{1}{n} + \frac{(x_{n+1} - \bar{x})^2}{\sum(x_i - \bar{x})^2})$

Au total, on obtient bien :

$$Var(\hat{\epsilon}_{n+1}) = \sigma^2(1 + \frac{1}{n} + \frac{(x_{n+1} - \bar{x})^2}{\sum(x_i - \bar{x})^2})$$

Ainsi la variance augmente lorsque x_{n+1} s'éloigne du centre de gravité du nuage. Autrement dit, faire de la prévision lorsque x_{n+1} est loin de $\bar{x}$ est périlleux, puisque la variance de l'erreur de prévision peut être très grande !Ceci s'explique intuitivement par le fait que plus une observation x_{n+1} est éloignée de la moyenne $\bar{x}$ et moins on a d'information sur elle.

2.1.3 Interprétation géométrique

Représentation des variables

Si nous abordons le probléme d'un point de vue vectoriel, nous avons deux vecteurs à notre disposition : le vecteur $X = [x_1, ..., x_n]'$ des n observations pour la variable explicative et le vecteur $Y = [y_1, ..., y_n]'$ des n observations pour la variableà expliquer. Ces deux vecteurs appartiennent au même espace $\mathbb{R}^n$: l'espace des variables. Si on ajoute à cela le vecteur $1 = [1, ..., 1]'$, on voit tout d'abord que par l'hypothése selon laquelle tous les x_i ne sont pas égaux, les vecteurs 1 et X ne sont pas colinéaires : ils engendrent donc un sous-espace de $\mathbb{R}^n$ de dimension 2, note $\mathcal{M}(X)$. On peut projeter orthogonalement le vecteur Y sur le sous-espace $\mathcal{M}(X)$
notons provisoirement. $\tilde{Y}$ ce projeté. Puisque (1,X)forme une base de $\mathcal{M}(X)$, il existe une unique décomposition de la forme. $\tilde{Y} = \tilde{\beta}_1 1 + \tilde{\beta}_2$, Par définition du projeté orthogonal, $\tilde{Y}$ est l'unique vecteur de $\mathcal{M}(X)$ minimisant la distance euclidienne $\|Y - \tilde{Y}\|$, ce qui revient au même que de minimiser son carré. Or, par définition de la norme euclidienne, cette quantité vaut :

$$\|Y - \tilde{Y}\|^2 = \sum_{i=1}^{n}(y_i - (\tilde{\beta}_1 + \tilde{\beta}_2 x_i))^2$$

ce qui nous raméne à la méthode des moindres carrés ordinaires. On en déduit que :
$\tilde{\beta}_1 = \hat{\beta}_1, \tilde{\beta}_2 = \hat{\beta}_2$ et $\tilde{Y} = \hat{Y} = [\hat{y_1}, ..., \hat{y_n}]'$, avec les expressions de $\hat{\beta}_2$,et $\hat{\beta}_2$. et $\hat{Y}$ vues précédemment Y.

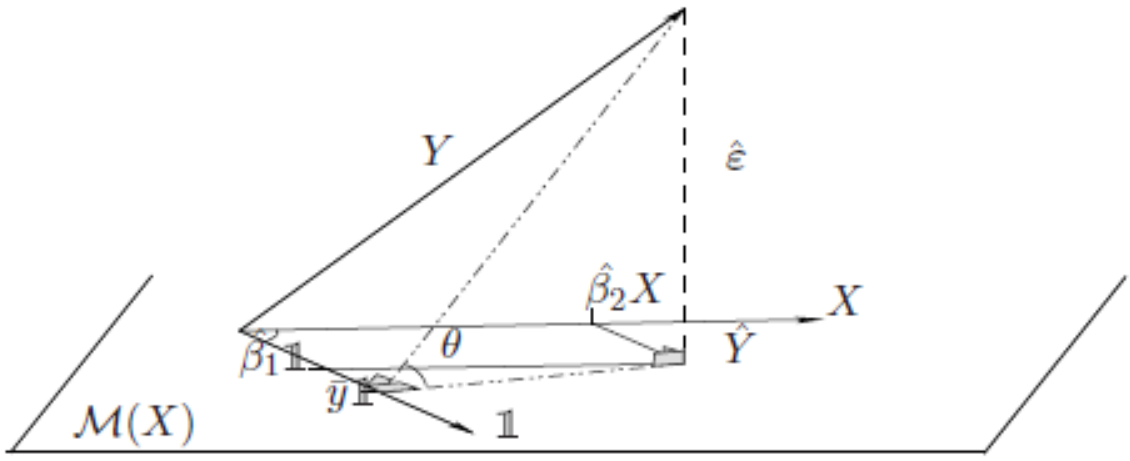

FIGURE 2.4 – Représentation de la projection dans l'espace des variables

Autrement dit,dans $\mathbb{R}^n$, $\hat{\beta}_1$ et $\hat{\beta}_2$ s'interprétent comme les coordonnées de la projection orthogonale. $\hat{Y}$ de Y sur le sous-espace de $\mathbb{R}^n$ engendré par 1 et X (voir figure 2.3).
Remarque :

1. Cette vision géometrique des choses peut sembler un peu abstraite, mais c'est en fait l'approche féconde pour comprendre la régression multiple.

2. Nous avons supposé que 1 et X ne sont pas colinéaires. En général, ces vecteurs ne sont pas orthogonaux (sauf si $\bar{x} = 0$), ce qui implique que $\hat{\beta}_1 1$ n'est pas la projection orthogonale de Y sur 1(laquelle vaut $\bar{y}1$), et que $\hat{\beta}_2 X$ n'est pas la projection orthogonale de Y sur X (laquelle vaut $\frac{<Y,X>}{\|X\|^2} X$).

Le coefficient de détermination R^2

Nous conservons les notations du paragraphe précédent, avec $\hat{Y} = [\hat{y}_1, ..., \hat{y}_n]'$ la projection orthogonale du vecteur Y sur $\mathcal{M}(X)$ est

$$\hat{\epsilon} = Y - \hat{Y} = [\hat{\epsilon}_1, .., \hat{\epsilon}_n]'$$

le vecteur des résidus déjà rencontrés en section (calculs de résidus). Le théorème de Pythagore donne alors directement :

$$\|Y - \bar{y}1\|^2 = \|\hat{Y} - \bar{y}1\|^2 + \|\hat{\epsilon}\|^2$$

$$\sum (y_i - \bar{y})^2 = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n} \hat{\epsilon}_i^2$$

$$SCT = SCE + SCR,$$

où SCT (respectivement SCE et SCR) représente la somme des carrés totale (respectivement expliquée par le modèle et résiduelle). Ceci peut se voir comme une formule typique de décomposition de la variance. Elle permet en outre d'introduire le coefficient de détermination de façon naturelle.

Définition :(Coefficient de détermination R^2) :

Le coefficient de détermination R^2 est défini par :

$$R^2 = \frac{SCE}{SCT} = \frac{\|\hat{Y} - \bar{y}1\|^2}{\|Y - \bar{y}1\|^2} = 1 - \frac{\|\hat{\epsilon}\|^2}{Y - \bar{y}1} = 1 - \frac{SCR}{SRT}$$

On voit sur la figure 2.3 que R^2 correspond au cosinus carré de l'angle θ. De façon schématique, on peut différencier les cas suivants :
-Si $R^2 = 1$, le modele explique tout, l'angle θ vaut zéro et Y est dans $\mathcal{M}(X)$, c'est-à-dire que $y_i = \beta_1 + \beta_2 x_i$ pour tout i : les points de l'échantillon sont parfaitement alignés sur la droite des moindres carrés
-Si $R^2 = 0$, cela veut dire que $\sum(\hat{y}_i - \bar{y})^2 = 0$, donc $\hat{y}_i = \bar{y}$ pour tout i. Le modele de regréssion linéaire est inadapte puisqu'on ne modélise rien de mieux que la moyenne
-Si R^2 est proche de zéro, cela veut dire que Y est quasiment dans l'orthogonal de $\mathcal{M}(X)$, le modele de régression linéaire est inadapte, la variable x n'explique pas bien la variable reponse y(du moins pas de façon affine)
De façon générale, l'interprétation est la suivante :le modéle de regréssion linéaire permet d'expliquer $100 \times R^2$ de la variance totale des données.

2.2 *La régression linéaire multiple*

Introduction[4]
La modélisation de la concentration d'ozone dans l'atmosphère évoquée au section1(la régression linéaire simple) est relativement simpliste. En effet, d'autres variables peuvent expliquer cette concentration, par exemple le vent qui pousse les masses d'air. Ce phénomène physique est connu sous le nom d'advectance (apport d'ozone) ou de dilution. D'autres variables telles le rayonnement, la précipitation, etc., ont une influence certaine sur la concentration d'ozone. L'association Air Breizh mesure ainsi en même temps que la concentration d'ozone d'autres variables susceptibles d'avoir une influence sur celle-ci. Voici quelques-unes de ces données :

T_{12}	23.8	16.3	27.2	7.1	25.1	27.5	19.4	19.8	32.2	20.7
V	9.25	-6.15	-4.92	11.57	-6.23	2.76	10.15	13.5	21.27	13.79
N_{12}	5	7	6	5	2	7	4	6	1	4
O_3	115.4	76.8	113.8	81.6	115.4	125	83.6	75.2	136.8	102.8

FIGURE 2.5 – 10 données journalières de température, vent, nébulosité et ozone

La variable V est une variable synthétique. En effet, le vent est normalement mesuré en degrés (direction) et mètres par seconde (vitesse). La variable V que nous avons créée est la projection du vent sur l'axe Est-Ouest, elle tient donc compte à la fois de la direction et de la vitesse. Pour analyser la relation entre la température T, le vent V, la nébulosité N à midi et l'ozone O_3, nous allons chercher une fonction f telle que :

$$O_{3i} \approx f(T_i, V_i, N_i)$$

Afin de préciser $\approx$ il va falloir definir comme au section 1(régression linéaire simple) un critére quantifiant la qualité de l'ajustement de la fonction f aux données, ou inversement le coût de non-ajustement. Cette notion de coût permet d'apprehender de maniére aisée les problémes d'ajustement économique dans certains modéles, d'ou son nom.
Minimiser un cout nécessite aussi la connaissance de l'espace sur lequel on minimise, c'est-à-dire la classe de fonctions F dans laquelle nous supposerons que se trouve la vraie fonction inconnue. Le probléme mathématique peut s'écrire de la façon suivante :

$$\arg\min_{f \in F} \sum L(y_i - f(x_i)) \tag{2.5}$$

où n représente le nombre de données à analyser,$L(.)$ est appelée fonction de coût, ou de perte, et x_i est une variable vectorielle pour tout i. La fonction de coût sera la même que celle utilisée précédemment, c'est-à-dire le coût quadratique. En ce qui concerne le choix de la classe F, par analogie avec le chapitre précédent, nous utiliserons la classe suivante :

$$F = \left\{ \ f : \mathbb{R}^P \longrightarrow \mathbb{R}, f(x_1, .., x_p) = \sum_{j=1}^{p} \beta_j x_j, \quad . \right.$$

En général, avec cette convention d'écriture, x_1 est constant égal à 1 et β_1 correspond à l'ordonnée d'origine. On parle de regréssion linéaire en raison de la linearité de f en les paramétres $\beta_1, ..., \beta_p$ non en les variables explicatives x_j. Par exemple, ce modéle inclut les fonctions polynomiales d'une seule variable x si l'on prend $x_1 = 1, x_2 = x, ..., x_p = x^{p-1}$
Cette section est donc la généralisation naturelle du précedente, mais nous allons cette fois manipuler sytematiquement des vecteurs et des matrices à la place des scalaires.

2.2.1 Modélisation

Le modele de regréssion linéaire multiple est une généralisation du modéle de régression simple lorsque les variables explicatives sont en nombre quelconque. Nous supposons donc que les données collectées suivent le modéle suivant :

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon_i, i = 1, .., n \tag{2.6}$$

où :

- les x_{ij} sont des nombres connus, non aléatoires, la variable x_{i1} valant souvent 1 pour tout i
- les paramétres β_j du modéle sont inconnus, mais non aléatoires
- les ϵ_i sont des variables aléatoires inconnues.

Remarque :

Du fait que la constante appartient généralement au modèle, beaucoup d'auteurs écrivent plutût le modèle sous la forme :

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon_i, i = 1, .., n$$

de sorte que p corréspond toujours au nombre de variables explicatives. Avec notre convention d'écriture (2.6), si x_{i1} vaut 1 pour tout i, p est le nombre de paramètres à estimer, tandis que le nombre de variables explicatives est, à proprement parler, $(p-1)$.
En utilisant l'écriture matricielle de(2.6)nous obtenons la définition suivante :
Définition(Modéle de régression linéaire multiple). Un modèle de régression linéaire multiple est défini par une équation de la forme :

$$Y = X\beta + \epsilon$$

où :
.Y est un vecteur aléatoire de dimension n
.X est une matrice de taille $n \times p$ connue, appelée matrice du plan d'éxperience
.β est le vecteur de dimension p des paramétres inconnus du modéle
.ϵ est le vecteur de dimension n des erreurs
Les hypothèses concernant le modèle sont :

$$(\mathcal{H}) \begin{cases} (\mathcal{H}_1) = rg(X) = p, \\ (\mathcal{H}_2) = E[\epsilon] = 0, Var(\epsilon) = \sigma^2 I_n, \end{cases}$$

L'hypothèse $(\mathcal{H}_2)$ signifie que les erreurs sont centrées, de même variance(homoscédasticité)et non corrélées entre elles.
Notation. On notera $X = [X_1 \mid .... \mid X_p]$, où X_j est le vecteur de taille n correspondant à la j-éme variable. La i-éme ligne de la matrice X sera quant à elle notée $x_i' = [x_{i1}, .., x_{ip}]$. Ainsi

l'équation (2.2) s'ecrit aussi :

$$\forall i \in 1, ..., n y_i = x_i' \beta - \epsilon_i$$

2.2.2 Estimateurs des Moindres Carrés Ordinaires

Comme pour la régression linéaire simple, nous allons considérer ici une fonction de coût quadratique, d'où la dénomination de Moindres Carrés Ordinaires (MCO).
Définition :(Estimateur des MCO) L'estimateur des moindres carrés.$\hat{\beta}$ est défini comme suit :

$$\hat{\beta} = arg \min_{\beta \in \mathbb{R}^P} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 = arg \min_{\beta \in \mathbb{R}^P} \|Y - X\beta\|^2 \tag{2.7}$$

Dans la suite de cette section, nous allons donner l'expression de l'estimateur.$\hat{\beta}$ ainsi que certaines de ses propriétés.

Calcul de $\hat{\beta}$

Pour determiner $\hat{\beta}$, une méthode consiste à se placer dans l'espace des variables, comme on l'a fait au section(régression simple), Section(representation des variables) Rappelons briévement le principe : $Y = [y_1, ..., y_n]'$ est le vecteur des variables à expliquer. La matrice du plan d'experience $X = [X_1 \mid ... \mid X_p]$ est formée de p vecteurs colonnes (la premiére colonne étant généralement constituée de 1). Le sous-espace de $\mathbb{R}^n$ engendre par les p vecteurs colonnes de X est appelé espace image, ou espace des solutions, et noté $\mathcal{M}(X)$. Il est de dimension p par l'hypothése $(\mathcal{H}_1)$ et tout vecteur de cet espace est de la forme $X\alpha$, où α est un vecteur de $\mathbb{R}^p$:

$$X\alpha = \alpha_1 X_1 + ..., + \alpha_p X_p$$

le vecteur Y est la somme d'un élément de $\mathcal{M}(X)$ et d'un bruit élément de $\mathbb{R}^n$, lequel n'a aucune raison d'appartenir à $\mathcal{M}(X)$. Minimiser $\|Y - X\alpha\|^2$ revient à chercher un élément de $\mathcal{M}(X)$ qui soit le plus proche de Y au sens de la norme euclidienne classique. Cet unique elémént est, par définition, le projeté orthogonal de Y sur $\mathcal{M}(X)$. Il sera noté $\hat{Y} = P_X Y$,où P_X est la matrice de projection orthogonale sur $\mathcal{M}(X)$. Il peut aussi s'ecrire sous la forme $\hat{Y} = X\hat{\beta}$, où $\hat{\beta}$ est l'estimateur des MCO de β. L'espace orthogonal à $\mathcal{M}(X)$, noté $M \perp (X)$, est souvent appelé espace des résidus. En tant que supplementaire orthogonal, il est de dimension $n - p = dim(\mathbb{R}^n) - dim(\mathcal{M}(X))$.

Proposition 2.2.1. *(Expréssion de $\hat{\beta}$) L'estimateur $\hat{\beta}$ des Moindres Carrés Ordinaires à pour expression :*

$$\hat{\beta} = (X'X)^{-1} X'Y.$$

et la matrice :

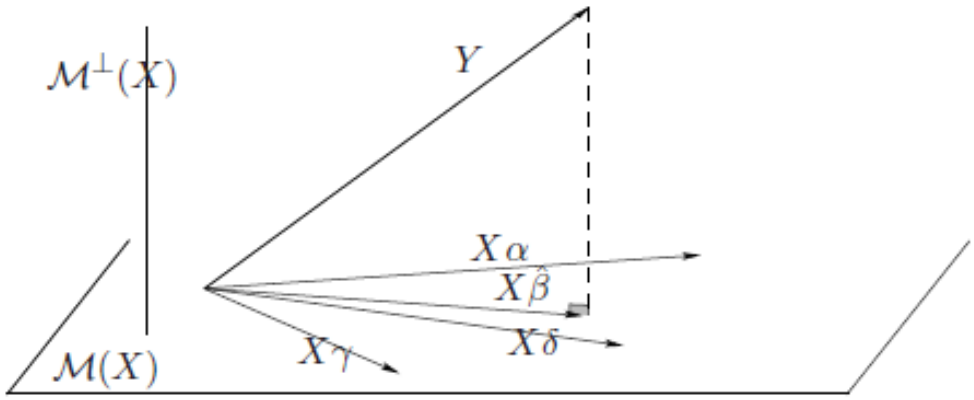

FIGURE 2.6 – Réprésentation de $X\hat{\beta}$ dans l'espace des variables

P_X de projection orthogonale sur $\mathcal{M}(X)$ s'écrit :

$$P_X = X(X'X)^{-1}X'.$$

Remarque :

L'hypothése $(\mathcal{H}_1)$ assure que la matrice $X'X$ est bien inversible. Supposons en effet qu'il existe un vecteur β de $\mathbb{R}^p$ tel que $(X'X)\beta = 0$. Ceci impliquerait que $\|X\beta\|^2 = \beta'(X'X)\beta = 0$, donc $X\beta = 0$, d'où $\beta = 0$, puisque $rg(X) = p$. Autrement dit, la matrice symétrique $X'X$ est définie positive.

Preuve :

On peut prouver ce résultat de plusieurs façons :

1. Par différentiation : on cherche $\beta \in \mathbb{R}^p$ qui minimise la fonction

$$S(\beta) = \|Y - X\beta\|^2 = \beta'(X'X)\beta - 2Y'X\beta + \|Y\|^2$$

Or S est de type quadratique en β, avec $X'X$ symétrique définie positive, donc le probléme admet une unique solution $\hat{\beta}$: c'est le point où le gradient de S est nul.

$$\nabla S(\hat{\beta}) = 2\hat{\beta}'X'X - 2Y'X = 0 \Longleftrightarrow (X'X)\hat{\beta} = X'Y$$

La matrice $X'X$ étant inversible par (H_1), ceci donne $\hat{\beta} = (X'X)^{-1}X'Y$. Puisque par définition. $\hat{Y} = P_XY = X\beta = X(X'X)^{-1}.X'Y$ et que cette relation est valable pour tout Y. $\mathbb{R}^n$, on en déduit que $P_X = X(X'X)^{-1}X'$.

2. Par projéction : une autre façon de proceder consiste à dire que le projeté orthogonal. $\hat{Y} = X\hat{\beta}$ est défini comme l'unique vecteur tel que $(Y - \hat{Y})$ soit orthogonal a $\mathcal{M}(X)$. Puisque $\mathcal{M}(X)$ est engendré par les vecteurs $X_1, ..., X_p$, ceci revient à dire que $(Y - \hat{Y})$ est orthogonal à chacun des X_i :

$$\begin{cases} X_1, Y - X\hat{\beta} = 0, \\ . \\ . \\ . \\ X_P, Y - X\hat{\beta} = 0, \end{cases}$$

Ces p équations se regroupent en une seule :$X'(Y - X\hat{\beta}) = 0$, d'où l'on déduit bien l'expression de $\hat{\beta}$, puis celle de P_X
Dorénavant nous noterons $P_X = X(X'X)^{-1}X'$ la matrice de projéction orthogonale sur $\mathcal{M}(X)$ et $P_{X^\perp} = (I - P_X)$ la matrice de projection orthogonale sur $M\perp(X)$. La décomposition

$$Y = \hat{Y} + (Y - \hat{Y}) = P_X Y + (I - P_X)Y = P_X Y + P_{X^\perp}$$

n'est donc rien de plus qu'une décomposition orthogonale de Y sur $\mathcal{M}(X)$ et $\mathcal{M}\perp(X)$

$$\hat{Y} = X\hat{\beta} = \hat{\beta}_1 X_1 + ... + \hat{\beta}_p X_p$$

signifie que les $\hat{\beta}_i$ sont les coordonnées de $\hat{Y}$ dans la base $(X_1, ..., X_p)$ de $\mathcal{M}(X)$. Il ne faudrait pas croire pour autant que les $\hat{\beta}_i$ sont les coordonnées des projections de Y sur les X_i : ceci n'est vrai que si la base $(X_1, ..., X_p)$ est orthogonale, ce qui n'est pas le cas en général.
Rappels sur les projecteurs. Soit P une matrice carrée de taille n. On dit que P est une matrice de projéction si $P^2 = P$. Ce nom est du au fait que pour tout vecteur x de $\mathbb{R}^n$, P_x est la projéction de x sur Im(P) parallèlement à $Ker(P)$. Si en plus de verifier $P^2 = P$, la matrice P est symétrique, alors P_x est la projéction orthogonale de x sur $Im(P)$ parallélement à $Ker(P)$, c'est-à-dire que dans la décomposition $x = P_x + (x - P_x)$, les vecteurs P_x et $(x - P_x)$ sont orthogonaux. C'est ce cas de figure qui nous concernera dans ce chapitre . Toute matrice symétrique réelle étant diagonalisable en base orthonormée, il existe une matrice orthogonale U(i.e $UU' = I_n$), ce qui signifie que les colonnes de U forment une base orthonormée de $\mathbb{R}^n$ et une matrice diagonale ∇ telles que $P = U\nabla U'$. On voit alors facilement que la diagonale de ∇ est composée de p"1"Â¡ et de $(n - p)$, "0", où p est la dimension de $Im(P)$, espace sur lequel on projette.
Revenons a nos moutons : on a vu que $P_X = X(X'X)^{-1}X'$. On vérifie bien que $P_X^2 = P_X$ et que P_X est symétrique. Ce qui précéde assure également que $Tr(P_X) = p$ et $Tr(P_{X\perp}) = n - p$. Cette derniére remarque nous sera utile pour construire un estimateur sans biais de σ^2. D'autre part, la matrice P_X est souvent notée H(comme Hat) dans la littérature anglo-saxonne, car elle met des chapeaux sur les vecteurs :$P_X Y = \hat{Y}$. De fait, les éléments de P_X sont notés $(h_{ij})_{1 \leq i,j \leq n}$.

Quelques propriétés

Comme en régression simple, l'estimateur obtenu est sans biais. On obtient de plus une expression trés simple pour sa matrice de covariance $Var(\hat{\beta})$. On rappelle que la matrice de covariance du vecteur aléatoire $\hat{\beta}$, ou matrice de variance-covariance, ou matrice de dispersion, est par définition :

$$Var(\hat{\beta}) = E[(\hat{\beta} - E[\hat{\beta}])(\hat{\beta} - E(\hat{\beta})'] = E[\hat{\beta}\hat{\beta}'] - E[\hat{\beta}]E[\hat{\beta}']$$

Puisque β est de dimension p, elle est de dimension $p \times p$. De plus, pour pour toute matrice A de taille $m \times p$ et tout vecteur B de dimension m déterministes, on a :
$E[A\hat{\beta} + B] = AE[\hat{\beta}] + B$ et $Var(A\hat{\beta} + B) = AVar(\hat{\beta})A'$. Ces propriétés élémentaires seront constamment appliquées dans la suite.

Proposition 2.2.2. *(Biais et matrice de covariance) L'estimateur $\hat{\beta}$ des moindres carrés est sans biais, i.e.$E[\hat{\beta}] = \beta$ et sa matrice de covariance est :*

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$$

Preuve :Pour le biais il suffit d'écrire :

$$E[\hat{\beta}] = E[(X'X)^{-1}X'Y] = (X'X)^{-1}X'E[Y] = (X'X)^{-1}X'E[X\beta + \epsilon]$$

et puisque $E[\epsilon] = 0$, il vient :

$$E[\hat{\beta}] = (X'X)^{-1}X'X\beta = \beta$$

Pour la variance, on procède de même :

$$Var(\hat{\beta}) = Var(X'X)^{-1}X'Y) = (X'X)^{-1}X'Var(Y)X(X'X)^{-1}$$

or $Var(Y) = Var(X\beta + \epsilon) = Var(\epsilon) = \sigma^2 I_n$,donc :

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1}X'X(X'X)^{-1} = \sigma^2(X'X)^{-1}$$

L'estimateur des MCO est optimal en un certain sens. C'est ce que précise le résultat suivant, généralisation de celui vu en régression linéaire simple.

Théorème 2.2.1. *(Gauss- Markov) L'estimateur $\hat{\beta}$ des MCO est de variance minimale parmi les estimateurs linéaires sans biais de β*

Remarque :

1. Linéaire signifie "linéaire par rapport à Y", c'est-à-dire de la forme AY où A est une matrice (p,n) : en ce sens, l'estimateur $\hat{\beta}$ des MCO est bien linéaire puisque $\hat{\beta} = (X'X)^{-1}X'Y$.
2. Rappelons qu'il existe une relation d'ordre partielle entre matrices symetriques réelles : dire que $S_1 \leq S_2$ signifie que $S = (S_2 - S_1)$ est une matrice symétrique réelle positive, c'est-à-dire que pour tout vecteur x, on a $x'S_1x \leq x'S_2x$. Ceci revient encore à dire que les valeurs propres de S sont toutes supérieures ou egales à 0.

Nous allons montrer que, pour tout autre estimateur $\tilde{\beta}$ de β linéaire et sans biais,$Var(\tilde{\beta}) \geq Var(\hat{\beta})$, où l'inégalité entre matrices de variance-covariance est à comprendre au sens precise ci dessus. Rappelons la formule générale pour la matrice de covariance de la somme deux vecteurs aléatoires U et V :
$Var(U+V) = Var(U) + Var(V) + Cov(U,V) + Cov(V,U)$, Où
Cov(U,V)=E[UV']-E[U]E[V]'=Cov(V,U)'. Décomposons ainsi la variance de $\tilde{\beta}$:
$Var(\tilde{\beta}) = Var(\tilde{\beta} - \hat{\beta} + \hat{\beta}) = Var(\tilde{\beta} - \hat{\beta}) + Var(\hat{\beta}) + Cov(\hat{\beta}, \tilde{\beta} - \hat{\beta})$
Les variances étant semi-définies positives, si nous montrons que $Cov(\tilde{\beta} - \hat{\beta}, \hat{\beta}) = 0$, nous aurons fini la démonstration. Puisque $\tilde{\beta}$ est linéaire, $\tilde{\beta} = AY$. De plus, nous savons qu'il est sans biais, c'est-à-dire $E[\tilde{\beta}] = \beta$ pour tout β, donc $AX = I$.La covariance devient :
$Cov(\tilde{\beta} - \hat{\beta}, \hat{\beta}) = Cov(AY, (X'X)^{-1}X'Y) - Var(\hat{\beta}) = \sigma^2 AX(X'X)^{-1} - \sigma^2(X'X)^{-1} = 0$

Résidus et variance résiduelle

Les résidus sont définis par :

$$\hat{\epsilon} = [\hat{\epsilon}_1, ..., \hat{\epsilon}_n]' = Y - \hat{Y} = (I - P_X)Y = P_{X^\perp} = P_{X^\perp}\epsilon$$

car $Y = X\beta + \epsilon$ et $X\beta \in$ ¸$\mathcal{M}(X)$. On peut alors énoncer les résultats suivants.
propriété :(Biais et variance de ϵ et $\hat{Y}$). Sous le jeu d'hypothèses ($\mathcal{H}$), on a :

1. $E[\hat{\epsilon}] = 0$
2. $Var(\hat{\epsilon}) = \sigma^2 P_{X^\perp}$
3. $E[\hat{Y}] = X\beta$
4. $Var(\hat{Y}) = \sigma^2 P_X$
5. $Cov(\hat{\epsilon}, \hat{Y}) = 0$

Preuve :

1. $E[\hat{\epsilon}] = E[P_{X^\perp}\epsilon] = P_{X^\perp}E[\epsilon] = 0$
2. $Var(hat\epsilon) = P_{X^\perp}Var(\epsilon)P'_{X^\perp} = P_{X^\perp}Var(\epsilon)P_{X^\perp} = \sigma^2 P_{X^\perp}$
3. $E[\hat{Y}] = E[X\hat{\beta}] = X\beta, car\hat{\beta}$ est sans biais
4. $Var(\hat{Y}) = E[X\hat{\beta}] = XVar(\hat{\beta})X' = \sigma^2 X(X'X)^{-1}X' = \sigma^2 P_X$

5. Rappelons que la covariance entre deux vecteurs aléatoires est une application bilinéaire et que Cov(U,U)=Var(U).ci, ceci donne :

$$Cov(\hat{\epsilon}, \hat{Y}) = Cov(\hat{\epsilon}, Y - \hat{\epsilon}) = Cov(\hat{\epsilon}, Y) - Var(\hat{\epsilon}) = Cov(P_{X^\perp} Y, Y) - \sigma^2 P_{X^\perp}$$

et puisque $Var(Y) = \sigma^2 I_n$, nous avons :

$$Cov(\hat{\epsilon}, \hat{Y}) = P_{X^\perp} Var(Y) - \sigma^2 P_{X^\perp} = 0$$

Comme en régression linéaire simple, un estimateur naturel de la variance résiduelle est donné par :

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i}) = \frac{1}{n}\sum_{i=1}^{n}\hat{\sigma_i^2} = \frac{1}{n}\|\hat{\epsilon}\|^2$$

Malheureusement on va voir que cet estimateur est biaisé. Ce biais est néanmoins facile à corriger, comme le montre le résultat suivant.
C'est une bête généralisation du résultat obtenu en régression linéaire simple, en remplaçant n-2 par n-p.

Proposition 2.2.3. *la statistique* $\hat{\sigma}^2 = \frac{\|\hat{\epsilon}\|^2}{n-p} = \frac{SCR}{n-p}$ *est un estimateur sans biais de* σ^2

Preuve :Nous calculons $E[\|\hat{\epsilon}\|^2]$. Ruse de sioux :puisque c'est un scalaire, il estégal à sa trace, ce qui donne :

$$E[\|\hat{\epsilon}\|^2] = E[Tr(\|\hat{\epsilon}\|)^2] = E[Tr(\hat{\epsilon}'\hat{\epsilon})]$$

,
et puisque pour toute matrice A, on a $Tr(AA') = Tr(A'A) = \sum_{i,j}\sigma_{i,j}^2$, il vient :

$$E[\|\hat{\epsilon^2}\|] = E[Tr(\hat{\epsilon}\hat{\epsilon'})] = Tr(Var(\hat{\epsilon})) = Tr(\sigma^2 P_{X^\perp})$$

Et comme $P_{X^\perp}$ est la matrice de la projection orthogonale sur un espace de dimension (n-p), on a bien : $E[\|\hat{\epsilon^2}\|] = (n-p)\sigma^2$
On déduit de cet estimateur de $\hat{\sigma^2}$ de la variance résiduelle σ^2 un estimateur $\hat{\sigma^2_{\beta}}$ de la variance $Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$:

$$\hat{\sigma^2_{\beta}} = \hat{\sigma}^2(X'X)^{-1} = \frac{\|\hat{\epsilon^2}\|}{n-p}(X'X)^{-1} = \frac{SCR}{n-p}(X'X)^{-1}$$

En particulier, un estimateur de l'écart-type de l'estimateur $\hat{\beta}_j$ du j-éme coefficient de la

régression est tout simplement :

$$\hat{\sigma}_{\hat{\beta}_j} = \hat{\sigma}\sqrt{[(X'X)^{-1}]_{ij}}$$

Afin d'alleger les notations, on écrira parfois $\hat{\sigma}_j$ pour $\hat{\sigma}_{\hat{\beta}_j}$

Prévision

Un des buts de la régression est de proposer des prédictions pour la variable à expliquer y lorsque nous avons de nouvelles valeurs de x. Soit donc $x'_{n+1} = [x_{n+1,1}, ..., x_{n+1,p}]$ une nouvelle valeur pour laquelle nous voudrions prédire y_{n+1}. Cette variable réponse est définie par $y_{n+1} = x'_{n+1}\beta + \epsilon_{n+1}$, avec $E[\epsilon_{n+1}] = 0$,$Var(\epsilon_{n+1}) = \sigma^2$ et $Cov(\epsilon_{n+1}, \epsilon_i) = 0$ pour i=1,..., n. La méthode naturelle est de prédire la valeur corréspondante graçe au modéle ajusté, soit : $\hat{y}_{n+1} = x'_{n+1}\beta$. L'erreur de prévision est à nouveau définie par :

$$\hat{\epsilon}_{n+1} = y_{n+1} - \hat{y}_{n+1} = x'_{n+1}(\beta - \hat{\beta}] + \epsilon_{n+1}$$

. Deux types d'erreurs vont alors entacher notre prévision : la première due à l'incertitude sur ϵ_{n+1} l'autre à l'incertitude inhérente à L'estimateur $\hat{\beta}$

Proposition 2.2.4. *(Erreur de prévision) l'erreur de prévision* $\hat{\epsilon}_{n+1} = (y_{n+1} - \hat{y}_{n+1})$ *satisfait les propriétés suivantes :*

$$\begin{cases} E[\hat{\epsilon}_{n+1}] = 0, \\ Var(\hat{\epsilon}_{n+1}) = \sigma^2(1 + x'_{n+1}(X'X)^{-1}x_{n+1}), \end{cases}$$

Preuve : Comme $E[\epsilon_{n+1}] = 0$ et puisque $\hat{\beta}$ est un estimateur sans biais de β, il est clair que :

$$E[\hat{\epsilon}_{n+1}] = Var(\epsilon_{n+1} + x'_{n+1}(\beta - \hat{\beta})) = \sigma^2 + x'_{n+1}Var(\hat{\beta})x_{n+1} = \sigma^2(1 + x'_{n+1}(X'X)^{-1}x_{n+1})$$

Nous retrouvons bien l'incertitude d'observation σ^2 à laquelle vient s'ajouter l'incertitude d'estimation. Enfin, comme en régression linéaire simple, on peut prouver qu'en présence de la constante, cette incertitude est minimale au centre de gravité des variables explicatives, c'est-à-dire lorsque

$x'_{n+1} = [1, \bar{x}_2, ..., \bar{x}_p]$ et qu'elle vaut encore $\sigma^2(1 + \frac{1}{n})$

2.2.3 Interprétation géométrique

À partir de cette figure, le théorème de Pythagore donne :

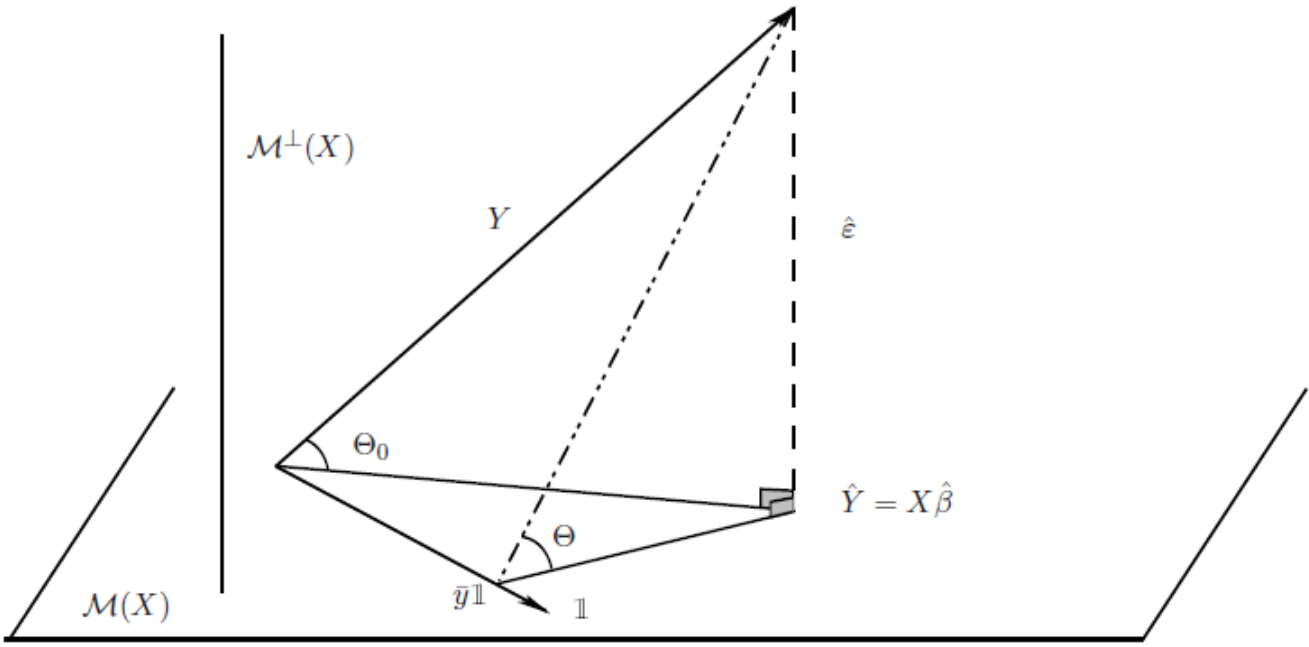

FIGURE 2.7 – Réprésentation des variables

SCT=SCE+SCR

$$\|Y\|^2 = \|\hat{Y}\|^2 + \|\hat{\epsilon}\|^2 = \|X\hat{\beta}\|^2 + \|Y - X\hat{\beta}\|^2$$

Si la constante fait partie du modèle (ce qui est généralement le cas), alors nous avons, toujours par Pythagore :

SCT=SCE+SCR

$\|Y - \bar{y}1\|^2 = \|\hat{Y} - \bar{y}1\|^2 + \|\hat{\epsilon}\|^2$.

Variation totale=V.expliquée par le modèle +V résiduelle.

Définition :
Le coéfficient de détérmination R^2 est défini par :

$R^2 = \cos^2\theta_0 = \frac{\|\hat{Y}\|^2}{\|Y\|^2} = 1 - \frac{\|\hat{\epsilon}\|^2}{\|Y\|^2} = 1 - \frac{SCR}{SCT}$

ou plus souvent, si la constante fait partie du modèle, par :

$$R^2 = \cos^2\theta = \frac{\|\hat{Y} - \bar{y}1\|^2}{\|Y - \bar{y}1\|^2} = 1 - \frac{\|\hat{\epsilon}\|^2}{\|Y - \bar{y}1\|^2} = 1 - \frac{SCR}{SCT}$$

.

Ce coéfficient mesure le cosinus carré de l'angle entre les vecteurs Y et $\hat{Y}$ pris à l'origine ou pris en $\bar{y}1$. Néanmoins, on peut lui reprocher de ne pas tenir compte de la dimension de l'espace de projection $\mathcal{M}(X)$, d'où la définition du coéfficient de détermination ajusté.

Définition : Le coéfficient de détermination ajusté R_a^2 est défini par :

$$R_a^2 = 1 - \frac{n}{n-p}.\frac{\|\hat{\epsilon}\|^2}{\|Y\|^2} = 1 - \frac{n}{n-p}(1 - R^2)$$

ou plus souvent, si la constante fait partie du modèle, par :

$$R_a^2 = 1 - \frac{n-1}{n-p}.\frac{\|\hat{\epsilon}\|^2}{\|Y - \bar{y}1\|^2} = 1 - \frac{n-1}{n-p}.\frac{SCR}{SCT} = 1 - \frac{n-1}{n-p}(1 - R^2)$$

Avec le logiciel R, le coéfficient de détermination R^2 est appelé Multiple R-Squared, tandis que le coéfficient de détermination ajusté R_a^2 est appelé Adjusted R-Squared.

conclusion :Ce chapitre introduit la notion de modèle de régresssion linéaire simple et multiple par la version la plus élémentaire :expliquer Y par une fonction affine de X. Après avoir expliciter les hypothèses nécessaires et les termes du modèle, les notions d'estimation des paramètres du modèle, de prévision par intervalle de confiance, la signification des tests d'hypothèse sont discutées.

Chapitre 3

ÉTUDE COMPARATIVE DES MÉTHODES D'OPTIMISATION SANS CONTRAINTE

3.1 gradient conjugué

[3] En premier, nous allons considérer la fonction $\min f(x)$ quadratique avec :
$f(x) = \frac{1}{2}(Ax, x) - (b, x)$ où A est symétrique et définie-positive et $f(\bar{x}) = \inf f \Longleftrightarrow A\bar{x} = b$.
C'est-à-dire, nous allons traiter de la résolution itérative du système linéaire $Ax = b$.

Directions conjugées : Un ensemble de vecteurs $\{w^{(0)}; w^{(1)}; .....; w^{(n-1)}\}$ est dit A-conjuguée si :

$$(Aw_i; w_j) \neq 0 \forall i \neq j.$$

Autrement dit, les $(w^(i))$ sont perpendiculaires entre eux par rapport au produit scalaire induit par la matrice $A :< U, V >= (AU, V)$.
Le but de l'algorithme du gradient conjugué est de construire deux suites de vecteurs : les itérés $\{x^{(0)}; x^{(1)}; x^{(2)}; ....x^{(k)}\}$; et les directions de descentes $\{w^{(0)}; w^{(1)}; .....; w^{(n-1}\}$ qui vérifient les propriétés suivantes. On note le vecteur résidu :$r^{(k)} = b - Ax^{(k)}$.
la suite des résidus $\{r^{(0)}; r^{(1)}; .....; r^{(k)}\}$ forme un système orthogonal (au sens usuel).
la suite des directions de descentes $\{w^{(0)}; w^{(1)}; .....; w^{(n-1)}\}$ forme un système A-conjuguées.
Conséquence : l'algorithme du gradient conjugué converge en au plus n itérations, i.e. fournit la solution exacte de $Ax = b$.
L'objectif est de construire les itérés $x^{(k)}$ et les directions conjuguées $w^{(k)}$.
Voici les étapes de l'algorithme du gradient conjugué :

- Mise à jour de $x^{(k)}$: $x^{(k+1)} = x^{(k)} + \alpha_k w^{(k)}$
- Mise à jour du résidu $r^{(k)}$: $r^{(k+1)} = r^{(k)}_{\alpha_k} A w^{(k)}$
- Mise à jour des directions conjuguées $w^{(k)}$: $w^{(k+1)} = r^{(k+1)} + \lambda_k w^{(k)}$

L'algorithme démarre avec le choix $w^{(0)} = r^{(0)} = b - Ax^{(0)}$ pour un certain point initial $x^{(0)}$ (par exemple $x^{(0)} = 0$).

Calcul des coéfficients α_k :
Il suffit d'exiger que $r^{(k+1)} \perp r_{(k)}$.

$0 = (r_{(k+1)}, r^{(k)}) = (r^{(k)} - \alpha_k A w^{(k)}, r^{(k)}) \Longrightarrow \alpha_k = \frac{(r^k, r^k)}{Aw^{(k)}, r^{(k)}}$.

Or :$r^{(k)} = w^{(k)} - \beta_{k-1} w^{(k-1)} \Longrightarrow (Aw^{(k)}, w^{(k)}) = (Aw^{(k)}, w^{(k)} - \beta_{k-1} w^{(k-1)}) = (Aw^{(k)}, w^{(k)})$.

car les $w^{(k)}$ sont A conjugués. Ceci fournit la valeur suivante :$\alpha_k = \frac{\|r^{(k)}\|^2}{Aw^{(k)}, w^{(k)}}$.

Calcul des coéfficient β_k :

$0 = (w^{(k+1)}; w^{(k)}) = (r^{(k+1)} + \beta_{(k)} w^{(k)} A w^{(k)} = -\frac{(r^{(k+1)}; Aw^{(k)}}{(Aw^{(k)}; w^{(k)}}$.

Mais $r^{(k+1)} = r^{(k)} - \alpha_k A w^{(k)} \longrightarrow -Aw^{(k)} = \frac{r^{(k+1)} - r^{(k)}}{\alpha_k}$. Ceci fournit la valeur :

$= \frac{r^{(k+1)}, r^{(k+1)}}{\alpha_k (Aw^{(k)}, w^{(k)})}$.

car $(r^{(k+1)}; r^{(k)}) = 0$. De plus, $\alpha_k = \frac{\|r^{(k)}\|^2}{Aw^{(k)}, w^{(k)}}$.

Ceci fournit la valeur finale :$\beta_k = \frac{\|r^{(k+1)}\|^2}{r^{(k)}}^2$.

Exemple d'application de l'algorithme du gradient conjugué pour une fonction f en utilisant le logiciel MATLAB :
on s'interesse a étudier le cas de fonction convexe suivante :

$f(x_1, x_2) = 4(x_1^2 + x_2^2) - 2x_1 x_2 - 6(x_1 + x_2)$, avec $x \in \mathbb{R}^2$,pour résoudre le systéme Ax=b ;

en appliquant l'algorithme du gradient conjugué qui a été traité en chapitre 1 en utilisant MATLAB, on obtient les résultat suivants :
sachant qu'on démarre avec un choix de $x^{(0)} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$
A l'itération k=0 ;

$$p = \begin{bmatrix} 0.9837 \\ 1.4239 \end{bmatrix}$$

A l'itération k=1 ;

$$xp = \begin{bmatrix} 1.0917 \\ 1.0459 \end{bmatrix}$$

A l'itération k=2 ;

$$xp = \begin{bmatrix} 0.9993 \\ 1.0194 \end{bmatrix}$$

A l'itération k=3 ;

$$xp = \begin{bmatrix} 1.0042 \\ 1.0021 \end{bmatrix}$$

A l'itération k=4 ;

$$xp = \begin{bmatrix} 1.0000 \\ 1.0009 \end{bmatrix}$$

A l'itération k=5 ;

$$xp = \begin{bmatrix} 1.0002 \\ 1.0001 \end{bmatrix}$$

A l'itération k=6 ;

$$xp = \begin{bmatrix} 1.0000 \\ 1.0000 \end{bmatrix}$$

Et ces figures répresente réspéctivement : Evolution de la norme du gradient pour différente itérations et les Résultats des itérations. La figure 3.1 montre qu'on obtient la solution exacte du systéme linéaire en 6 itération Ainsi la figure 3.2 montre qu'on obtient la solution

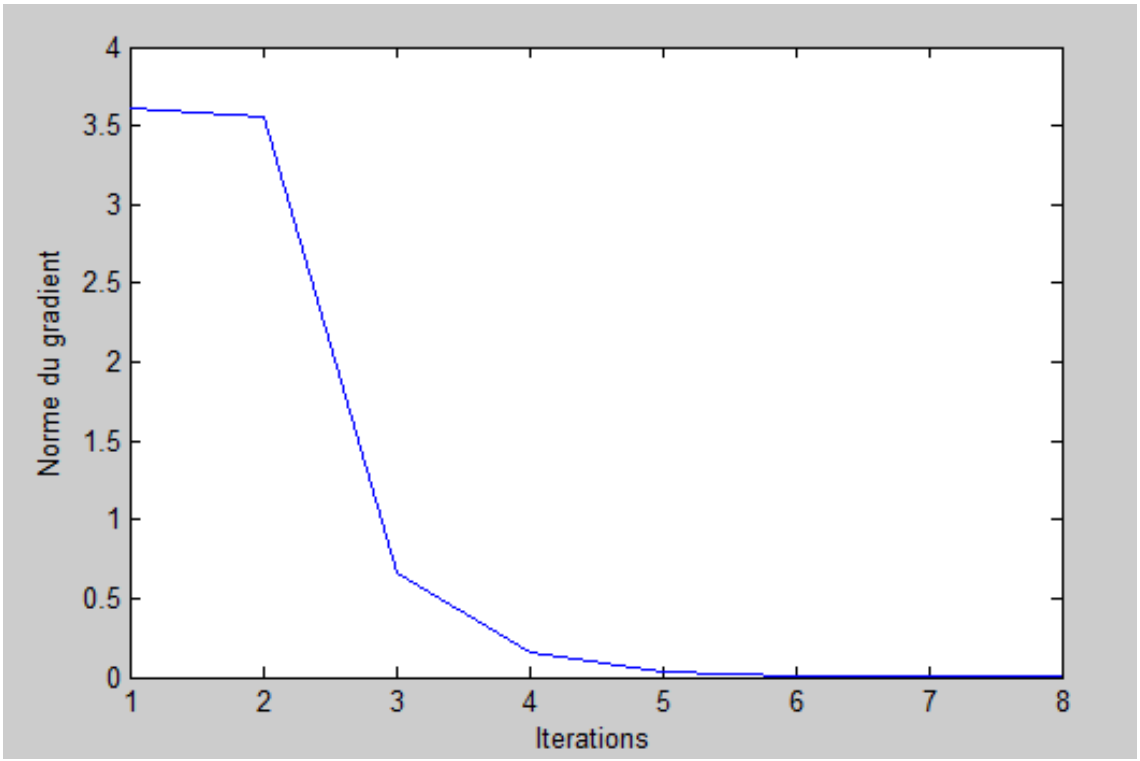

FIGURE 3.1 – Evolution de la norme du gradient pour differente itérations

en 6 itérations

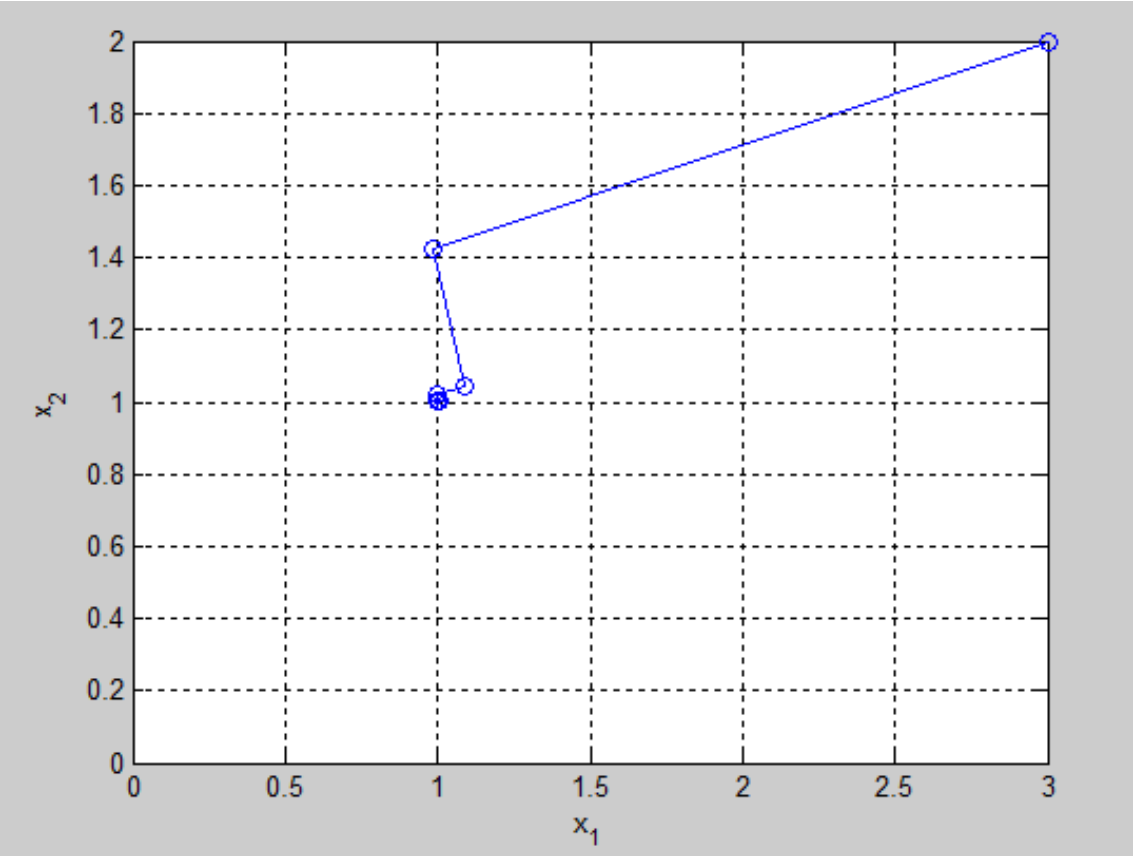

FIGURE 3.2 – les Résultats des itérations

3.2 Méthode de quasi Newton

[3] La méthode de Newton pose plusieurs difficultés.
Approximation de la matrice hessienne
En premier, il y a la nécessité de calculer la matrice hessienne. Pour certains types de problèmes, cela peut devenir problématique. Dans ce cas, on peut avoir recoursà une approximation de la matrice hessienne. Pour cela, on utilise la formule de différences finies

$$H(x)_{e_j} \approx \frac{\nabla f(x + h_{e_j} - \nabla f(x))}{h}$$

où e_j est le j ième vecteur de la base canonique de $\mathbb{R}^n$. $h > 0$ est une petite valeur de l'ordre de 10^{-8}. On notera que $H(x)_{e_j}$ représente la j ième colonne de la matrice hessienne.
Méthode de quasi-Newton modifiée
Cette approche est basée sur l'observation que si M_k est une matrice symétrique et définie-positive, alors :

$d_k = -M_k^{-1}\nabla f(x_k)$ est une direction de descente. En effet, on applique de nouveau Taylor :

$f(x_{k+1}) = f(x_k + d_k) = f(x_k) + (\nabla f(x_k), d_k) + \frac{1}{2}(Hd_k, d_k)$.

En négligeant le terme d'ordre 2, on obtient :
$f(x_{k+1}) = f(x_k) - (M_k^{-1}\nabla f(x_k), \nabla f(x_k) + \frac{1}{2}(Hd_k, d_k)$.

$\approx f(x_k) - (M_k^{-1}\nabla f(x_k), \nabla f(x_k))$;

$\leq f(x_k)$ car $M_k^{-1} > 0$.

Le choix de newton comme direction de descente $d_k = -H(x_k^1\nabla f(x_k)$ ne fonctionne que si la matrice est $H(x_k) > 0$.
Supposons qu'à une certaine itération, la matrice $H(x_k)$ n'est pas définie-positive. Elle admet les valeurs propres $\lambda_1 \leq \lambda_2 \leq ....\lambda_n$.

Si $\lambda_1 < \epsilon$ pour une petite valeur $\epsilon > 0$, on peut translater la matrice $H(x_k)$ de sorte que les valeurs propres soient toujours plus grande que $\epsilon > 0$. Il suffit de poser :
$M_k = H(x_k) + (\epsilon - \lambda_1)$:

Méthode de Newton modifiée :

1. Etant donné une approximation initiale x_0,
2. Calculer la première valeur propre $\lambda_1 de H(x_k)$. Si $\lambda_1 < \epsilon$, poser $M_k = H(x_k) + (\epsilon - \lambda_1)$, sinon $M_k = H(x_k)$.
3. Calculer la direction de descente :$M_k d_k = -\nabla f(x_k)$,
4. Mettre à jour la solution :$x_{k+1} = x_k + d_k$.

Finalement, il est souvent nécessaire de garantir que $f(x_{k+1}) \leq f(x_k)$.
Ceci peut être fait en modifiant la mise à jour de la solution $x_{k+1} = x_k + \alpha_k d_k$ où $\alpha_k > 0$ est choisi de sorte que $\min_\alpha f(x_k + \alpha d_k)$
Dans la pratique, il est préférable de faire plutôt une recherche linéaire à partir de la valeur $\alpha = 1$. Afin de garantir la convergence quadratique, il est important de terminer les itérations avec le choix de $\alpha_k = 1$. Les algorithmes de recherche linéaire seront présentés plus loin.

Exemple d'application de méthode du quasi Newton pour une fonction f en utilisant le logiciel MATLAB :

on s'interesse a étudier le cas de fonction convexe suivante :
$f(x_1, x_2) = 4(x_1^2 + x_2^2) - 2x_1x_2 - 6(x_1 + x_2)$, avec $x \in \mathbb{R}^2$

en appliquant l'algorithme du quasi Newton que nous avons traité dans cette section en utilisant MATLAB, on obtient les résultat suivants :

sachant qu'on démarre avec un choix de $x^{(0)} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ la figure 3.3 suivante montre que la méthode de quasi Newton converge en seulement deux itération

la figure 3.4 suivante montre qu'on obtient la solution en deux itérations Conculsion :vu que

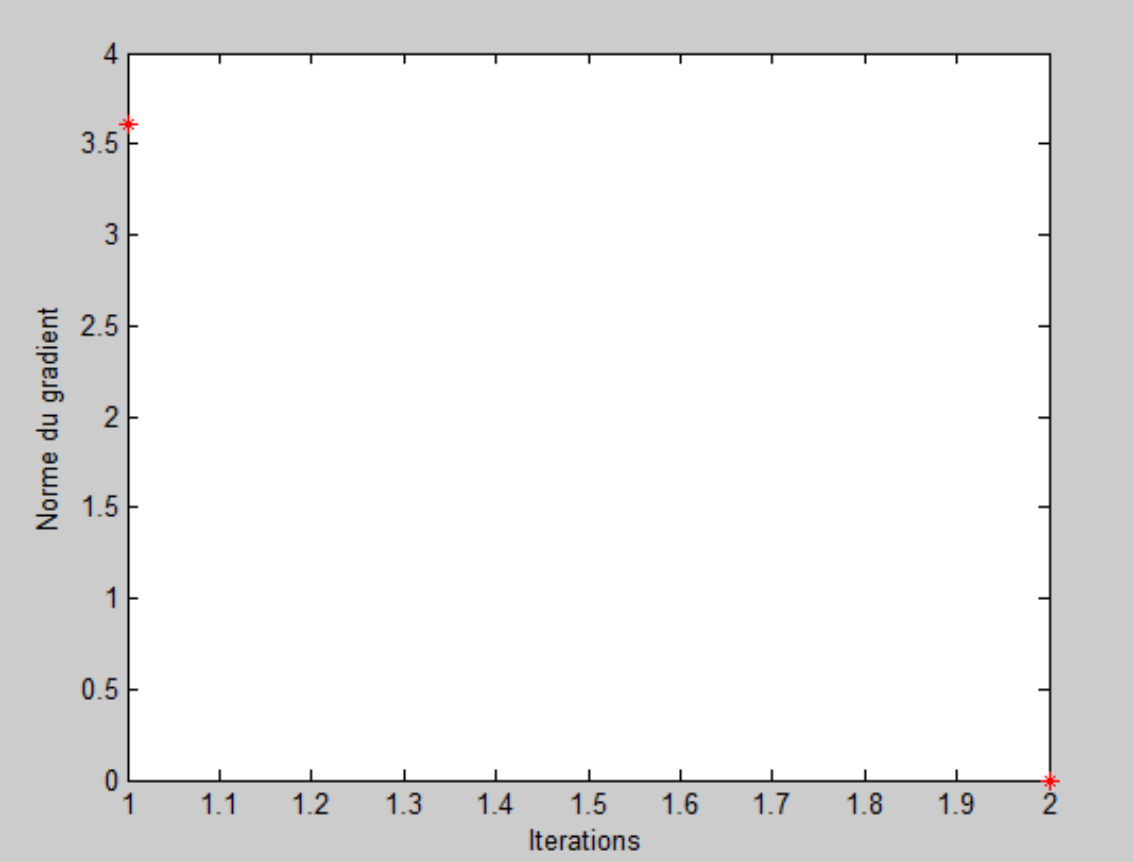

FIGURE 3.3 – Evolution de la norme du gradient pour differentes itérations

la méthode de quasi Newton converge en deux itération seulement, on peux dire qu'elle est meilleure que celle de gradient conjugué qui converge en 5 itérations, par contre cette derniére peut devenir exellente à condition qu'on utilise un préconditionnement(n assez grand). Autrement dit, la méthode de quasi Newton converge rapidement par rapport à la méthode de gradient conjugué.

Le code MATLAB utilisé pour la méthode du gradient conjugué :

```
clc
clear
xp = [3 ; 2] ;
xpp = [10 ;10] ;
tol = 1e-3 ;
i = 1 ;
XP = [xp] ;
Norm = [norm(xp,2)] ;
figure(1)
while norm(grad(xp),2) >= tol
```

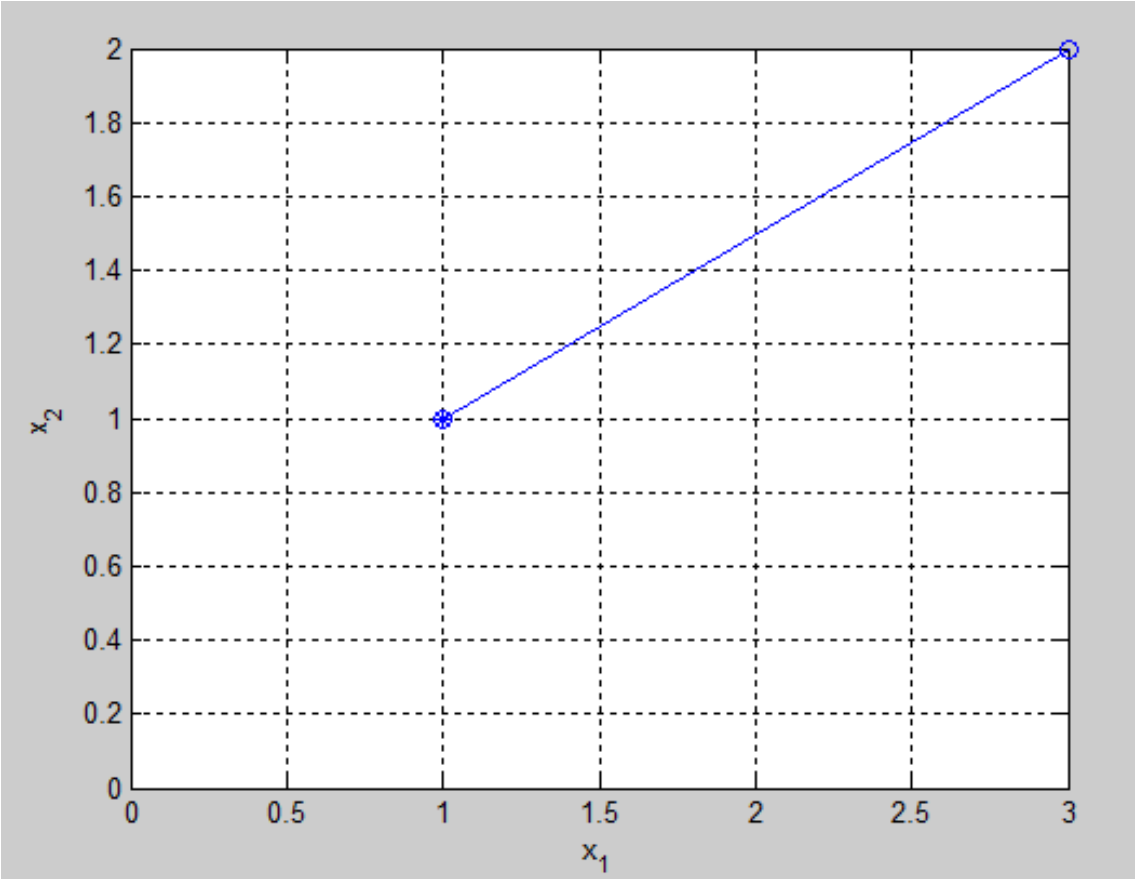

FIGURE 3.4 – les Résultats des itérations

```
if i == 1
beta = 0;
else
beta = norm(grad(xp),2)/norm(grad(xpp),2);
end
p = -grad(xp) + beta^2 * grad(xpp);
alpha = (6 * (p(1) + p(2)) + 2 * (xp(1) * p(2) + xp(2) * p(1)) - 8 * (xp(1) * p(1) + xp(2) *
p(2)))/...(8 * (p(1)^2 + p(2)^2) - 4 * p(1) * p(2));
xp = xp + alpha*p
xpp = xp;
XP = [XP,xp];
Norm = [Norm,norm(grad(xp),2)];
end
XP = [[3;2],XP];
figure(1)
plot(1 :length(Norm),Norm)
xlabel('Iterations')
ylabel('Norme du gradient')
```

figure(2)
plot(XP(1, :),XP(2, :))
hold on
plot(XP(1, :),XP(2, :),'bo')
hold on
plot(XP(1,end),XP(2,end),'*')
axis([0 3 0 2])
grid
xlabel($'x_1'$)
ylabel($'x_2'$)
le code MATLAB utilisé pour la méthode de quasi Newton :

```
clc
clear
close all
xp = [3 ; 2] ;
tol = 1e-3 ;
i = 1 ;
XP = [xp] ;
Norm = [norm(xp,2)] ;
inv_A = inv([8 - 2; -28]);
figure(1)

    while norm(grad(xp),2) >= tol
alpha = 1 ;
xp = xp - alpha * inv_A * grad(xp);
xpp = xp ;
XP = [XP,xp] ;
Norm = [Norm,norm(grad(xp),2)] ;
end
XP = [[3 ;2],XP] ;
figure(1)
plot(1 :length(Norm),Norm,'*r')
xlabel('Iterations')
ylabel('Norme du gradient')
figure(2)
plot(XP(1, :),XP(2, :))
hold on
plot(XP(1, :),XP(2, :),'bo')
hold on
plot(XP(1,end),XP(2,end),'*')
axis([0 3 0 2])
grid
xlabel('x_1')
ylabel('x_2')
```

conclusion

Au cours de cette étude notre savoir a été enrichi en découvrant le domaine de l'optimisation non linéaire et de programmation. Ce projet très intéressant nous a permis de concilier un travail théorique avec une tâche pratique d'implémentation. IL nous a permis d'apprécier l'efficacité des divers types de méthodes, de juger de leur convergence. IL nous reste tant à découvrir et le sujet n'a bien sur pas été épuisé à travers ce document et de nombreux défis restent ouverts, en particulier l'implémentation de nouvelles méthodes. Finalement, nous espérons que ce travail soit bénéfique, et avoir apporté une contribution pour les prochaines recherche.

Bibliographie

[1] Jean Beney. Classification supervisée de documents. *théorie et pratique, Hermes Science*, page 184p, février 2008.

[2] ACHEMINE FARIDA et MERAKEB ABDELKADER. Initiation à l'optimisation non linéaire,aspects théorique et algorithmiques. Thése, Faculté des sciences UMMTO, 2013/2014.

[3] Robert Guénette. Chapitre 3 optimisation différentiable sans contrainte.

[4] Arnaud Guyader. Régression linéaire. thése, Université Rennes 2 Master de Statistique, Année 2012/2013 Premier Semestre.

[5] M. BOUAZIZ Khelifa. Etude de la convergence d'une combinaison de familles du gradient conjugue. Master's thesis, AU DEPARTEMENT DE MATHEMATIQUE FACULTÉ DES SCIENCES UNIVERSITÉ MOHAMED CHERIF MESSAADIA SOUK-AHRAS, 2012 /2013.

[6] Aude Rondepierre. Méthodes numériques pour l'optimisation non linéaire déterministe. thése, Département Génie Mathématique et Modélisation, 4ème année, 2017-2018.

[7] OUKACHA Brahim (UMMTO). Optimisation mono-critère et multi-critère non linéaire. thése, UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU Faculté des Sciences Département de Mathématiques, 15/10/2012.

Résumé

L'optimisation est un outil important en sciences appliquées et pour l'analyse des systèmes physiques. Elle cherche à améliorer une performance en se rapprochant d'un point optimal une fois qu'on a bien identifier l'objectif qui peut être le profit, le temps, l'énergie potentielle ou n'importe quelle quantité ou combinaison de qualité. Notre but est de trouver les valeurs des variables qui optimisent l'objectif.
Les méthodes du gradient conjugué sont très importantes pour la résolution des problèmes d'optimisation non linéaire, en particulier pour les problèmes de grande taille. Cependant, contrairement aux méthodes quasi-Newton. Les méthodes du gradient conjugué sont généralement analysées individuellement.
Mots clés : Optimisation non linéaire, optimisation sans contraintes, gradient conjugué, conditions d'optimalité, méthodes d'optimisation.