UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UNE NOUVELLE APPROCHE POUR LA SÉLECTION DES VARIABLES DANS LE CAS DE MODÈLES DE DISCRIMINATION EN GRANDES DIMENSIONS

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

PAR

RACHID KHAROUBI

JUIN 2016

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

RÉSUMÉ

Le Séparateur à Vaste Marge (*SVM*) est un algorithme d'apprentissage initialement défini pour la discrimination, c'est-à-dire, la prévision d'une variable qualitative binaire (ex. groupes malades et non-malades). Malgré son utilité dans plusieurs domaines d'applications, l'approche *SVM* standard ne permet pas la sélection des prédicteurs importants pour la discrimination, en particulier dans la présence d'un grand nombre de prédicteurs. Plusieurs régularisations de l'approche *SVM* ont été proposées dans la littérature. Parmi les plus importantes, on trouve l'approche de Wang *et al.* (2008). En imposant une contrainte de type ℓ_1-ℓ_2 sur la fonction de perte de la méthode *SVM*, cette approche favorise la parcimonie dans la sélection des prédicteurs et tient compte de la corrélation entre ces derniers. Yang et Zou (2013) proposent un algorithme de type descente par coordonnée qui est efficace et rapide.

Dans certaines situations, les prédicteurs peuvent agir en groupes sur la variable réponse. Ainsi, l'exploitation de telle structure peut s'avérer très utile pour discriminer les deux classes de la variable réponse. Par exemple, dans le domaine de la génétique, les gènes opèrent en groupes pour la régularisation et la survie d'un organisme, et ils agissent de-même pour causer plusieurs maladies complexes comme les cancers. Dans ce mémoire, nous présentons une extension de la méthode *SVM* de Yang et Zou afin d'obtenir une meilleure discrimination de la variable réponse, dans le cas de données de grandes dimensions. Nous proposons un nouveau modèle pour ce type de données. Pour estimer les paramètres de notre modèle et remédier à plusieurs problèmes d'optimisation, nous proposons un algorithme d'estimation qui utilise les techniques de maximisation-minimisation et l'algorithme de descente par coordonnée. Ceci, afin d'accélérer la convergence de notre algorithme. Nous allons montrer que notre méthode favorise la parcimonie et tient compte de la structure de groupes des prédicteurs qui discriminent davantage les deux classes de la variable réponse. Nous illustrons la méthodologie proposée dans ce mémoire à l'aide des études de simulations. Finalement, nous analysons un jeu de données réelles contenant deux groupes de sujets, un groupe de patients atteints du cancer de la prostate et un groupe de sujets non-malades, et décrits par 6033 expressions de gènes (prédicteurs).

INTRODUCTION

L'apprentissage statistique joue un rôle très important dans différents domaines, entre autres, la science, la finance, la santé et l'industrie. C'est un outil de la statistique qui permet un meilleur traitement de données. Il y a deux catégories d'apprentissage statistique : supervisé et non supervisé. Le cas supervisé a pour objectif d'expliquer une variable y par rapport à un ensemble de prédicteurs. Alors que dans le cas non supervisé, on dispose d'un ensemble de p prédicteurs qu'on veut décrire et/ou dont on veut réduire la dimension.

Notons que la régression linéaire et la discrimination sont deux approches de l'apprentissage statistique supervisé.

Actuellement, le traitement des données de grandes dimensions est très fréquent. Ainsi, l'utilisation des méthodes statistiques standard qui ont pour objectif d'extraire de l'information utile dans les données, rencontre des difficultés. Par exemple, dans un modèle de régression linéaire standard, nous minimisons la somme des carrés des résidus, cependant, l'utilisation d'une telle technique n'est pas appropriée et conduit à des estimateurs instables pour les paramètres du modèle. Autrement dit, la méthode de régression linéaire standard donne toujours des estimateurs sans biais pour les paramètres du modèle, cependant leurs variance a tendance à être très grande en présence d'un grand nombre de prédicteurs. Plusieurs techniques statistiques ont été proposées pour remédier à ce problème.

Dans le cadre de la régression linéaire, la méthode *Ridge* consiste à minimiser la somme des carrés des résidus en respectant une contrainte de type ℓ_2 sur les coefficients. En effet, cette méthode rétrécit les estimateurs de tous les coefficients

vers zéro et elle ne produit pas un modèle parcimonieux, i.e. simple à interpréter et qui peut se généraliser à la population globale de la variable réponse. La méthode Lasso consiste à minimiser la somme des carrés des résidus en respectant une contrainte de type ℓ_1 sur les coefficients. En effet, cette méthode produit la parcimonie dans le modèle : elle rétrécit les coefficients et sélectionne les variables de manière simultanée. Mais, la pénalité ne tient pas compte de la corrélation entre les prédicteurs. Pour remédier aux problèmes de la méthode *Ridge* et de la méthode *Lasso*, Zou et Hastie (2005) proposent Elastic Net qui fournit à la fois un modèle parcimonieux et tient compte de la corrélation entre les prédicteurs. Cependant, pour certaines situations réelles, nous prétendons qu'il y a une structure inconnue de groupes dans les prédicteurs que nous désirons capturer et exploiter pour obtenir de meilleurs estimateurs des paramètres du modèle, ainsi qu'une meilleure sélection de prédicteurs (i.e. groupes de prédicteurs qui sont associés à la variable réponse). Ainsi, la méthode CEN (de l'anglais Cluster Elastic Net) proposée par Witten *et al.*(2014) est la plus appropriée dans ce contexte. En effet, la méthode CEN cherche une structure de groupes qui est informative et associée à la variable réponse.

Dans le cas de la discrimination, la régression logistique consiste à prédire une variable réponse catégorielle, en minimisant le logarithmique de la vraisemblance qui est considérée comme une fonction de perte.

Une autre technique performante dans ce cadre est la méthode *SVM* (de l'anglais Support Vector Machines) de Vapnik (1995). Comme dans le cas de la régression linéaire, plusieurs régularisations de cette technique ont été proposées, entre autres, la méthode *SVM*-ℓ_2, Hastie *et al.*(2001), la méthode *SVM*-ℓ_1, Bradley et Mangasarian (1998) et la méthode HHSVM-EN (Huberized Hinge Support Vector Machines via Elastic Net), Wang *et al.* (2008). La méthode de Wang *et al.* (2008) consiste à minimiser une fonction de perte convexe, ϕ_c, mais la résolution

du problème d'optimisation sous-jacent ne mène pas à des solutions explicites pour les estimés des paramètres du modèle, et elle utilise ainsi un algorithme itératif. Ceci rend la convergence de l'algorithme de cette méthode très lente. En particulier, la fonction de perte ϕ_c est dérivable mais sa dérivée première n'est pas lisse. Pour remédier à ceci et pouvoir généraliser la technique de la descente par coordonnée, Yang et Zou (2013) proposent un algorithme qui se base sur la technique de majoration-minimisation de la fonction objective proposée par Wang *et al.* (2008). Cette technique est rapide et efficace, mais, elle a tendance à rétrécir les estimateurs des coefficients du même groupe de prédicteurs vers zéro.

La pénalité Elastic Net de HHSVM permet de tenir compte de la structure de groupe de prédicteurs de manière implicite avec peu de garantie. Tandis que la méthode CEN a été introduite, en régression linéaire, afin de corriger cet inconvénient en ajoutant dans la pénalité un terme permettant le regroupement de prédicteurs suivi d'une étape de rétrécissement. L'objectif de notre recherche est d'adapter la pénalité CEN dans le cadre de l'approche HHSVM afin de rendre la méthode HHSVM-EN apte à tenir compte de la structure de groupes des prédicteurs qui sont associés à la variable réponse.

Ce mémoire est composé de cinq chapitres. Dans le premier chapitre, nous allons définir l'apprentissage statistique et nous allons montrer son utilité. Dans le deuxième chapitre, nous allons présenter plusieurs méthodes dans le cadre de la régression linéaire et nous allons discuter différentes techniques de régularisation. Dans le chapitre trois, nous allons introduire des méthodes linéaires pour la classification. Nous allons présenter notre nouvelle approche dans le chapitre quatre, et nous allons conserver le chapitre cinq pour les résultats numériques des études de simulation et l'étude d'une base de données réelles, afin d'illustrer la méthodologie que nous proposons.

CHAPITRE I

L'APPRENTISSAGE STATISTIQUE

1.1 Qu'est ce que l'apprentissage statistique ?

L'apprentissage statistique est un ensemble d'outils de la statistique que l'on utilise en vue de comprendre des bases de données. Ces outils peuvent être classés en deux grandes catégories : supervisé ou non supervisé, selon la nature des données.

L'apprentissage statistique supervisé a pour objectif la prédiction d'une variable réponse, y, en se basant sur des prédicteurs (des variables de type entrée), dans le but de généraliser la prédiction à la population de la variable réponse (c.-à-d. généraliser la prédiction/décision pour des données qui ne font pas partie de l'échantillon sous étude). Dans l'apprentissage statistique supervisé, les données sont collectées sous la forme d'un échantillon d'apprentissage de type entrée-sortie :$\{(\mathbf{x}_{1.}, y_1), ..., (\mathbf{x}_{n.}, y_n)\}$, où $y_i, i \in 1, ..., n$ est la variable réponse pour l'observation i, et $\mathbf{x}_{i.} = (x_{i1}, ..., x_{ip}), i \in 1, ..., n$, $\mathbf{x}_{i.} \in \mathbb{R}^p$, sont les p prédicteurs. Dans le cas de la régression multiple, la variable réponse $\mathbf{y} \in \mathbb{R}^n$ est continue et dans le cas de discrimination, elle est discrète : $\mathbf{y} \in \{0, 1\}^n$ ou $\mathbf{y} \in \{0, 1, ..., q\}^n$, avec $q \geq 2$ dans le cas ou le nombre de classes est supérieur à 2.

L'apprentissage statistique non supervisé consiste à décrire et résumer l'information sur les données en absence d'une variable réponse. En effet, pour ce type

d'apprentissage, on cherche une structure de groupes, des classes homogènes, ou réduire la dimension des données, etc.

La méthode que nous proposons dans ce mémoire se situe dans la première catégorie. Il s'agit d'une nouvelle technique de classification, où $\mathbf{y} \in \{0,1\}^n$ est une variable catégorielle.

1.2 Mise en situation

La base de données «publicité», James *et al.*(2013), comprend les ventes d'un produit particulier dans 200 marchés différents, ainsi que les budgets dans chacun de ces marchés pour trois médias différents, soient Tv, Radio et Journaux. Notre objectif est d'améliorer les ventes de ce produit.

En effet, développer un modèle pour prédire les ventes sur la base des trois budgets médias peut s'avérer très utile. Un tel modèle pour cette situation peut être donné comme suit, $\mathbf{y}$: ventes (réponse quantitative), $\mathbf{x} = (x_1, x_2, x_3)$ désigne le budget publicitaire avec x_1 est le budget Tv, x_2 est le budget Radio, x_3 est le budget Journaux. Ce modèle suppose qu'il y a une relation entre $\mathbf{y}$ et $\mathbf{x}$, qui est décrite comme suit

$$\mathbf{y} = f(\mathbf{x}) + \epsilon, \tag{1.1}$$

où f est une fonction inconnue et ϵ est un bruit ou un terme d'erreur.

1.3 Pourquoi estimer la fonction f ?

Il y a deux raisons principales pour lesquelles on estime cette fonction :

* Prédiction et généralisation : dans plusieurs scénarios, des données $\mathbf{x}$ (prédicteurs) sont disponibles. Cependant, la variable réponse y n'est pas facile à obtenir. Ainsi, il faut la prédire en utilisant l'estimé de f en fonction de

$\mathbf{x}$. On peut écrire

$$\hat{\mathbf{y}} = \hat{f}(\mathbf{x}).$$

* Inférence : Nous sommes souvent intéressés à la façon dont $\mathbf{y}$ est affectée lorsque $\mathbf{x}$ change. Dans cette situation, on estime $\hat{f}$ dans le but de comprendre la relation entre $\mathbf{y}$ et $\mathbf{x}$. Autrement dit, comment $\mathbf{y}$ change lorsque $\mathbf{x}_1$,...,$\mathbf{x}_p$ changent ? Dans ce cadre, on peut s'intéresser à répondre aux questions suivantes :

  1. Quels sont les véritables prédicteurs associés à la variable réponse ?
  2. Quelle est la relation entre la réponse et chacun des prédicteurs ?
  3. Peut-t-on utiliser un modèle linéaire pour présenter la relation entre $\mathbf{y}$ et les prédicteurs, ou bien recourir à des formes de $f(.)$ beaucoup plus complexes ?

  Exemple :

  Pour la base de données «publicité», on peut se poser les questions suivantes :

  1. Quel type de média contribue d'avantage aux ventes ?
  2. Lequel des trois médias génère le plus grand gain pour les ventes ?
  3. De combien augmentent les ventes lors d'une hausse de la publicité Tv ?

1.4 Comment estimer la fonction f ?

Dans la littérature, il existe plusieurs approches linéaires et non linéaires, paramétriques et non paramétriques, pour estimer la fonction f. Rappelons-nous que l'échantillon d'apprentissage contient les couples $\{(\mathbf{x}_1., y_1), ..., (\mathbf{x}_n., y_n)\}$, où $y_i, i \in 1, ..., n$ est la variable réponse pour l'observation i, et $\mathbf{x}_i. = (x_{i1}, ..., x_{ip}), i \in 1, ..., n$, $\mathbf{x}_i. \in \mathbb{R}^p$, sont les p prédicteurs. Notre but est d'appliquer une méthode d'apprentissage statistique à cet échantillon afin d'estimer la fonction f. En d'autres termes,

on cherche une fonction $\hat{f}$, tel que $\hat{y}_i = \hat{f}(\mathbf{x}_{i.})$ pour toute observation $(\mathbf{x}_{i.}, y_i)$. Pour obtenir une telle fonction, on minimise une fonction de perte $L\{y_i, f(\mathbf{x}_{i.})\}$ qui mesure les écarts entre les observations y_i et le modèle $f(\mathbf{x}_{i.})$. Une telle fonction de perte peut prendre différentes formes : la différence en norme L_1, en norme L_2, etc.

Par exemple, la fonction de perte avec la forme L_2 est définie par

$$L(\mathbf{y}, f(\boldsymbol{X})) = ||\mathbf{y} - f(\boldsymbol{X})||^2 = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_{i.})^2,$$

avec $f(\boldsymbol{X}) = (f(\mathbf{x}_{1.}), ..., f(\mathbf{x}_{n.}))$.

1.4.1 Méthodes paramétriques

Les méthodes paramétriques se font en deux étapes : une première étape dans laquelle on spécifie le modèle et une deuxième étape pour estimer les paramètres du modèle.

Étape 1 : On suppose une forme explicite pour la fonction f. Une façon simple consiste à supposer une relation linéaire entre y et les $\mathbf{x}_{i.}$, avec f est définie par

$$f(\mathbf{x}_{i.}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}.$$

C'est un modèle de régression linéaire qui suppose que f est linéaire en β_j. Ainsi, estimer la fonction f revient à estimer les paramètres $\beta_0, \beta_1, ..., \beta_p$.

Étape 2 : Après avoir choisi le modèle, nous avons besoin d'une approche ou d'une procédure que nous appliquons à l'échantillon d'apprentissage afin d'estimer les paramètres $\beta_0, \beta_1, ..., \beta_p$ qui minimise la fonction de perte $L(y, .)$.

L'approche populaire la plus utilisée dans ce genre de problèmes est celle des moindres carrés, où L est la perte définie par $L(y, f(x)) = (y - f(x))^2$.

En effet, cette méthode donne toujours des solutions explicites pour les paramètres du modèle et fonctionne bien si nous avons une relation linéaire entre la variable réponse et les prédicteurs. Dans le cas contraire, il existe plusieurs méthodes d'apprentissage statistique dont nous allons discuter plus tard dans ce mémoire.

1.4.2 La régression locale comme méthode non paramétrique

Les méthodes non paramétriques ne font pas d'hypothèses explicites sur la forme fonctionnelle de f. La régression locale est une approche non paramétrique qui consiste à estimer la fonction f en différents points x_0, en tenant compte des observations $x_i, \quad i = 1, 2, .., n$ qui sont proches de $x_0 \in \mathbb{R}^p$. Aux fins d'illustration, nous allons supposer le cas où $p = 1$ (un seul prédicteur). Nous pouvons définir la régression locale de la manière suivante.

On considère $\mathbf{x}= (x_1, x_2, ..., x_n)$ un prédicteur et $\mathbf{y}= (y_1, ..., y_n)$ la variable réponse. Pour estimer f au point x_0, on assigne des poids aux observations x_i, qui sont proches de x_0. De tels poids sont notés : $K_i = K(x_i, x_0)$, et nous donnons un poids égal à zéro pour les points qui sont éloignés de x_0. Un exemple pour K_i est le noyau uniforme qui est défini par

$$K_i = \frac{1}{2}\mathbb{1}_{\{|\frac{x_0 - x_i}{h}| \leq 1\}},$$

avec $h > 0$, est la taille de la fenêtre qu'elle faut choisir judicieusement. La prochaine étape consiste à appliquer la méthode des moindres carrés pondérés pour estimer les paramètres du modèle.

En effet, on cherche le couple $(\hat{\beta}_0, \hat{\beta}_1)$ qui minimise la somme suivante :

$$\sum_{i=1}^{n} K_i(y_i - \beta_0 - \beta_1 x_i)^2,$$

et l'estimé de la fonction f en x_0 est donné par la forme

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0.$$

CHAPITRE II

LES MÉTHODES LINÉAIRES DANS LE CAS DE RÉGRESSION LINÉAIRE

Dans ce chapitre, nous allons introduire quelques méthodes linéaires d'apprentissage statistique où la variable y est continue.

2.1 La régression linéaire multiple

Ce n'est qu'une généralisation de la régression linéaire simple. On considère une matrice $n \times p$ pour les prédicteurs observés,

$$\boldsymbol{X} = (\mathbf{x}_1, ..., \mathbf{x}_p),$$

avec $\mathbf{x}_j = (x_{1j}, ..., x_{nj})^\top$, $j = 1, ..., p$. Nous allons supposer dans le reste de ce mémoire que $\boldsymbol{X}$ est centrée et réduite. Autrement dit,

$$\sum_{i=1}^{n} x_{ij} = 0, \sum_{i=1}^{n} x_{ij}^2 = 1.$$

L'objectif de la régression linéaire multiple est de prédire la valeur de la variable réponse $\mathbf{y} \in \mathbb{R}^n$.

Supposons que $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^\top$ et β_0 est l'ordonnée à l'origine. Le modèle de l'équation (1.1) dans le cas d'une fonction linéaire peut s'écrire sous la forme

suivante :

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i, \quad i = 1, ..., n. \tag{2.1}$$

Les vecteurs de données $\mathbf{x}_j$ peuvent prendre différentes formes :

+ des données originales ;
+ des transformations de données originales (log, carré, etc) ;
+ autres représentations : polynomiales, etc.

Pour toute forme de données, le modèle est linéaire en terme des paramètres β_j. Ainsi, pour estimer les coefficients β_j, la méthode des moindres carrés est la plus utilisée dans ce genre de résolutions de problème. L'objectif est de minimiser la somme des carrés des résidus définie par

$$RSS(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2. \tag{2.2}$$

La solution de l'équation (2.1) sous forme matricielle est donnée par

$$\hat{\boldsymbol{\beta}}^{MCO} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \mathbf{y}. \tag{2.3}$$

2.2 Méthodes de régularisations ou de rétrécissements

2.2.1 Motivation

La régression linéaire multiple est souvent utilisée pour évaluer un modèle et prévoir des valeurs futures de la variable réponse ainsi qu'examiner la relation de cette dernière avec les prédicteurs. D'une part, l'exactitude de la prédiction du modèle est importante. D'autre part, la parcimonie du modèle a plus d'intérêt (c.-à-d. un modèle simple à interpréter est toujours préférable en pratique). En présence d'un grand nombre de prédicteurs, la méthode des moindres carrés donne des estimateurs de faible biais et de grande variance. Au cours des dernières

décennies, plusieurs méthodes de régularisations ou de rétrécissements ont été développées pour résoudre un tel problème, entre autres, la régression *Ridge* (Hoerl et Kennard, 1970 a,b), suivie par la régression *Bridge* (Frank et Freidman, 1993), *Lasso* (Tibshirani, 1996), et d'autres méthodes récentes comme *LARS* de Efron *et al.*(2004) et la méthode Élastique Net (Zou et Hastie, 2005). Dans ce qui suit, nous allons montrer les problèmes rencontrés par la méthode des moindres carrés.

- Si la véritable relation entre la variable réponse et les prédicteurs est sensiblement linéaire, les estimateurs issus de la méthode des moindres carrés auront un biais faible.
  - Si le nombre n d'observations est beaucoup plus grand que le nombre p de variables ($n \gg p$), alors les estimateurs de la méthode des moindres carrés ont tendance à avoir également une petite variance.
  - Si $p > n$, alors les estimateurs de la méthode des moindres carrés ne sont pas uniques et leur variance a tendance à être très grande, bien que leur biais reste petit. Ainsi, on ne peut pas utiliser la méthode en présence de tous les prédicteurs. Les méthodes de rétrécissement offrent une réduction de cette variance au prix d'une augmentation du biais. Ceci se fait en ajoutant des pénalités à la fonction de perte de la méthode des moindres carrés. De telles pénalités forcent les coefficients à être petits en les rétrécissant. Ainsi, cela permet de réduire la variance des estimateurs, même si cela introduit une petite augmentation du biais.
- Interprétation du modèle : souvent, certains ou plusieurs prédicteurs utilisés dans un modèle de régression multiple ne sont pas associés à la variable réponse y. Les prédicteurs non pertinents compliquent la résolution du modèle résultant. En éliminant ces variables (c.-à-d. en forçant les estimateurs des coefficients correspondants à être nuls), nous pouvons obtenir un modèle qui est plus facile à interpréter. Cependant, il est peu probable que

la méthode des moindres carrés nous donne des estimateurs de coefficients qui sont exactement des zéros.

Dans cette section, nous allons voir quelques méthodes qui permettent la résolution de ce problème et qui peuvent fournir des estimations nulles des coefficients de variables qui s'apparentent à des bruits.

- Le rétrécissement : cette approche consiste à ajuster un modèle impliquant tous les p prédicteurs. Toutefois, les coefficients estimés sont rétrécis vers zéro par rapport aux estimateurs de la méthode des moindres carrés. Ce rétrécissement (également connu sous le nom de régularisation) a pour effet de réduire la variance. Selon la méthode de régularisation effectuée, certains coefficients peuvent être estimé exactement par zéro. Par conséquent, ces méthodes peuvent également effectuer la sélection des variables importantes pour la variable réponse.

On considère le modèle linéaire de l'équation (2.1). Rappelons que la matrice des données $\boldsymbol{X}$ est centrée et réduite et on suppose que $\mathbf{y}$ est centrée. Les méthodes de régularisation cherchent à estimer $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^\top$ comme solution du problème de minimisation

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda \boldsymbol{p}(\boldsymbol{\beta}),$$

où $\lambda \geq 0$ est un paramètre de régularisation que l'on choisit avec la méthode de validation croisée ou un critère d'information AIC ou BIC et $\boldsymbol{p}(\boldsymbol{\beta})$ est une pénalité qui dépend de $\boldsymbol{\beta}$. Notons que AIC est le critère d'information d'Akaike et BIC est le critère d'information bayésien. Dans les sections suivantes, trois formes de pénalités importantes seront présentées comme techniques de régularisation. Avant de présenter ses techniques je vais énoncer quelques définitions importantes pour l'optimisation.

2.2.2 Rappel

Dans cette section, nous allons rappeler quelques notions de l'optimisation qui vont nous servir dans le reste de ce mémoire.

1) Soit E un espace euclidien et $C \subset E$ un convexe. On dit qu'une fonction $f : C \to \mathbb{R}$ est convexe si pour tout $x, y \in C$ et $0 \leq \lambda \leq 1$, nous avons

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y).$$

2) Un vecteur $s \in \mathbb{R}^p$ est un sous-gradient d'une fonction f en un point x ssi

$$f(y) \geq f(x) + s^\top (y - x); \forall \quad y \in \mathbb{R}^p.$$

3) $\partial f(x)$ est l'ensemble de tous les sous gradients de f en x.

4) Pour une fonction f convexe et dérivable, on a

$$\partial f(x) = \{\nabla f(x)\}.$$

5) Soit $f : C \to \mathbb{R}$, où C est un ouvert de $\mathbb{R}^p$.

La dérivée directionnelle de f en $x \in C$ dans la direction $d \in \mathbb{R}^n$ est définie par la limite quand elle existe

$$f'(x; d) = \lim_{t \downarrow 0} \frac{f(x + td) - f(x)}{t}.$$

Lorsque f est convexe, la limite existe toujours.

6) Nous avons également $s \in \partial f(x)$ si et seulement si $f'(x; d) \geq s^\top d; \quad \forall \quad d \in \mathbb{R}^p$.

Remarque :

Notez que la fonction sous-gradient existe même si la fonction n'est pas différentiable. Dans ce mémoire plusieurs fonctions objectives ne sont pas dérivables au point zéro à cause de la pénalité L_1.

2.2.3 La méthode *Ridge*

La méthode *Ridge* est une technique de régularisation qui se base sur la norme L_2. Cette méthode a comme pénalité

$$\boldsymbol{p}(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_2^2,$$

avec

$$||\boldsymbol{\beta}||_2^2 = \sum_{i=1}^{p} \beta_i^2.$$

L'estimateur *Ridge* de $\boldsymbol{\beta}$ est défini par

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

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

$$\hat{\boldsymbol{\beta}}^{Ridge} = \arg\min_{\boldsymbol{\beta}} \{\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\}.$$

La fonction à minimiser peut s'écrire sous forme matricielle :

$$RSS(\lambda) = (\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta},$$

ou encore

$$RSS(\lambda) = \mathbf{y}^\top \mathbf{y} \ - 2\boldsymbol{\beta}^\top \boldsymbol{X}^\top \mathbf{y} + \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta}.$$

Si on dérive par rapport à $\boldsymbol{\beta}$ et qu'on pose les dérivées égales à zéro, on trouve

$$\begin{aligned} \boldsymbol{X}^\top \mathbf{y} &= \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + \lambda \boldsymbol{\beta}, \\ &= (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbf{I})\boldsymbol{\beta}. \end{aligned}$$

La matrice symétrique $\boldsymbol{X}^\top \boldsymbol{X}$ étant définie semi-positive, toutes ses valeurs propres sont non négatives et donc $\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbf{I}$ est symétrique et définie positive ($\lambda > 0$), donc inversible. Alors, l'estimateur *Ridge* est donné par la formule explicite :

$$\hat{\boldsymbol{\beta}}^{Ridge} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbf{I})^{-1} \boldsymbol{X}^\top \mathbf{y}.$$

Dans le cas où $\boldsymbol{X}$ est orthogonale,

$$\hat{\boldsymbol{\beta}}^{Ridge} = (1+\lambda)^{-1} \boldsymbol{X}^\top \mathbf{y} = (1+\lambda)^{-1} \hat{\boldsymbol{\beta}}^{MCO},$$

et

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

avec $\hat{\beta}_j^{MCO}$ est donné par l'équation (2.3).

1 - Avantages de la méthode *Ridge* :

- rétrécir les coefficients $\boldsymbol{\beta}$;
- très performante, en présence de corrélation entre les colonnes de $\boldsymbol{X}$;
- elle améliore l'erreur de prédiction, en réduisant la variance des estimateurs.

2 - Inconvénients :

- c'est une méthode non appropriée pour la sélection des variables. En effet, si des prédicteurs sont fortement corrélés entre eux, leurs coefficients seront très proches les uns des autres ;
- elle ne produit pas de parcimonie dans le modèle. Autrement dit, la méthode ne pénalise pas les variables nuisibles par des coefficients exactement nuls.

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

Dans la section suivante, nous allons voir la méthode Lasso comme méthode appropriée pour la prévision et la sélection des variables simultanément.

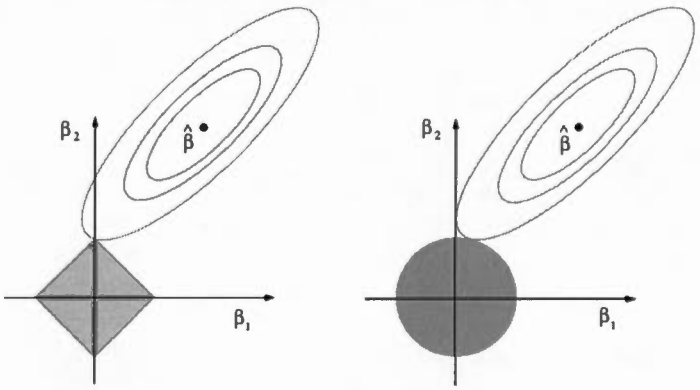

Figure 2.1: Le carré désigne la contrainte $|\beta_1|+|\beta_2| \leq s$ pour la régression *Lasso*, le cercle illustre la contrainte $|\beta_1|^2+|\beta_2|^2 \leq s$ pour la régression *Ridge*, les contours des $RSS(\boldsymbol{\beta})$ et $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^{MCO}$.

2.2.4 La méthode *Lasso*

La méthode *Lasso* (Tibshirani, 1996) est une technique de régularisation qui se base sur la norme L_1. La pénalité de la méthode *Lasso* est donnée par

$$\boldsymbol{p}(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_1,$$

avec

$$||\boldsymbol{\beta}||_1 = \sum_{i=1}^{p} |\beta_i|.$$

La figure (2.1), James *et al.*(2013), décrit les comportements des solutions des deux techniques *Lasso* et *Ridge*.

Remarque: Les sommes de carrés des résidus $RSS(\boldsymbol{\beta})$ peuvent s'écrire sous la forme

$$RSS(\boldsymbol{\beta}) = (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{MCO})^\top \boldsymbol{X}^\top \boldsymbol{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{MCO}) + \textit{constante},$$

qui est l'équation des ellipses que nous observons dans la figure 2.1. L'estimateur de $\boldsymbol{\beta}$ par la méthode Lasso est défini par

$$\hat{\boldsymbol{\beta}}^{Lasso} = \arg\min_{\beta}\{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2\},$$

tel que

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

Noter qu'on peut écrire le problème *Lasso* en un problème équivalent avec le lagrangien

$$\hat{\boldsymbol{\beta}}^{Lasso} = \arg\min_{\boldsymbol{\beta}}\{\frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\}. \tag{2.4}$$

2.2.5 Calcul analytique de la solution de la méthode *Lasso*

Cas général où $\boldsymbol{X}$ est quelconque :

Nous allons chercher maintenant la forme explicite de la solution de l'équation (2.4) pour chaque β_j, en gardant tous les autres paramètres fixes. L'équation (2.4) est équivalente à l'écriture matricielle suivante

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{Lasso} &= \arg\min_{\boldsymbol{\beta}}\{\frac{1}{2}(\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\sum_{j=1}^{p}|\beta_j|\},\\ &= \arg\min_{\boldsymbol{\beta}}\{\frac{1}{2}\mathbf{y}^\top\mathbf{y} - \mathbf{y}^\top\boldsymbol{X}\boldsymbol{\beta} + \frac{1}{2}\boldsymbol{\beta}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\beta} + \lambda\sum_{j=1}^{p}|\beta_j|\}.\end{aligned}$$

Si nous gardons seulement les quantités qui dépendent de β_j, la fonction à minimiser en β_j sera

$$L(\beta_j) = -\mathbf{x}_j^\top\mathbf{y}\beta_j + \frac{1}{2}\sum_{i=1}^{n}x_{ij}^2\beta_j^2 + \sum_{i=1}^{n}\sum_{k\neq j}x_{ik}x_{ij}\beta_k\beta_j + \lambda|\beta_j|.$$

C'est une équation quadratique en β_j dérivable pour $\beta_j \neq 0$ et qui a pour sous-gradient

$$\frac{\partial L(\beta_j)}{\partial \beta_j} = -\mathbf{x}_j^\top \mathbf{y} + \sum_{i=1}^{n} x_{ij}^2 \beta_j + \sum_{k \neq j} x_{ik} x_{ij} \beta_k + \lambda \text{signe}\{\beta_j\},$$

Alors

$$-\mathbf{x}_j^\top \mathbf{y} + \sum_{i=1}^{n} x_{ij}^2 \beta_j + \sum_{k \neq j} x_{ik} x_{ij} \beta_k + \lambda \text{signe}\beta_j = 0.$$

Nous avons que $\sum_{i=1}^{n} x_{ij}^2 = 1$. Ainsi, nous obtenons

$$\hat{\beta}_j = \sum_{i=1}^{n} (y_i - r_i^{(j)}) x_{ij} - \lambda \text{signe}\{\beta_j\},$$

avec $\sum_{k \neq j} x_{ik} \beta_k = r_i^{(j)}$.

Ceci est équivalent à

$$\hat{\beta}_j = \text{signe}\{\sum_{i=1}^{n} (y_i - r_i^j) x_{ij}\} (|\sum_{i=1}^{n} (y_i - r_i^j) x_{ij}| - \lambda)_+,$$

avec $(x)_+ = \max(0, x)$.

C'est une application directe du théorème 1, Antoniadis et Fan, (2001). En effet, la fonction a minimiser est de la forme $\ell(x) = \frac{1}{2}(x-a)^2 + p_\lambda(|x|)$, tel que $\boldsymbol{p}_\lambda(|x|) = \lambda|x|$.

Remarque : Il faut signaler que $\hat{\beta}_j$ dépend des autres β_l, $l \neq j$. Cependant, ces valeurs sont connues à chaque étape de l'algorithme de la descente par coordonnée, une notion qui sera abordée en détail dans le chapitre 4.

Cas particulier où $\boldsymbol{X}$ est orthogonale :

La fonction objective à minimiser dans ce cas sera

$$L(\beta_j) = -x_j^\top y \beta_j + \frac{1}{2}\beta_j^2 + \lambda|\beta_j|.$$

Ainsi, la solution est donnée par

$$\hat{\beta}_j = \text{signe}(x_j^\top \boldsymbol{y})(|x_j^\top \boldsymbol{y}| - \lambda)_+.$$

Ceci nous donne

$$\hat{\beta}_j^{Lasso} = \text{signe}\{\hat{\beta}_j^{MCO}\}(|\hat{\beta}_j^{MCO}| - \lambda)_+.$$

1 - **Avantages de la méthode *Lasso*** :

- elle crée une parcimonie. Cela veut dire qu'elle élimine les variables nuisibles dans le modèle en estimant leur coefficients dans le modèle par des zéros ;
- c'est une bonne méthode pour choisir les variables qui contribuent le plus dans le modèle ;
- elle rétrécit les coefficients $\boldsymbol{\beta}$ vers zéro.

2 - **Inconvénients** :

- c'est une méthode non appropriée pour la sélection des groupes des prédicteurs. En effet, si des prédicteurs sont fortement corrélés entre eux, la méthode *Lasso* choisit un prédicteur et pénalise les autres avec des coefficients nuls ;
- dans le cas où $p > n$, l'approche Lasso choisie au maximum n variables.

Dans la prochaine section, nous allons présenter la méthode Élastique Net qui remédie aux problèmes rencontrés par la méthode *Lasso*.

2.2.6 La méthode Élastique Net (EN)

La méthode EN est une technique de régularisation qui combine les deux normes, la norme L_1 et la norme L_2. Ainsi, cette méthode est un compromis entre la méthode *Lasso* et la méthode *Ridge*. Elle est introduite la première fois par Zou

et Hastie (2005). Sa pénalité est donnée par

$$p(\boldsymbol{\beta}) = \lambda_1 ||\boldsymbol{\beta}||_1 + \lambda_2 ||\boldsymbol{\beta}||_2^2. \tag{2.5}$$

Cas général où $\boldsymbol{X}$ est quelconque :

L'estimateur $\hat{\boldsymbol{\beta}}^{EN}$ est donné par la solution du problème d'optimisation suivant

$$\hat{\boldsymbol{\beta}}^{EN} = \arg\min_{\boldsymbol{\beta}} \{||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda_1 ||\boldsymbol{\beta}||_1 + \lambda_2 ||\boldsymbol{\beta}||_2^2\}.$$

Les calculs ressemblent aux calculs faits pour la méthode Lasso. Ainsi, la solution $\hat{\beta}_j$ est donné par

$$\hat{\beta}_j^{\,EN} = \frac{S(\sum_{i=1}^n (y_i - r_i^{(j)}) x_{ij}, \frac{\lambda_1}{2})}{1 + \lambda_2},$$

avec S est la fonction *Soft-tresholding* définie par

$$S(a, b) = \text{signe}(a)(|a| - b)_+. \tag{2.6}$$

Cas où $\boldsymbol{X}$ est orthogonale :

Dans ce cas, la fonction à minimiser est donnée par

$$L(\beta_j) = -2\mathbf{x}_j^\top \mathbf{y}\ \beta_j + (1 + \lambda_2)\beta_j^2 + \lambda_1 |\beta_j|.$$

C'est une équation quadratique en β_j dérivable pour $\beta_j \neq 0$ et qui a pour sous-gradient

$$\frac{\partial L(\beta_j)}{\partial \beta_j} = -2\mathbf{x}_j^\top \mathbf{y} + 2(1 + \lambda_2)\beta_j + \lambda_1 \text{signe}(\beta_j).$$

Alors

$$\frac{\partial L(\beta_j)}{\partial \beta_j} = 0,$$

qui est équivalent à

$$(1 + \lambda_2)\beta_j = \mathbf{x}_j^\top \mathbf{y} - \frac{\lambda_1}{2} \text{signe}(\beta_j).$$

D'où

$$\hat{\beta}_j^{\,EN} = \frac{\mathbf{x}_j^\top \mathbf{y} - \frac{\lambda_1}{2} \text{signe}(\beta_j)}{(1 + \lambda_2)},$$

que l'on peut écrire sous la forme

$$\hat{\beta}_j^{\,EN} = \frac{S(\mathbf{x}_j^\top \mathbf{y}, \frac{\lambda_1}{2})}{1+\lambda_2}.$$

Avantages de la méthode Élastique Net :

- La méthode favorise la parcimonie ;
- La méthode peut sélectionner $p > n$ variables dans le modèle, au contraire de la méthode Lasso ;
- Elle tient compte de la corrélation entre les prédicteurs.

Dans la prochaine section, nous allons montrer l'importance de bien choisir les paramètres de régularisation pour les techniques de régularisation citées dans la section précédente. Nous allons également présenter l'outil qui permet de faire un tel choix.

2.3 La sélection des paramètres de régularisation

La sélection des paramètres de régularisation joue un rôle très important dans la performance des méthodes de régularisation. Ainsi, nous avons besoin d'une technique efficace pour choisir ces paramètres. Dans la littérature, la méthode la plus utilisée dans ce domaine est la validation croisée. C'est une méthode simple pour estimer les paramètres de régularisation λ.

2.3.1 La validation croisée

La validation croisée est un outil important dans l'application pratique d'un grand nombre d'approches de l'apprentissage statistique. Par exemple, elle peut être utilisée pour estimer l'erreur associée à une méthode d'apprentissage statistique donnée, pour évaluer la performance d'un modèle à l'échelle de la population des données. Ainsi, la méthode consiste à estimer les paramètres du modèle sur un

jeu de données appelé jeu de données d'apprentissage et valider la performance du modèle en calculant une statistique qui mesure les écarts entre les données observées et ce qui est prédit dans un jeu de données qui n'a pas été utilisé pour estimer les paramètres du modèle.

2.3.2 L'approche de l'ensemble de validation

Supposons que l'on a un échantillon d'apprentissage, et que l'on désire estimer l'erreur de test d'une méthode d'apprentissage statistique de façon appropriée. L'approche de l'ensemble de validation est une technique simple que l'on utilise pour atteindre un tel but. L'idée de cette méthode consiste à diviser l'ensemble sous étude arbitrairement en deux ensembles, un ensemble pour l'apprentissage, et un autre pour la validation. On utilise l'échantillon d'apprentissage pour ajuster le modèle et estimer les paramètres, et l'échantillon de validation pour choisir le bon modèle, c-à.d le modèle qui donne la plus petite valeur de l'erreur de validation MSE_v définie par

$$MSE_v = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

où (x_i, y_i) $\{1, 2, ..., n\}$ est l'ensemble de validation et $\hat{f}(x_i)$ est l'estimé de notre modèle qui est obtenu par le jeu de données d'apprentissage.

2.3.3 La validation croisée simple *LOOCV* (de l'anglais Leave One Out Cross Validation)

L'approche *LOOCV* se base également sur la division de l'ensemble de données sous étude en deux parties, un couple d'observation pour la validation, disons (x_1, y_1), et le reste de l'échantillon $(x_2, y_2), ..., (x_n, y_n)$, comme échantillon d'apprentissage. Notons que le couple (x_1, y_1) n'est pas utilisé pour l'ajustement du

modèle. On utilise x_1 et $\hat{f}$ pour trouver la valeur de $\hat{y}_1$. Ensuite, on calcule l'erreur de test pour ce couple d'observations comme suit :

$$MSE_1 = (y_1 - \hat{y}_1)^2.$$

On répète ce processus pour chaque couple (x_i, y_i), $i = 1, ..., n$. Pour chaque i on calcule une erreur

$$MSE_i = (y_i - \hat{y}_i)^2.$$

L'erreur de test pour la méthode est la moyenne des erreurs

$$CV = \frac{1}{n} \sum_{i=1}^{n} MSE_i.$$

Cette approche est avantageuse par rapport à l'approche de l'ensemble de validation, puisqu'elle utilise des échantillons de tailles grandes pour estimer les paramètres du modèle comparée à la méthode de l'ensemble de validation.

2.3.4 La validation croisée avec k groupes

Cette approche est une solution de replacement à l'approche précédente qui laisse un ensemble d'observation pour la validation et le reste pour l'échantillon d'apprentissage. La validation croisée avec k groupes consiste à diviser l'échantillon sous l'étude en k groupes de même taille. En effet, nous laissons un groupe pour la validation et les autres $k - 1$ groupes pour l'apprentissage. Par exemple, on prend le groupe 1 pour la validation, on ajuste le modèle pour les autres $k - 1$ groupes et on calcule l'erreur MSE_1. Ensuite, on refait le même processus pour les autres groupes en calculant les erreurs $MSE_2, MSE_3, ..., MSE_k$. L'estimé $CV_{(k)}$ est obtenu en prenant la moyenne des k erreurs

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

Dans l'appendice C, nous donnons un exemple d'application de la validation croisée, ainsi qu'une comparaison entre les différentes techniques de régularisation citées en haut, pour un jeu de données réelles sur le cancer de la prostate.

Un autre moyen très utilisé pour sélectionner le paramètre de régularisation des méthodes type-Lasso est les critères d'information *AIC* ou *BIC*, Yang (2005). L'avantage de cette approche est qu'elle est moins coûteuse en temps de calculs.

CHAPITRE III

LES MÉTHODES LINÉAIRES POUR LA DISCRIMINATION

Dans le cadre des méthodes linéaires pour la discrimination, prédire une variable réponse y discrète revient à lui attribuer une classe. Ainsi, souvent les méthodes utilisées pour la classification commencent d'abord par prédire la probabilité d'appartenir à chacune des classes.

Il existe plusieurs techniques de classification pour prédire une variable réponse discrète. Dans ce chapitre, nous étudierons trois méthodes les plus utilisées, à savoir, la régression logistique, l'analyse discriminante linéaire et les machines à vecteurs supports *SVM*.

3.1 Un aperçu sur la discrimination

Les problèmes de discrimination se produisent souvent. Voici quelques exemples.

- Une personne arrive à la salle d'urgence avec un ensemble de symptômes qui pourraient éventuellement êtres imputés à l'une des trois conditions médicales, à savoir, traitement immédiat (le patient ne peut pas attendre), urgent (le patient peut attendre une heure) et état normal (le patient peut attendre plus de temps). Laquelle des trois conditions convient à cette personne ?
- Un service bancaire en ligne doit être en mesure de déterminer si une transaction effectuée sur le site est frauduleuse, selon l'historique des transactions

de l'utilisateur.

- Sur la base de données d'*ADN* pour un certain nombre de patients avec ou sans une maladie donnée, un biologiste aimerait savoir lesquelles des mutations de l'*ADN* sont nuisibles (causant des maladies) et celles qui ne le sont pas.

Comme dans le cas de la régression, nous avons besoin d'un échantillon d'apprentissage pour construire un modèle de discrimination.

3.2 Pourquoi ne pas utiliser une régression linéaire ?

Nous avons souligné que la régression linéaire n'est pas appropriée dans le cas d'une réponse discrète. Mais, pourquoi pas ? Supposons que nous essayons de prédire l'état médical d'un patient dans la salle d'urgence sur la base de ses symptômes. Dans cet exemple simplifié, il existe trois diagnostiques possibles : accident vasculaire cérébral, sur-dose de médicament et crise épileptique. Il faut coder ces valeurs qualitatives en une variable y catégorielle

$$y = \begin{cases} 1 & \text{si crise épileptique,} \\ 2 & \text{si accident vasculaire cérébral,} \\ 3 & \text{si sur-dose de médicament.} \end{cases}$$

En utilisant ce codage, la méthode des moindres carrés peut être utilisée pour mettre en place un modèle pour la régression linéaire afin de prédire la variable y sur la base d'un ensemble de prédicteurs $\mathbf{x}_1$,...,$\mathbf{x}_p$. Ce codage peut également être converti en deux niveaux, en considérant que les deux variables, «crise épileptique» et «accident vasculaire cérébral» sont proches. Le codage devient ainsi

$$y = \begin{cases} 0 & \text{si accident vasculaire cérébral,} \\ 1 & \text{si sur-dose de médicament.} \end{cases}$$

On peut encore faire une régression linéaire sur cette variable binaire. On prédit la valeur de y_0 pour un individu en tenant compte des prédicteurs $\boldsymbol{X} = (\mathbf{x}_1, ..., \mathbf{x}_p)$, et on le classifie en se basant sur la règle de classification suivante :

- si $\hat{y_0} > 0.5$, on classe l'individu dans la catégorie «sur-dose de médicament»,
- si $\hat{y_0} < 0.5$, on classe l'individu dans la catégorie «accident vasculaire cérébral».

Dans ce cas, $\boldsymbol{X}^\top \hat{\boldsymbol{\beta}}$ obtenue en utilisant la régression linéaire est considérée comme proportionnelle à une valeur estimée de $Pr(\text{sur-dose de médicament}|\boldsymbol{X})$.

Mais, l'utilisation de la régression linéaire peut nous donner des valeurs estimées en dehors de l'intervalle $[0, 1]$, et on ne peut pas les interpréter comme des probabilités. Alors, on a besoin d'une méthode qui remédie à cet inconvénient. Une telle méthode est la régression logistique, entres autres.

3.3 La régression logistique

La régression logistique est une approche très connue que l'on utilise pour prédire une variable qualitative y avec deux niveaux ou plus. Elle est très utilisée dans plusieurs domaines, entre autres, santé, finance, économétrie. Dans un modèle de régression logistique, on considère y la variable qualitative à deux modalités $\{0, 1\}$, que l'on cherche à prédire à partir de l'observation $\mathbf{x}= (x_1, ..., x_p)^\top$. Dans le cas d'un seul prédicteur, on cherche à modéliser la relation entre $p(x) = p(y = 1|x)$ et x. Cette probabilité peut s'écrire sous la forme linéaire suivante

$$p(x) = \beta_0 + \beta_1 x.$$

On espère trouver des valeurs entre 0 et 1, mais malheureusement ce n'est pas le cas dans la plupart des problèmes. D'où l'idée de penser à une autre fonction qui va nous donner des estimations entre 0 et 1.

Remarque :

La discrimination bayésienne est fondé sur la règle de Bayes : $\hat{k} = \underset{k}{arg\,max}\, \mathbf{P}(y = k|x)$, dont la régression logistique est un cas particulier, Maclachlan, (2004).

La fonction de la régression logistique est définie par

$$p(x) = Pr(y = 1|x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}. \tag{3.1}$$

Pour ajuster ce modèle, on utilise la technique du maximum de vraisemblance que nous allons présenter dans la section suivante. Ainsi, nous avons

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}},$$

ou, de manière équivalente,

$$\frac{p(x)}{1 - p(x)} = e^{\beta_0+\beta_1 x}.$$

Le terme $\frac{p(x)}{1-p(x)}$, s'appelle la cote (le rapport de chance). Il prend ses valeurs entre 0 et $+\infty$. Si on applique la fonction log des deux côtés de l'équation (3.1), on obtient

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x.$$

On dit que la régression logistique a une fonction de lien logit qui est linéaire en x.

3.3.1 Estimation des coefficients du modèle logit

Les coefficients β_0 et β_1 sont inconnus, il faut les estimer à partir d'un échantillon d'apprentissage. Dans le cas de la régression linéaire, on utilise la méthode des moindres carrés pour estimer ces coefficients. Cependant, ici nous allons utiliser la méthode du maximum de vraisemblance. Soit $P(y|x)$ la distribution conditionnelle

de y étant donné x. On définit $P(y|x)$ par

$$P(y|x) = \begin{cases} p(x) & \text{si, } y = 1, \\ 1 - p(x) & \text{si, } y = 0, \end{cases}$$

avec $p(x)$ est définie dans (3.1). Ainsi, la fonction de vraisemblance peut s'écrire comme suit

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}.$$

La fonction log-vraisemblance est donnée ainsi par

$$\begin{aligned} \ell(\beta_0, \beta_1) &= \log L(\beta_0, \beta_1) \\ &= \log(\prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}) \\ &= \sum_{i=1}^{n} y_i \log(p(x_i)) + (1 - y_i)\log(1 - p(x_i)) \\ &= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i \log\left(\frac{p(x_i)}{1 - p(x_i)}\right) \\ &= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i(\beta_0 + x_i\beta_1) \\ &= -\sum_{i=1}^{n} \log(1 + e^{\beta_0 + x_i\beta_1}) + \sum_{i=1}^{n} y_i(\beta_0 + x_i\beta_1). \end{aligned}$$

Donc, pour trouver les estimateurs de β_0 et β_1, on va différencier la fonction log-vraisemblance par rapport à $\boldsymbol{\beta} = (\beta_0, \beta_1)$. Pour β_0, on obtient

$$\begin{aligned} \frac{\partial \ell(\beta_0, \beta_1)}{\partial \beta_0} &= -\sum_{i=1}^{n} \frac{e^{\beta_0 + x_i\beta_1}}{1 + e^{\beta_0 + x_i\beta_1}} + \sum_{i=1}^{n} y_i \\ &= \sum_{i=1}^{n} \{y_i - p(x_i)\}. \end{aligned}$$

Pour β_1, nous avons plutôt

$$\begin{aligned}\frac{\partial \ell(\beta_0, \beta_1)}{\partial(\beta_1)} &= -\sum_{i=1}^{n} \frac{x_i e^{\beta_0 + x_i\beta_1}}{1 + e^{\beta_0 + x_i\beta_1}} + \sum_{i=1}^{n} y_i x_i \\ &= \sum_{i=1}^{n} x_i(y_i - p(x_i)).\end{aligned}$$

Cependant, ces équations nous ne donnent pas une forme explicite pour $\hat{\boldsymbol{\beta}}$. Alors, il faut chercher à les résoudre numériquement. La méthode du score ou de Newton-Raphson peut être utilisée pour résoudre ce problème, (voir Minh *et al*, 2002). Dans l'appendice A nous donnons un exemple d'application de la régression logistique pour prédire la direction du marché d'un indice boursier.

3.4 L'analyse discriminante linéaire

La régression logistique consiste à modéliser directement la probabilité d'appartenir à une des deux classes étant donnée $\mathbf{x}$, en utilisant la fonction logistique.

L'analyse discriminante linéaire est une méthode de classement qui utilise les deux densités $f_0(\mathbf{x}) = f(\mathbf{x}|y = 0)$ et $f_1(\mathbf{x}) = f(\mathbf{x}|y = 1)$ pour calculer la probabilité d'appartenir à chaque classe à l'aide du théorème de Bayes. En effet, soit π_0 et π_1 les probabilités d'appartenir aux deux classes 0 et 1 respectivement, tel que $\pi_0 + \pi_1 = 1$. En appliquant le théorème de Bayes, on obtient

$$P(y = 0|x) = \frac{f_0(x)\pi_0}{f_0(x)\pi_0 + f_1(x)\pi_1}$$

et

$$P(y = 1|x) = \frac{f_1(x)\pi_1}{f_0(x)\pi_0 + f_1(x)\pi_1}$$

avec $f_0(x)$ la fonction de densité pour le groupe 1 et $f_1(x)$ la densité pour le groupe 2. La règle de décision consiste à assigner y au groupe 1 si

$$\pi_0 f_0(\mathbf{x}) > \pi_1 f_1(\mathbf{x}).$$

En particulier, si $\mathbf{x}|y = 0 \sim N_p(\mu_0, \Sigma)$ et $\mathbf{x}|y \sim N_p(\mu_1, \Sigma)$, alors nous pouvons montrer que la règle de décision est linéaire en x.

L'analyse discriminante linéaire est une méthode robuste face à l'hypothèse de normalité pour des dimensions modérées. C'est aussi une méthode que l'on utilise comme séparateur linéaire surtout dans le cas de données linéairement séparables.

La définition suivante d'un hyperplan va nous aider à comprendre d'avantage la prochaine section.

Définition 3.4.1. *Soit V un espace vectoriel de dimension p. On dit que $A \subset V$ est un sous espace affine de V si $D = \{x - y : x, y \in A\}$ est un sous-espace vectoriel de V. La dimension de A est définie comme la dimension de D. Un hyperplan H est un espace affine de dimension $p - 1$.*

Dans un espace vectoriel de dimension $p = 3$, les hyperplans affines sont des espaces affines de dimensions $p - 1 = 2$, donc c'est un plan d'équation

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 = 0,$$

en x_1, x_2, x_3. Dans un espace vectoriel de dimension $p = 2$, les hyperplans affines sont des droites.

3.5 Les machines à vecteurs supports

Les machines à vecteurs supports sont des techniques très populaires de classification connues comme des méthodes de séparation à vaste marges. On utilise la méthode *SVM* pour résoudre des problèmes de classification (ou discrimination).

3.5.1 Les séparateurs linéaires pour les données linéairement séparables

Dans cette section, nous allons parler des séparateurs linéaires. Un séparateur linéaire est défini comme une fonction linéaire des prédicteurs dont le signe fournira la classe de l'observation à la prévision.

La méthode des moindres carrés ordinaire (MCO)

Supposons que $p = 2$ et $\{(x_{i1}, y_1), (x_{i2}, y_2), i = 1, ..., n\}$ un ensemble d'apprentissage avec $y_i = \{1, -1\}$. Si nous ajustons un modèle de régression multiple standard en $\mathbf{y}$ et $\mathbf{x}$ comme dans la section (2.1), l'hyperplan séparateur issu de ce modèle satisfait

$$\{\mathbf{x} : \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 = 0\}, \tag{3.2}$$

où $\hat{\beta}_0$,$\hat{\beta}_1$ et $\hat{\beta}_2$ sont les estimés MCO de β_0, β_1 et β_2 respectivement. Cet hyperplan basé sur les moindres carrés peut donner lieu à des mauvaises affectations. La figure 3.1 dans Freidman *et al.*(2001) illustre un exemple où un tel problème peut survenir.

Dans le prochain paragraphe, nous allons définir la distance d'un point à un hyperplan. Une telle distance va nous servir pour mieux comprendre le fonctionnement des séparateurs *SVM*.

Propriété 3.5.1. *On considère l'hyperplan H qui est défini pour $\boldsymbol{x} \in H$ par l'équation*

$$f(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x} = 0. \tag{3.3}$$

(a) *Tout point $\boldsymbol{x}_0 \in H$ satisfait $\boldsymbol{\beta}^\top \boldsymbol{x}_0 = -\beta_0$.*

(b) *La distance d'un point $\boldsymbol{x}$ à l'hyperplan H est donnée par*

$$d(\boldsymbol{x}, H) = \frac{|\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}|}{||\boldsymbol{\beta}||} = \frac{|f(\boldsymbol{x})|}{||f'(\boldsymbol{x})||},$$

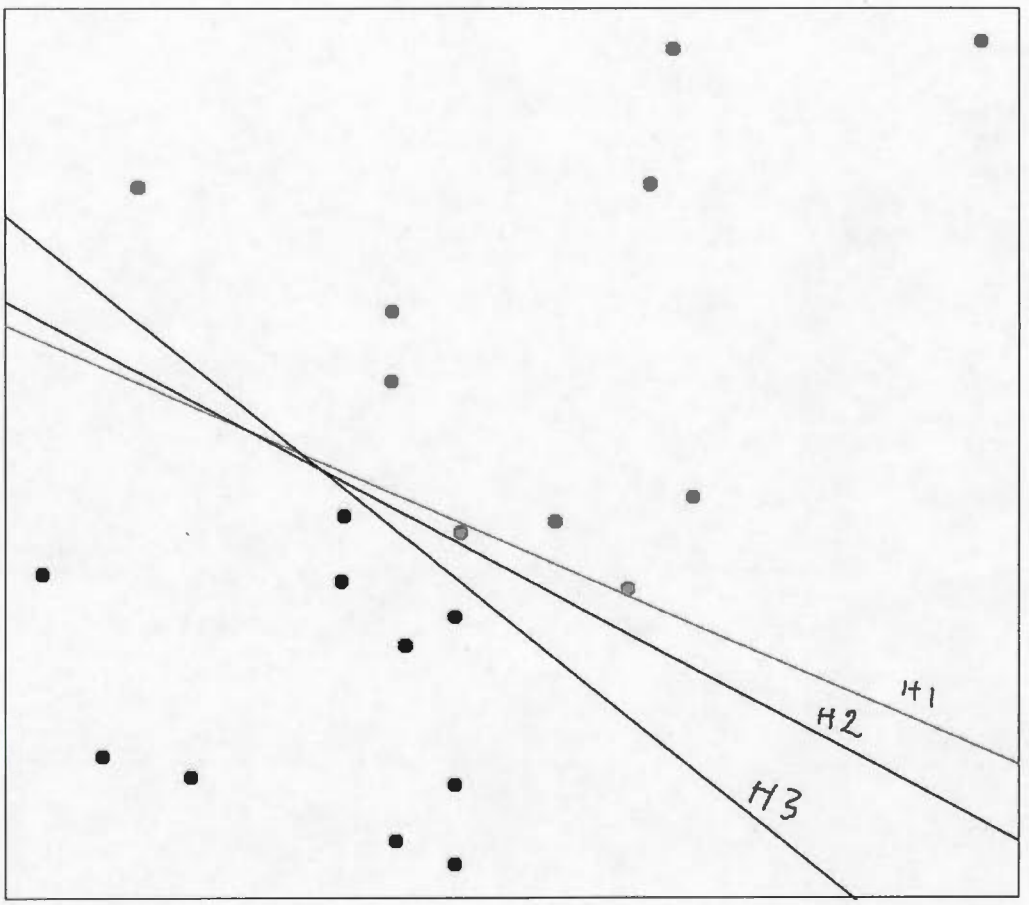

Figure 3.1: Un exemple pour des données linéairement séparables par un hyperplan. La droite H_1 présente la solution de la méthode MCO et les deux autres droites H_2 et H_3 sont deux solutions différentes données par l'algorithme d'apprentissage de perceptron.

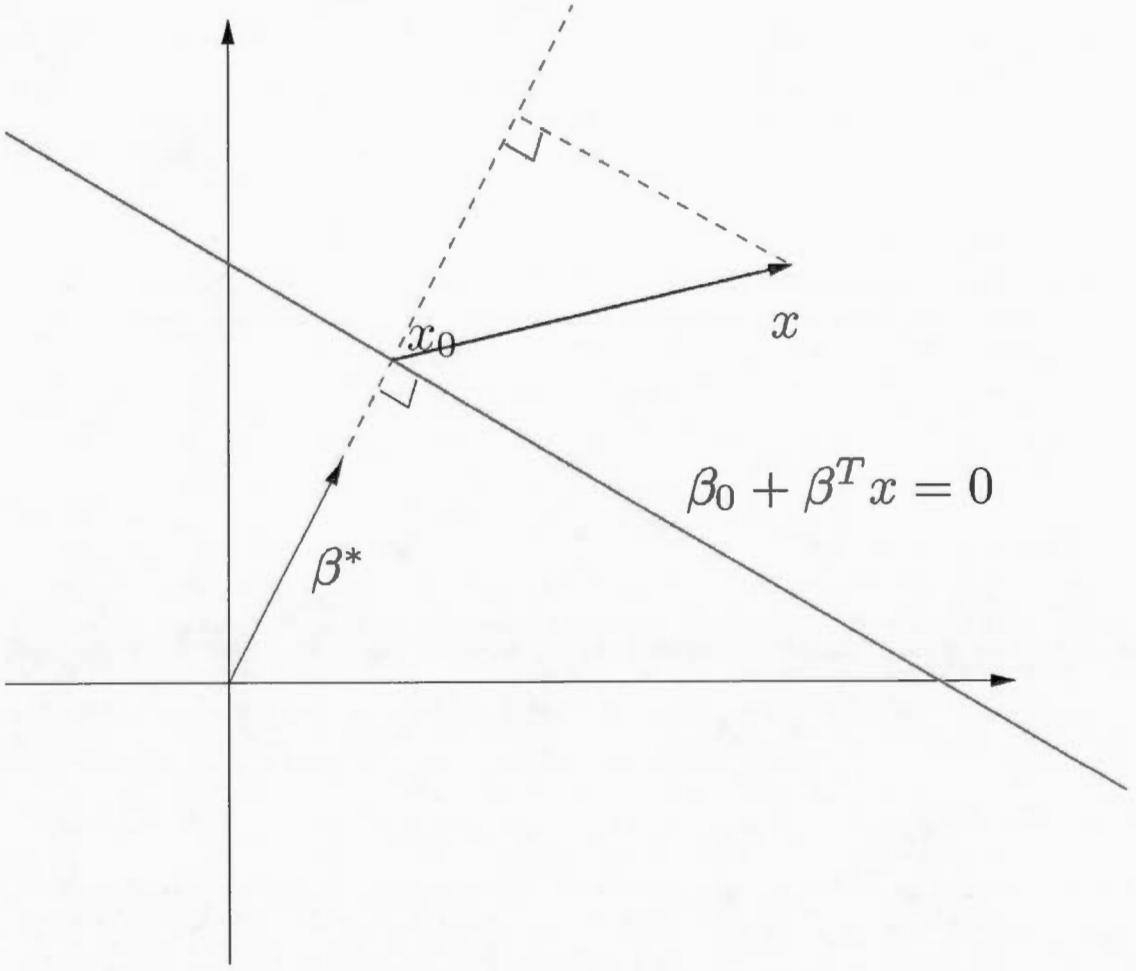

Figure 3.2: Illustration géométrique d'un hyperplan.

et donc $|d(x, H| \propto |f(x)|$.

(c) Le vecteur $\beta^* = \frac{\beta}{||\beta||}$ *est orthogonal à* H.

La distance dans (**b**) est le résultat de $< x - x_0, \frac{\beta}{||\beta||} >$.

Pour démontrer (**c**), il suffit de prendre deux points $\boldsymbol{x}_1$ et $\boldsymbol{x}_2$ de H pour avoir

$$\beta^\top(\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0.$$

Dans $\mathbb{R}^2$, l'équation (3.3) est une droite. La figure 3.2 (Freidman *et al.*, 2009), donne un exemple d'un hyperplan et de la distance d'un point **x** par rapport à cet hyperplan.

L'algorithme d'apprentissage de Perceptron

Cet algorithme cherche l'hyperplan qui sépare les données en deux classes en minimisant la distance entre les points mal classés et la frontière de décision qui est définie par (3.3). La quantité à minimiser dans le cas ou $p = 2$ est

$$D(\beta_0, \beta_1, \beta_2) = -\sum_{j \in EM} y_j(\beta_0 + \beta_1 x_1 + \beta_2 x_2),$$

avec EM désigne l'ensemble des points mal classés. L'algorithme est basé sur la technique de la descente du gradient. En effet, on initialise l'algorithme par un hyperplan quelconque afin de calculer $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$. Ensuite, on fait la mise à jour suivante

$$\hat{\beta}_j \leftarrow \hat{\beta}_j - \frac{\partial D(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)}{\partial \hat{\beta}_j},$$

pour $j = 0, 1, 2$. On répète le processus jusqu'à convergence. Cet algorithme est sensible aux valeurs de départ et à chaque fois il donne un hyperplan différent.

L'hyperplan de séparation optimal

Nous avons vu que la solution de l'algorithme d'apprentissage de perceptron n'est pas unique et elle dépend de l'état initial. Ainsi, l'objectif de cette section est de

chercher un hyperplan optimal. En effet, l'hyperplan de séparation optimal est la solution du problème d'optimisation suivant

$$\max_{\beta,\beta_0,||\beta||=1} M$$

$$\textbf{sous contrainte } y_i(\mathbf{x}_{i.}^\top \beta + \beta_0) \geq M, i = 1, ..., n. \tag{3.4}$$

Si on pose $M = \frac{1}{||\beta||}$, le problème de l'équation (3.4) devient

$$\min_{\beta,\beta_0,||\beta||=1} \frac{1}{2}||\beta||^2$$

$$\textbf{sous contrainte } y_i(\mathbf{x}_{i.}^\top \beta + \beta_0) \geq 1, i = 1, ..., n. \tag{3.5}$$

Le problème (3.5) est convexe. Alors, le lagrangien est donné par

$$L(\beta, \beta_0, \alpha) = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{n} \alpha_i \{y_i(\beta^\top \mathbf{x}_{i.} + \beta_0) - 1\}. \tag{3.6}$$

Ainsi, nous allons calculer les dérivées partielles du Lagrangien définit dans (3.6) comme suit

$$\frac{\partial L(\beta, \beta_0, \alpha)}{\partial \beta} = 2\frac{1}{2}\beta - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i.}, = \beta - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i.}.$$

$$\frac{\partial L(\beta, \beta_0, \alpha)}{\partial \beta_0} = 0 - \sum_{i=1}^{n} \alpha_i y_i.$$

En posant les deux dérivées égales à 0, on obtient

$$\beta = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i.}, \tag{3.7}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{3.8}$$

On remplace ces deux quantités dans l'équation (3.6), on obtient

$$\tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_{i.}^\top \mathbf{x}_{j.}$$

$$\textbf{sous contrainte } \alpha_j \geq 0. \tag{3.9}$$

La solution du problème (3.5) doit satisfaire les conditions de Karush-Kuhn-Tucker, à savoir (3.7),(3.8),(3.9) et

$$\alpha_i\{y_i(\mathbf{x}_{i.}^\top \boldsymbol{\beta} + \beta_0)\} = 0 \quad \forall i = 1, ..., n. \tag{3.10}$$

L'hyperplan de séparation optimal est donné par l'équation

$$\hat{f}(\mathbf{x}) = \mathbf{x}^\top \hat{\boldsymbol{\beta}} + \hat{\beta}_0. \tag{3.11}$$

Ainsi, on affecte $\mathbf{x}$ à la classe 1 si $\hat{f}(\mathbf{x}) > 0$ et on l'affecte à la classe -1 si $\hat{f}(\mathbf{x}) < 0$. Nous pouvons traduire cette règle de classification comme suit

$$\hat{\text{classe}}(\mathbf{x}) = \text{signe}(\hat{f}\{\mathbf{x}\}).$$

3.6 Le séparateur à vecteurs supports

Nous avons vu dans la section précédente la détermination de l'hyperplan optimal qui sépare des données linéairement séparables en deux classes disjointes. Nous allons généraliser cette théorie pour des données qui ne sont pas linéairement séparables. Supposons que $p = 2$ et $y_i \in \{-1, 1\}$. On définit l'hyperplan $H : \{\mathbf{x} : \mathbf{x}^\top \boldsymbol{\beta} + \beta_0 = 0\}$, tel que $\boldsymbol{\beta} = (\beta_1, \beta_2)^\top$ et $||\boldsymbol{\beta}|| = 1$. La règle de décision est donnée par

$$\hat{\text{classe}}(\mathbf{x}) = \text{signe}(\mathbf{x}^\top \boldsymbol{\beta} + \beta_0).$$

Pour les données linéairement séparables, le problème d'optimisation est donné par (3.5). Pour les données qui ne sont pas linéairement séparables, on maximise M, cependant, on va permettre à certains points d'être mal classés. On considère $\boldsymbol{\xi} = (\xi_1, ..., \xi_n)^\top$ qui sont associées aux observations $(\mathbf{x}_{i.}, y_i)$, $i = 1, ..., n$. Nous pouvons définir les écarts ξ_i de la façon suivante : l'observation $(\mathbf{x}_{i.}, y_i)$ est bien classée si $y_i f(\mathbf{x}_{i.}) \geq 1$ et dans ce cas on a $\xi_i = 0$, l'observation $(\mathbf{x}_{i.}, y_i)$ est mal classée si $y_i f(\mathbf{x}_{i.}) < 1$ et dans ce cas on a $\xi_i = 1 - y_i f(\mathbf{x}_{i.})$.

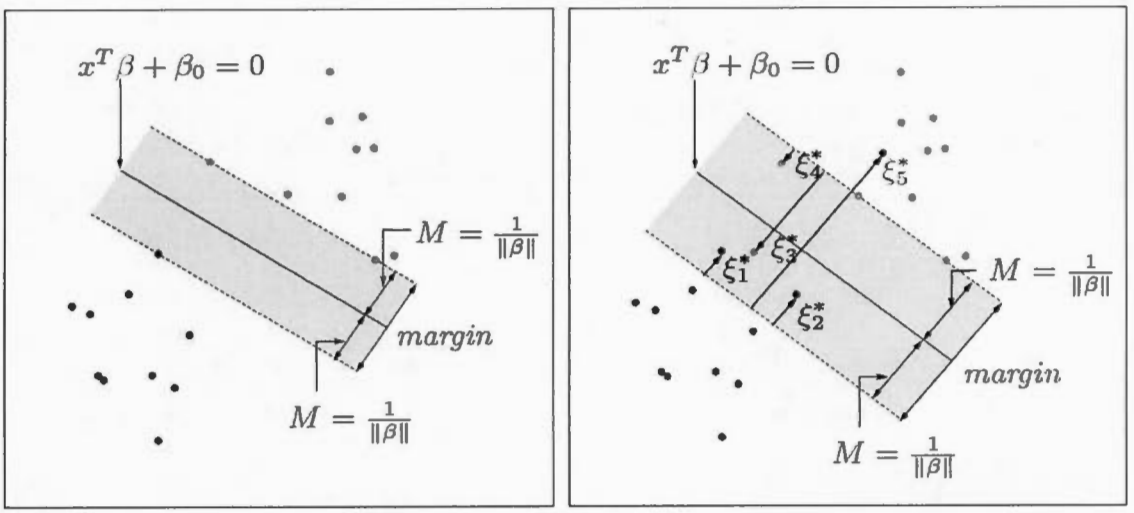

Figure 3.3: L'illustration est divisée en deux parties. La partie à gauche présente le cas des données linéairement séparables et la partie à droite illustre des données non linéairement séparables. Les points qui sont mal classés sont liés par des quantités ξ_i^* qui désignent les distances entre ces points et leurs classes.

La figure 3.3, tirée de Freidman *et al.*, (2001), illustre le cas des données non linéairement séparables. Le problème d'optimisation dans ce cas est donné par l'équation

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{j=1}^{n} \xi_i,$$

$$\textbf{sous contraintes } \xi_i \geq 0, \quad y_i(\mathbf{x}_{i.}^\top \beta + \beta_0) \geq 1 - \xi_i, \forall i = 1, ..., n, \tag{3.12}$$

avec C est une constante positive et $C\xi_i$ est la quantité par laquelle $f(\mathbf{x}_{i.})$ se situe au mauvais coté de la marge maximale. Le lagrangien est donné par

$$L = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i[y_i(\mathbf{x}_{i.}^\top \beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^{n} \mu_i \xi_i.$$

Nous allons minimiser L par rapport à β, β_0 et $\boldsymbol{\xi}$. La dérivée partielle de L par

rapport à $\boldsymbol{\beta}$ est donnée par

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} + 0 - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i.}^{\top} - 0,$$

ce qui implique que

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i.}^{\top}.$$

En posant égal à zéro, on obtient

$$\boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i.}^{\top}.$$

Ensuite, on calcule la dérivée partielle de L par rapport à β_0 comme suit

$$\frac{\partial L}{\partial \beta_0} = 0 + 0 - \sum_{i=1}^{n} \alpha_i y_i - 0.$$

En posant cette dérivée égal à zéro, on obtient $\sum_{i=1}^{n} \alpha_i y_i = 0$. Finalement, la dérivée partielle de L par rapport à $\boldsymbol{\xi}$ est donnée par

$$\frac{\partial L}{\partial \xi_i} = 0 + C - \alpha_i - \mu_i.$$

En posant cette dérivée égal à zéro, on trouve $\alpha_i = C - \mu_i$, $\forall i$. En remplaçant les quantités trouvées par leurs valeurs dans l'expression de L, on obtient

$$\tilde{L} = \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i.}^{\top} (\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i.}) - \sum_{i=1}^{n} \alpha_i y_i \beta_0 + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} \mu_i \xi_i.$$

On peut simplifier pour obtenir

$$\tilde{L} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\top} \mathbf{x}_j.$$

On minimise $\tilde{L}$ afin d'estimer α_i, ensuite, on utilise $\hat{\alpha}_i$ pour calculer $\hat{\boldsymbol{\beta}}$ à l'aide de l'équation

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{n} \hat{\boldsymbol{\alpha}}_j y_i \mathbf{x}_{i.}.$$

Dans la prochaine section, nous allons voir la méthode SVM comme une méthode de régularisation.

3.7 La méthode *SVM* vue comme méthode de régularisation (*SVM*-ℓ_2)

Le problème d'optimisation défini dans l'équation (3.12) a une forme équivalente, Freidman et al (2001), qui s'écrit comme suit :

$$\min_{\beta,\beta_0} \frac{1}{n}\sum_{i=1}^{n}[1-y_i(\beta_0+\mathbf{x}_{i.}^{\top}\beta)]_+ + \frac{\lambda}{2}||\beta||^2. \tag{3.13}$$

En effet, les contraintes dans l'équation (3.12) :

$$\xi_i \geq 0, \quad y_i(\beta_0+\mathbf{x}_{i.}^{\top}\beta) \geq 1-\xi_i, \forall i$$

peuvent s'écrire de manière équivalente comme suit :

$$\xi_i \geq [1-y_i(\beta_0+\mathbf{x}_{i.}^{\top}\beta)]_+, \forall i,$$

avec la fonction de perte $L(t)=[1-t]_+$ définie par

$$[1-t]_+ = \begin{cases} 0, & t \geq 1 \\ 1-t, & t \leq 1. \end{cases}$$

Soit $\lambda = \frac{1}{nC}$. Ainsi, nous avons

$$\begin{aligned} \arg\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i &= \arg\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + \frac{1}{n\lambda}\sum_{i=1}^{n}\xi_i \\ &= \arg\min_{\beta,\beta_0} \frac{\lambda}{2}||\beta||^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ &= \arg\min_{\beta,\beta_0} \frac{\lambda}{2}||\beta||^2 + \frac{1}{n}\sum_{i=1}^{n}[1-y_i(\beta_0+\mathbf{x}_{i.}^{\top}\beta)]_+. \end{aligned}$$

Malgré sa performance, la méthode *SVM*-ℓ_2 souffre du problème de rétrécissement des coefficients β comme la méthode *Ridge* vue dans le chapitre 2 et elle utilise tous les prédicteurs pour faire la discrimination. Ainsi, elle ne fournit pas un modèle parcimonieux. Un tel comportement est un grand inconvénient dans le

cas de données de grandes dimensions où l'objectif est de réduire les variables de bruit. Bradley et Mangasarian (1998) ont proposé une autre version de la méthode *SVM* en utilisant la pénalité Lasso vue dans le chapitre 2, au lieu de la pénalité L_2.

3.8 La méthode *SVM* vue comme méthode de régularisation avec pénalité de type ℓ_1

Les méthodes classiques de sélection , pas à pas ou celles qui utilisent une validation croisée sont instables (Breiman, 1995). La pénalité de type ℓ_1, comme *Lasso*, permet d'estimer et de sélectionner les variables de manière simultanée. Le problème d'optimisation est défini par

$$\min_{\beta,\beta_0} \frac{\lambda}{2}||\beta||_1 + \frac{1}{n}\sum_{i=1}^{n}[1 - y_i(\beta_0 + \mathbf{x}_{i.}^{\top}\beta)]_+. \tag{3.14}$$

Comme dans le cas de la régression linéaire avec la pénalité *Lasso*, la méthode *SVM*-ℓ_1 a tendance à rétrécir beaucoup les coefficients β. Alors, pour remédier aux problèmes des méthodes *SVM*-ℓ_1 et *SVM*-ℓ_2, Wang *et al.*(2006) ont proposé la méthode *DrSVM* (de l'anglais Doubly regularized Support Vector Machine) qui correspond à la méthode *SVM* avec la pénalité Elastic Net.

Ensuite, Wang *et al.*(2008) ont proposé la méthode HHSVM-EN en minimisant une fonction d'Huber qui possède des propriétés intéressantes. Yang et Zou (2013) ont proposé une généralisation de la méthode de descente par coordonnée afin d'accélérer l'algorithme l'algorithme proposée par Wang *et al.* (2008). C'est l'approche de Yang et Zou (2013) qui nous intéresse d'avantage dans ce mémoire. Ainsi, elle sera présentée en détails dans les premières sections du chapitre 4.

3.9 La discrimination et la validation croisée

Dans le chapitre 2, nous avons introduit la notion de la validation croisée pour la régression linéaire pour une variable réponse quantitative. Nous avons utilisé le critère MSE pour donner une valeur quantitative à l'erreur de test. Dans le cas de classification, nous pouvons utiliser la validation croisée pour calculer l'erreur de classification. En effet, pour la méthode $LOOCV$, l'erreur de classement est donnée par la formule explicite

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} ERR_i,$$

avec

$$ERR_i = I(y_i \neq \hat{y}_i).$$

CHAPITRE IV

UN ALGORITHME EFFICACE POUR LE CALCUL DES CHEMINS DE SOLUTIONS DES COEFFICIENTS DE LA MÉTHODE HHSVM-CEN EN GRANDE DIMENSION

4.1 Introduction

Yang et Zou (2013) proposent un algorithme rapide pour calculer les chemins de solutions des coefficients de la méthode HHSVM avec la pénalité EN, en appliquant la méthode de la descente par coordonnée. La méthode HHSVM-EN est une extension de la méthode *SVM* qui tente de minimiser une fonction de perte, ϕ_c, possèdant des propriétés intéressantes afin d'accélérer la convergence de l'algorithme d'estimation des paramètres du modèle $SVM\text{-}\ell_1 + \ell_2$. Cependant, pour certaines données où une structure inconnue de groupes existe, cette méthode rétrécit tous les coefficients des prédicteurs vers zéro et elle ne détecte pas une telle structure.

Witten *et al.*(2014) proposent de remplacer la pénalité EN par la pénalité CEN comme nouvelle méthode de régularisation, dans le cadre des modèles de régression avec variables réponses quantitatives en présence de données de grandes dimensions. Cette méthode de régularisation tient compte de la structure de groupes des prédicteurs ainsi que de l'association de ces derniers avec la variable réponse.

Ainsi, notre objectif dans ce projet est d'adapter la pénalité CEN à la méthode HHSVM. Spécifiquement, nous voulons présenter un nouvel algorithme pour calculer les chemins de solutions des coefficients de la méthode HHSVM avec la pénalité CEN. Le grand défi pour la méthode HHSVM est l'application de la descente par coordonnée, car la fonction de perte ϕ_c n'admet pas une dérivée première partout. Alors, pour surmonter ce problème, nous allons utiliser une technique de majoration-minimisation de la fonction objective.

D'abord, nous allons présenter la méthode CEN dans le cas de la régression linéaire. Ensuite, nous allons présenter la méthode HHSVM avec la pénalité EN pour des problèmes de classification pour mettre le lecteur dans le contexte de notre contribution. Dans la prochaine section, nous allons définir un outil très pratique dans la littérature d'optimisation, à savoir la descente par coordonnée.

4.2 La descente par coordonnée standard (Tseng, 2001)

La descente par coordonnée est une méthode très populaire et très utile pour minimiser une fonction objective (fonction de perte) f à plusieurs variables. En effet, à chaque itération on minimise f par rapport à une coordonnée, en gardant les autres coordonnées fixes.

Le lemme suivant donne des conditions à satisfaire par une fonction à plusieurs variables pour qu'elle admette un minimum global en x.

Lemme 4.2.1. *Soit $f : \mathbb{R}^p \mapsto \mathbb{R}$, une fonction convexe et dérivable. S'il existe un point $\boldsymbol{x} \in \mathbb{R}^p$, tel que f est minimisée en chaque coordonnée de $\boldsymbol{x}$, alors f admet un minimum global en x.*

Preuve. Nous savons d'après l'énoncé que

$$\forall i = 1, ..., p : \frac{\partial f}{\partial x_i}(\mathbf{x}) = 0.$$

Alors, nous avons

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), ..., \frac{\partial f}{\partial x_p}(\mathbf{x})\right) = 0.$$

Puisque f est convexe, $\mathbf{x}$ est un minimum global. D'où le résultat. □

Remarque : Dans le cas où f est une fonction convexe non dérivable, f n'admet pas de minimum global. Voici un contre exemple, la fonction $f(x,y) = x^2 + y^2 + 2\lambda|x-y|, \quad (x,y) \in [-\lambda,\lambda]^2$ admet 0 comme minimum pour l'axe des abscisses et pour l'axe des ordonnées. Cependant, le point $(0,0)$ n'est pas un minimum global pour la fonction f.

Lemme 4.2.2. *Soit $f : \mathbb{R}^p \mapsto \mathbb{R}$, qui peut s'écrire de la façon suivante*

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + \sum_{i=1}^{p} h_i(x_i),$$

où g est une fonction convexe et dérivable, et h_i, $i = 1, ..., p$ sont des fonctions convexes.

S'il existe un point $\boldsymbol{x} \in \mathbf{R}^p$ tel que f est minimisée en chaque coordonnée de $\boldsymbol{x}$, alors f admet $\boldsymbol{x}$ comme minimum global.

Preuve. On cherche à montrer que $\mathbf{x} \in \mathbb{R}^p$ est un minimum global pour f sachant que les entrées de $\mathbf{x}$ sont les minimums de f pour chacune des coordonnées. Il faut montrer que

$$\forall \mathbf{z} \in \mathbb{R}^p : f(\mathbf{z}) \geq f(\mathbf{x}).$$

En effet,

$$f(\mathbf{z}) - f(\mathbf{x}) \;=\; g(\mathbf{z}) - g(\mathbf{x}) + \sum_{i=1}^{p} [h_i(z_i) - h_i(x_i)].$$

D'abord, nous allons montrer que

$$g(\mathbf{z}) - g(\mathbf{x}) \;\geq\; \nabla g(\mathbf{x})^\top (\mathbf{z} - \mathbf{x}).$$

En effet, dans le cas $p = 1$, puisque g est convexe, alors pour tout $0 \leq t \leq 1$, nous avons

$$g(tz + (1-t)x) \leq tg(z) + (1-t)g(x).$$

Ce qui implique

$$\frac{g\{tz + (1-t)x\} - g(x)}{t} \leq g(z) - g(x).$$

Puisque g est dérivable, on fait tendre t vers zéro pour avoir

$$g(z) - g(x) \geq \nabla g(x)^{\top}(z - x). \tag{4.1}$$

Pour le cas général, nous allons poser

$$G(t) = g(t\mathbf{z} + (1-t)\mathbf{x}).$$

Puisque g est convexe et dérivable, alors G est convexe et dérivable. Ensuite, nous pouvons écrire

$$G(1) \geq G(0) + G'(0).$$

En effet, nous avons appliqué l'équation 4.1 pour la fonction G entre les points 0 et 1. Ce qui équivaut à

$$g(\mathbf{z}) - g(\mathbf{x}) \geq \nabla g(\mathbf{x})^{\top}(\mathbf{z} - \mathbf{x}).$$

Alors, nous pouvons écrire

$$\begin{aligned} f(\mathbf{z}) - f(\mathbf{x}) &\geq \nabla g(\mathbf{x})^\top (\mathbf{z} - \mathbf{x}) + \sum_{i=1}^{p} [h_i(z_i) - h_i(x_i)] \\ &= \sum_{i=1}^{p} [\nabla_i g(\mathbf{x})(z_i - x_i) + h_i(z_i) - h_i(x_i)] \\ &\geq \sum_{i=1}^{p} [\nabla_i g(\mathbf{x})(z_i - x_i) + h_i'(x_i, d)], \text{où} \quad d = z_i - x_i \\ &\geq 0. \end{aligned}$$

□

L'algorithme de la descente par coordonnée (Tseng, 2001) nous montre comment calculer le minimum global d'une fonction dans le contexte du lemme 4.2.2.

Algorithme pour la descente par coordonnée

Si $f : \mathbb{R}^p \mapsto \mathbb{R}$, peut s'écrire sous la forme suivante

$$f(\mathbf{x}) = g(\mathbf{x}) + \sum_{i=1}^{p} h_i(x_i),$$

avec g est une fonction convexe et dérivable et $h_i, i = 1, ..., p$ sont des fonctions convexes. On peut utiliser la descente par coordonnée pour trouver un minimum global pour f.

L'algorithme procède de la façon suivante :

* On commence par $\mathbf{x}^{(0)}$.
* On répète pour tout $k = 1, 2, 3, ...$

$$x_1^{(k)} \in \arg\min_{x_1} f(x_1, x_2^{(k-1)}, x_3^{(k-1)}, .., x_p^{(k-1)})$$

$$x_2^{(k)} \in \arg\min_{x_2} f(x_1^{(k)}, x_2, x_3^{(k-1)}, .., x_p^{(k-1)})$$

$$x_3^{(k)} \in \arg\min_{x_3} f(x_1^{(k)}, x_2^{(k)}, x_3, .., x_p^{(k-1)})$$

...

$$x_p^{(k)} \in \arg\min_{x_p} f(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, .., x_p)$$

jusqu'à convergence, c.-à-d. jusqu'à

$$x_j^{(k)} \simeq x_j^{(k-1)}, \quad j = 1, ..., p.$$

Notons que les minimisations dans le dernier algorithme sont atteintes si les conditions suivantes sont satisfaites :

- $\mathbf{x}^0 = \{x : f(x) \leq f(x^0)\}$ est compact. On dit qu'une partie E de $\mathbb{R}$ est compacte si E est un ensemble fermé et borné ;
- f est continue.

La proposition suivante (Tseng, 2001), garantit la convergence de la technique de la descente par coordonnée lorsque les deux conditions citées ci-dessus sont satisfaites.

Proposition 4.2.1. *Si f est continue sur un compact $\{x : f(x) \leq f(x^{(0)})\}$ et f atteint son minimum, alors toute limite du point $x^{(k)}$, $k = 1, 2, 3, ..,$ est un minimum de f.*

Autrement dit, la proposition 4.2.1 nous dit que toute séquence générée par l'algorithme de la descente par coordonnée converge vers un point limite $\mathbf{x}$ qui est un point stationnaire pour f (i.e. un minimum global).

Dans la section suivante nous allons présenter la méthode CEN dans le cadre de la régression linéaire en grandes dimensions.

4.3 La méthode CEN pour la régression linéaire en grandes dimensions

On considère une matrice $\boldsymbol{X}$ de prédicteurs (*i.e.* les colonnes $\mathbf{x}_j \in \mathbf{R}^p$), de dimension $n \times p$, et $\mathbf{y} \in R^n$ est la variable réponse. Nous cherchons à prédire $\mathbf{y}$ avec la matrice $\boldsymbol{X}$ selon le modèle (2.1). On suppose que $\mathbf{y}$ est centrée et on rappelle que $\boldsymbol{X}$ est centrée et réduite.

En présence de grandes dimensions $p >> n$, la matrice $\boldsymbol{X}$ est singulière. Dans le cas de la régression linéaire, l'utilisation de la méthode des moindres carrés en présence de grandes dimensions est limitée. Nous avons vu dans le chapitre 2, que les méthodes de régularisation peuvent remédier à ce problème. La méthode CEN, proposée par Witten *et al.*(2014), est une méthode de régularisation qui utilise la corrélation entre les prédicteurs et leurs associations avec la variable réponse pour les classifier davantage dans des groupes afin d'améliorer l'ajustement du modèle. Un tel problème d'optimisation avec la méthode CEN peut être présenté comme suit :

On suppose dans le modèle qu'il y a K groupes (ou *clusters*) $C_1, ..., C_K$ inconnus dans les prédicteurs. Le vecteur $\boldsymbol{\beta}$ associé aux prédicteurs dans le modèle (2.1) est la solution du problème d'optimisation suivant

$$\min_{C_1,...,C_K,\boldsymbol{\beta}} \{||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda_1||\boldsymbol{\beta}||_1 + \frac{\lambda_2}{2}\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{j,l\in C_k, l\neq j}||\mathbf{x}_j\beta_j - \mathbf{x}_l\beta_l||^2\}, \quad (4.2)$$

où $|C_k|$ désigne le cardinal de l'ensemble C_k.

La pénalité CEN est définie comme suit

$$p_{CEN}(C,\boldsymbol{\beta}) = \lambda_1||\boldsymbol{\beta}||_1 + \frac{\lambda_2}{2}\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{j,l\in C_k}||\mathbf{x}_j\beta_j - \mathbf{x}_l\beta_l||^2, \quad (4.3)$$

où λ_1 et λ_2 sont des paramètres de régularisation positifs. Les ensembles $C_1, ..., C_K$ désignent les groupes des p prédicteurs, avec $C_k \cap C_l = \varnothing$ si $k \neq l$ et $C_1 \cup C_2 \cup ... \cup$

$C_k = \{1, ..., p\}$. Cette pénalité peut se transformer en une forme qui va faciliter les calculs un peu plus tard. Ainsi, nous pouvons écrire (4.3) sous la forme suivante :

$$p_{CEN}(C, \beta) = \lambda_1 ||\beta||_1 + \lambda_2 \sum_{k=1}^{K} \sum_{j \in C_k} ||\mathbf{x}_j \beta_j - \frac{1}{|C_k|} \sum_{l \in C_k} \mathbf{x}_l \beta_l||^2.$$

Le problème d'optimisation avec la méthode CEN est non convexe. Ainsi, pour le résoudre, il faut le séparer en deux parties :

- On fixe les groupes $C_1, ..., C_K$, et on résout le problème (4.2) par rapport à $\boldsymbol{\beta}$.
- On fixe $\boldsymbol{\beta}$, et on minimise le deuxième terme de la partie à droite de l'équation (4.3) par rapport à $C_1, ..., C_K$.

Ainsi, un algorithme pour résoudre ce problème peut s'écrire comme suit :

Algorithme pour résoudre le problème d'optimisation avec la méthode CEN

1. On initialise $\boldsymbol{\beta}$ par la solution du problème d'optimisation

$$\min_{\boldsymbol{\beta}} \{||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda_1 ||\boldsymbol{\beta}||_1 + \lambda_2 ||\boldsymbol{\beta}||^2\}.$$

2. On itère jusqu'à convergence les étapes (2a) et (2b) :
   a) On fixe $\boldsymbol{\beta}$ et on minimise par rapport à $C_1, ..., C_K$. Cela veut dire

$$\min_{C_1, ..., C_K} \{\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{j,l \in C_k} ||\mathbf{x}_j \beta_j - \mathbf{x}_l \beta_l||^2\}.$$

   On obtient un optimum local.

   b) On fixe $C_1, ..., C_K$ et on minimise par rapport à $\boldsymbol{\beta}$ l'équation

$$\min_{\boldsymbol{\beta}} \{||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda_1 ||\boldsymbol{\beta}||_1 + \frac{\lambda_2}{2} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{j,l \in C_k} ||\mathbf{x}_j \beta_j - \mathbf{x}_l \beta_l||^2\}. \quad (4.4)$$

La solution de l'équation (4.4) est donnée par la proposition suivante.

Proposition 4.3.1. *Soit $\boldsymbol{X}_{-j}$ la matrice $n \times (p-1)$ qui ne contient pas le vecteur $\boldsymbol{x}_j$ et $\boldsymbol{\beta}_{-j}$ le vecteur $\boldsymbol{\beta}$ sans l'entrée β_j. On pose*

$$\tilde{\boldsymbol{y}}_j = \boldsymbol{y} - \boldsymbol{X}_{-j}\beta_{-j}.$$

On suppose que $j \in C_k$. On pose $r_{jl} = \boldsymbol{x}_j^\top \boldsymbol{x}_l$, ainsi la solution de l'équation (4.4) est donnée par

$$\hat{\beta}_j = \frac{S(\tilde{\boldsymbol{y}}_j^\top \boldsymbol{x}_j + \frac{\lambda_2}{|C_k|}\sum_{l \in C_k, j \neq l} \beta_l r_{jl}, \lambda_1/2)}{r_{jj}(1 + \lambda_2 \frac{|C_k|-1}{|C_k|})},$$

avec S est la fonction définie dans l'équation (2.6).

Noter que la preuve de la proposition 4.3.1 est similaire à celle de la proposition 4.4.2 que nous allons présenter en détails pour notre approche. La preuve de la proposition 4.3.1 est laissée au lecteur.

Witten *et al.*(2014) ont prouvé, sur des données simulées et réelles, que la méthode CEN fournit de meilleurs résultats que les techniques existantes en présence d'une structure inconnue de groupes des prédicteurs.

Dans la section suivante nous allons présenter la méthode HHSVM-EN dans le cadre de la classification linéaire.

4.4 La méthode HHSVM avec la pénalité Élastique Net

Dans un problème de classification binaire standard, nous rappelons que nous avons n paires $\{(\mathbf{x}_{i.}, y_i), i = 1, 2, ..., n\}$ comme échantillon d'apprentissage avec $\mathbf{x}_{i.} \in \mathbb{R}^p$ vecteur d'observation sur les p prédicteurs et $y_i \in \{-1, 1\}$, les deux classes.

Nous rappelons que les données sont standardisées, c.-à-d.

$$\frac{1}{n}\sum_{1}^{n} x_{ij} = 0,$$

et

$$\frac{1}{n}\sum_{1}^{n} x_{ij}^2 = 1, \forall j = 1, ..., p.$$

Dans la section suivante, nous présentons la méthode *SVM* pénalisée avec la pénalité de type $\ell_1 + \ell_2$.

4.4.1 La méthode *SVM* vue comme méthode de régularisation avec une pénalité de type $\ell_1 + \ell_2$

Le problème d'optimisation est donné par

$$\min_{\beta,\beta_0} \frac{1}{n}\sum_{i=1}^{n}[1 - y_i(\beta_0 + \mathbf{x}_{i.}^\top \boldsymbol{\beta})]_+ + \boldsymbol{p}_{\lambda_1,\lambda_2}(\boldsymbol{\beta}), \tag{4.5}$$

où $\boldsymbol{p}_{\lambda_1,\lambda_2}(\boldsymbol{\beta})$ est la pénalité qui est définie par l'équation (2.5). La partie lasso dans la pénalité permet la sélection des prédicteurs et la partie Ridge tient compte de la corrélation entre les prédicteurs.

Cependant, la fonction de perte standard n'est pas dérivable partout, ainsi, Wang *et al.*(2008) ont remplacé la fonction de perte $[1 - y_i(\beta_0 + \mathbf{x}_{i.}^\top \boldsymbol{\beta})]_+$ dans le problème de l'équation (4.5) par une fonction d'Huber dénotée ϕ_c. Elle est lisse et dérivable partout. Le problème d'optimisation est défini par l'équation

$$\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n} \phi_c(y_i(\beta_0 + \mathbf{x}_{i.}^\top \boldsymbol{\beta})) + \boldsymbol{p}_{\lambda_1,\lambda_2}(\boldsymbol{\beta}), \tag{4.6}$$

où

ϕ_c est la fonction d'Huber (de l'anglais *Huberized hing loss*) qui est définie par

$$\phi_c(t) = \begin{cases} 0 & , t \geq 1 \\ \frac{(1-t)^2}{2\delta} & , 1 - \delta < t \leq 1 \\ 1 - t - \frac{\delta}{2} & , t \leq 1 - \delta. \end{cases} \tag{4.7}$$

Cette méthode est appelée l'approche HHSVM avec la pénalité EN que nous allons noter dans le reste du mémoire par HHSVM-EN.

4.4.2 Une généralisation de l'algorithme de la descente par coordonnée

Wang *et al.*(2008) ont proposé l'algorithme *LARS* (de *l'anglais least-angle regression*) pour calculer les chemins de solutions des coefficients par la méthode HHSVM-EN. Nous allons préciser que les quantités $\tilde{\boldsymbol{\beta}}$ et $\tilde{\beta}_0$ sont connues pour chaque étape de l'algorithme. On définit la quantité $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_{i\cdot}^{\mathsf{T}}\tilde{\boldsymbol{\beta}})$ et on pose

$$F(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n} \phi_c\{r_i + y_i x_{ij}(\beta_j - \tilde{\beta}_j)\} + \boldsymbol{p}_{\lambda_1,\lambda_2}(\beta_j). \tag{4.8}$$

Pour des valeurs de λ_1 et λ_2 fixées, l'algorithme de descente par coordonnée standard (Tseng 2001) dans le contexte de la méthode HHSVM-EN procède de la manière suivante :

1. initialiser $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$;
2. faire une descente par coordonnée pour $j = 0, 1, 2, ..., p$: mettre à jour de $\tilde{\beta}_j$, en minimisant la fonction objective

$$\tilde{\beta}_j = arg\min_{\beta_j} F(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}); \tag{4.9}$$

3. répéter l'étape 2 jusqu'à convergence.

Ce dernier algorithme pose un très grand défi, parce que le problème (4.9) n'admet pas une forme explicite pour les estimateurs $\hat{\beta}_j$, comme dans le cas de la méthode de la moindre carré pénalisée, pour laquelle β_j admet une solution explicite.

Ainsi, résoudre l'équation 4.9 demande un algorithme itératif, par exemple la méthode de Newton. Cependant, dans notre contexte, la fonction de perte ϕ_c n'admet pas une dérivée seconde, d'où la difficulté d'appliquer cette technique pour la méthode HHSVM-EN. Pour remédier à un tel problème, Yang et Zou (2013) ont recouru à une technique de maximisation-minimisation (MM) (voir Hunter et Lange, 2004). En vue de F donnée en (4.8). On majore la perte par une forme quadratique résultat en Q.

Proposition 4.4.1. *Une fonction Q qui majore la fonction F est définie par*

$$Q\{(\beta_j|\tilde{\beta}_0,\tilde{\boldsymbol{\beta}})\} = \frac{\sum_{i=1}^{n}\phi_c(r_i)}{n} + \frac{\sum_{i=1}^{n}\phi_c'(r_i)y_i x_{ij}(\beta_j-\tilde{\beta}_j)}{n} + \frac{(\beta_j-\tilde{\beta}_j)^2}{\delta} + \boldsymbol{p}_{\lambda_1,\lambda_2}(\beta_j).$$

Preuve. Puisque ϕ_c est continue et différentiable sur l'intervalle $[r_i, r_i+y_i x_{ij}(\beta_j-\tilde{\beta}_j)]$, alors, d'après le théorème de la valeur moyenne, il existe $d \in (0,1)$ tel que

$$\phi_c(r_i + y_i x_{ij}(\beta_j-\tilde{\beta}_j)) = \phi_c(r_i) + \phi_c'\,(r_i + d y_i x_{ij}(\beta_j-\tilde{\beta}_j))y_i x_{ij}(\beta_j-\tilde{\beta}_j).$$

Ainsi, nous pouvons écrire

$$\phi_c(r_i+y_i x_{ij}(\beta_j-\tilde{\beta}_j)) = \phi_c(r_i)+\phi_c'(r_i)y_i x_{ij}(\beta_j-\tilde{\beta}_j)+[\phi_c'(r_i+d\; y_i x_{ij}(\beta_j-\tilde{\beta}_j))-\phi_c'(r_i)]y_i x_{ij}(\beta_j-\tilde{\beta}_j).$$

Maintenant on cherche à majorer

$$[\phi_c'(r_i + d\; y_i x_{ij}(\beta_j-\tilde{\beta}_j)) - \phi_c'(r_i)].$$

On pose $h = d\; y_i x_{ij}(\beta_j-\tilde{\beta}_j)$. Puis, on calcule la dérivée première de ϕ_c :

$$\phi_c'(t) = \begin{cases} 0, & t \geq 1 \\ \frac{t-1}{\delta}, & 1-\delta < t \leq 1 \\ -1, & t \leq 1-\delta. \end{cases}$$

Soit $b > a$, d'après le graphique nous pouvons écrire

$$\frac{\phi_c'(b)-\phi_c'(a)}{b-a} \leq \frac{\phi_c'(b)-\phi_c'(a')}{b-a'} \leq \frac{\phi_c'(b')-\phi_c'(a')}{b'-a'} \leq \frac{1}{\delta}.$$

Nous remarquons par symétrie que pour tout a, b

$$|\phi_c'(a)-\phi_c'(b)| \leq \frac{|a-b|}{\delta},$$

ce qui donne

$$|\phi_c'(r_i + c\; y_i x_{ij}(\beta_j-\tilde{\beta}_j)) - \phi_c'(r_i)| \leq |\frac{1-r_i-h-(1-r_i)}{\delta}|.$$

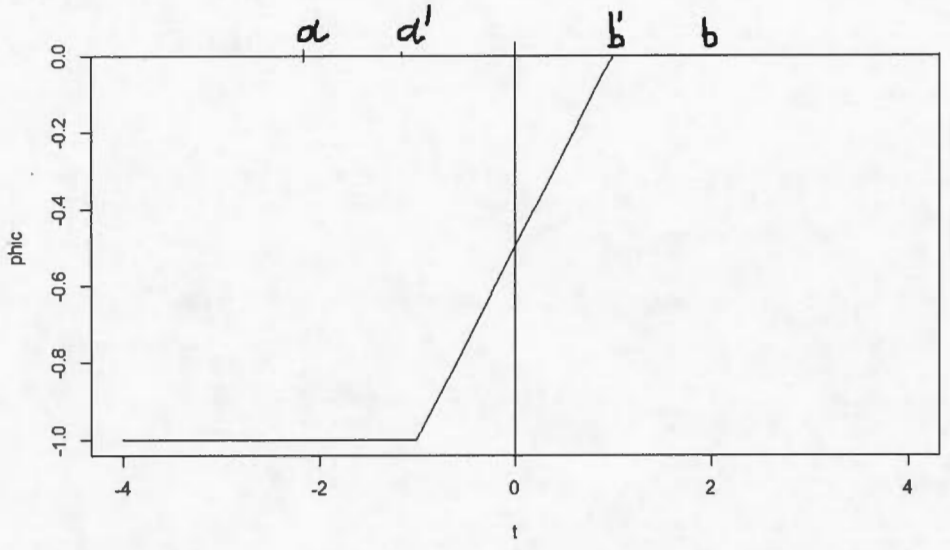

Figure 4.1: La dérivée première de ϕ_c.

Ainsi, nous avons

$$|\phi_c'(r_i + c\, y_i x_{ij}(\beta_j - \tilde{\beta}_j)) - \phi_c'(r_i)| \leq |\frac{h}{\delta}|,$$

d'où

$$\phi_c\{r_i + y_i x_{ij}(\beta_j - \tilde{\beta}_j)\} = \phi_c(r_i) + \phi_c'(r_i) y_i x_{ij}(\beta_j - \tilde{\beta}_j) + [\phi_c'\{r_i + c\, y_i x_{ij}(\beta_j - \tilde{\beta}_j)\} - \phi_c'(r_i)] y_i x_{ij}(\beta_j - \tilde{\beta}_j)$$

$$\leq \phi_c(r_i) + \phi_c'(r_i) y_i x_{ij}(\beta_j - \tilde{\beta}_j) + \frac{1}{\delta}(\beta_j - \tilde{\beta}_j)^2. \tag{4.10}$$

A l'aide des équations (4.8) et (4.10) nous constatons que F est majorée par une fonction quadratique Q telle que

$$Q(\beta_j | \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{\sum_{i=1}^n \phi_c(r_i)}{n} + \frac{\sum_{i=1}^n \phi_c'(r_i) y_i x_{ij}(\beta_j - \tilde{\beta}_j)}{n} + \frac{(\beta_j - \tilde{\beta}_j)^2}{\delta} + \boldsymbol{p}_{\lambda_1,\lambda_2}(\beta_j). \tag{4.11}$$

□

Ainsi, minimiser la fonction $Q(.)$ fait décroitre la fonction $F(.)$ à chaque itération. Une discussion détaillée de la propriété de descente de l'algorithme MM est donnée dans la section 4.8.

La solution de l'algorithme de descente par coordonnée appliqué à $Q(.)$ est donnée dans la proposition suivante.

Proposition 4.4.2. *La solution* $\hat{\beta^C}_j$ *qui minimise l'équation (4.11) est donnée par*

$$\hat{\beta^C}_j = \arg\min_{\beta_j} Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{S(\frac{2}{\delta}\tilde{\beta}_j - \frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}, \lambda_1)}{\frac{2}{\delta} + \lambda_2},$$

où la fonction S *est définie dans l'équation (2.6).*

La preuve de la proposition 4.4.2 est donnée dans l'appendice B. L'algorithme suivant permet de calculer les estimateurs des coefficients $\boldsymbol{\beta}$ pour la méthode HHSVM-EN.

Algorithme pour la méthode HHSVM-EN

- Initialiser $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$.
- Itérer les étapes (2a) et (2b) jusqu'à convergence :

  (2a) la descente par coordonnée pour $j = 1, ..., p$

  * calculer $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_{i.}^\top \tilde{\boldsymbol{\beta}})$;

  * calculer

$$\hat{\beta}_j^C = \frac{S(\frac{2}{\delta}\tilde{\beta}_j - \frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}, \lambda_1)}{\frac{2}{\delta} + \lambda_2};$$

  * poser $\tilde{\beta}_j = \hat{\beta}_j^C$;

  (2b) mettre à jour l'ordonnée à l'origine :

  * calculer $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_{i.}^\top \tilde{\boldsymbol{\beta}})$,

  * calculer

$$\hat{\beta}_0^C = \tilde{\beta}_0 - \frac{\delta}{2}\frac{\sum_{i=1}^n \phi_c'(r_i)y_i}{n};$$

  * poser $\tilde{\beta}_0 = \hat{\beta}_0^C$.

Dans la prochaine section, nous allons présenter notre nouvelle méthode qui adapte la pénalité CEN à la méthode HHSVM.

4.5 L'approche HHSVM avec la pénalité CEN dans le cas de grandes dimensions

Dans cette section, nous allons introduire une nouvelle méthode pour calculer les chemins de solutions des coefficients $\boldsymbol{\beta}$ pour la méthode HHSVM en utilisant la pénalité CEN vue dans la section 4.3. Nous allons noter notre méthode par HHSVM-CEN. Nous avons vu que la pénalité CEN tient compte de la structure de groupes des prédicteurs et leurs associations avec la variable réponse dans le cadre de la régression. Ainsi, nous allons suggérer un nouvel algorithme qui a pour but l'amélioration de l'algorithme proposé par Yang et Zou (2013), afin de tenir compte d'une telle structure de groupes des prédicteurs dans le cadre de la discrimination.

Le problème d'optimisation est défini par

$$\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n} \phi_c(y_i(\beta_0 + \mathbf{x}_i^{\top}\boldsymbol{\beta})) + \boldsymbol{p}_{CEN}(\boldsymbol{\beta}),$$

où $\boldsymbol{p}_{CEN}$ est la pénalité CEN définie dans l'équation (4.1). Ce problème d'optimisation n'est pas convexe. Alors, pour surmonter cette difficulté nous avons besoin de le séparer en deux parties, comme dans Witten *et al.*(2014). Ainsi, nous pouvons écrire l'algorithme comme suit :

1. On fixe les groupes $C_1, ..., C_K$ et on minimise la fonction objective $F(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ en faisant une descente par coordonnée pour chaque β_j, avec

$$F(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n} \phi_c\{r_i + y_i x_{ij}(\beta_j - \tilde{\beta}_j)\} + \boldsymbol{p}_{CEN}(\beta_j). \tag{4.12}$$

2. Après le calcul de la solution $\boldsymbol{\beta}$, on fait une *K-means* pour estimer les groupes $C_k, k = 1, ..., K$. K-means est une méthode de partitionnement de

données qui va nous servir dans cette étape à trouver la structure inconnue dans les prédicteurs, MacQueen, (1967).

Le problème d'optimisation pour la partie 1 est convexe en $\boldsymbol{\beta}$. Donc, nous allons faire une descente par coordonnée, (voir Friedman *et al.* 2010). La solution de ce problème est donnée par la proposition suivante.

Proposition 4.5.1. *Soit* $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_p)$, *une matrice* $n \times p$ *qui présente les prédicteurs et* $\boldsymbol{y} \in \mathbb{R}^n$ *est la variable réponse. La fonction objective est définie par*

$$Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{\sum_{i=1}^n \phi_c(r_i)}{n} + \frac{\sum_{i=1}^n \phi_c'(r_i) y_i x_{ij} (\beta_j - \tilde{\beta}_j)}{n} + \frac{(\beta_j - \tilde{\beta}_j)^2}{\delta} + \boldsymbol{p}_{CEN}(\beta_j),$$

avec $p_{CEN}(\boldsymbol{\beta})$ *est donnée dans l'équation (4.3). On suppose que* $j \in C_k$ *et que* $\rho_{jl} = \boldsymbol{x}_j^\top \boldsymbol{x}_l$. *Alors* β_j *qui minimise la fonction objective (4.12) est donnée par la forme explicite*

$$\hat{\beta}_j = \frac{S(-\frac{\sum_{i=1}^n \phi_c'(r_i) y_i x_{ij}}{n} + \frac{2\lambda_2}{|C_k|} \sum_{l \neq j} \rho_{jl} \tilde{\beta}_l + \frac{2}{\delta} \tilde{\beta}_j, \lambda_1)}{(\frac{2}{\delta} + 2\lambda_2 \frac{|C_k|-1}{|C_k|})}.$$

Preuve. Si nous gardons tous les autres variables fixes, la fonction $Q(.)$ est deux fois dérivable pour $\beta_j \neq 0$ et son sous-gradient est donné par

$$\frac{\partial Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})}{\partial \beta_j} = \frac{\sum_{i=1}^n \phi_c'(r_i) y_i x_{ij}}{n} + \frac{2}{\delta}(\beta_j - \tilde{\beta}_j) + \lambda_1 \text{signe}(\beta_j) + 2\lambda_2 \frac{|C_k| - 1}{|C_k|} \rho_{jj} \beta_j - \frac{\lambda_2}{|C_k|} \sum_{j \neq l} \rho_{jl} \tilde{\beta}_l.$$

Alors, nous avons

$$\frac{\partial Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})}{\partial \beta_j} = 0.$$

Ceci est équivaut à

$$\beta_j(\frac{2}{\delta} + 2\lambda_2 \rho_{jj} \frac{|C_k| - 1}{|C_k|}) = -\frac{\sum_{i=1}^n \phi_c'(r_i) y_i x_{ij}}{n} + \frac{\lambda_2}{|C_k|} \sum_{j \neq l} \rho_{jl} \tilde{\beta}_l + \frac{2}{\delta} \tilde{\beta}_j - \lambda_1 sign\{\beta_j\}.$$

Nous allons étudier les trois cas suivants :

Premier cas : si signe$(\beta_j) = 1$ alors $\beta_j > 0$ et nous avons

$$-\frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i x_{ij}}{n} + \frac{\lambda_2}{|C_k|} \sum_{j \neq l} \rho_{jl} \tilde{\beta}_l + \frac{2}{\delta} \tilde{\beta}_j > \lambda_1.$$

C'est-à-dire

$$\hat{\beta}_j = \frac{-\frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i x_{ij}}{n} + \frac{\lambda_2}{|C_k|} \sum_{j \neq l} \rho_{jl} \tilde{\beta}_l + \frac{2}{\delta} \tilde{\beta}_j - \frac{\lambda_1}{2}}{(\frac{2}{\delta} + 2\lambda_2 \rho_{jj} \frac{|C_k|-1}{|C_k|})}.$$

Deuxième cas : si signe$(\beta_j) = -1$ alors $\beta_j < 0$ et on peut écrire

$$-\frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i x_{ij}}{n} + \frac{\lambda_2}{|C_k|} \sum_{l \neq j} \rho_{jl} \tilde{\beta}_l + \frac{2}{\delta} \tilde{\beta}_j < \lambda_1,$$

qui donne

$$\hat{\beta}_j = \frac{-\frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i x_{ij}}{n} + \frac{\lambda_2}{|C_k|} \sum_{l \neq j} \rho_{jl} \tilde{\beta}_l + \frac{2}{\delta} \tilde{\beta}_j + \lambda_1}{(\frac{2}{\delta} + 2\lambda_2 \rho_{jj} \frac{|C_k|-1}{|C_k|})}.$$

Troisième cas : si $\beta_j = 0$ alors $\hat{\beta}_j = 0$. Ensuite, nous pouvons résumer le tout par

$$\beta_j^{\hat{CEN}} = \frac{S(-\frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i x_{ij}}{n} + \frac{\lambda_2}{|C_k|} \sum_{l \neq j} \rho_{jl} \tilde{\beta}_l + \frac{2}{\delta} \tilde{\beta}_j, \lambda_1)}{(\frac{2}{\delta} + 2\lambda_2 \frac{|C_k|-1}{|C_k|})},$$

avec S est la fonction définie dans l'équation (2.6). □

Nous allons chercher maintenant $\beta_0^{\hat{CEN}}$. On procéde de la même façon que pour $\beta_j^{\hat{CEN}}$. En effet, il faut minimiser la fonction quadratique $Q(\beta_0|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$, qui est définie comme

$$Q(\beta_0|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{\sum_{i=1}^{n} \phi_c(r_i)}{n} + \frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i (\beta_0 - \tilde{\beta}_0)}{n} + \frac{(\beta_0 - \tilde{\beta}_0)^2}{\delta}.$$

Ce qui est équivaut à minimiser

$$L(\beta_0) = \frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i}{n} \beta_0 + \frac{1}{\delta}(\beta_0^2 - 2\beta_0 \tilde{\beta}_0).$$

La dérivée de L par rapport à β_0 est donnée par

$$\frac{\partial L(\beta_0)}{\partial \beta_0} = \frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i}{n} + \frac{2}{\delta}(\beta_0 - \tilde{\beta}_0).$$

Donc

$$\frac{\partial L(\beta_0)}{\partial \beta_0} = 0 \Leftrightarrow \hat{\beta_0^{CEN}} = \tilde{\beta}_0 - \frac{\delta}{2}\frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i}{n}.$$

Les étapes de l'algorithme HHSVM-CEN sont résumées ci-dessous.

Algorithme 1 : La technique de la descente par cordonnée pour la méthode HHSVM-CEN

- Initialiser $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$.
- Itérer (2a) et (2b) jusqu'à convergence :
  - (2a) Faire une K-means pour $\mathbf{x}_1\tilde{\beta}_1, ..., \mathbf{x}_p\tilde{\beta}_p$ pour trouver les regroupements des prédicteurs en spécifiant un nombre K à l'avance.
  - (2b) Faire l'algorithme de la descente par coordonnée pour $j = 1, ..., p$:
    - (2b1) mettre à jour les β_j
      * calculer $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_{i.}^{\top}\tilde{\boldsymbol{\beta}})$,
      * calculer

$$\hat{\beta}_j^{CEN} = \frac{S(-\frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i x_{ij}}{n} + \frac{2\lambda_2}{|C_k|}\sum_{l \neq j} \rho_{jl}\tilde{\beta}_l + \frac{2}{\delta}\tilde{\beta}_j, \lambda_1)}{(\frac{2}{\delta} + 2\lambda_2\frac{|C_k|-1}{|C_k|})},$$

      * poser $\tilde{\beta}_j = \hat{\beta}_j^{CEN}$;

    . (2b2) mettre à jour l'ordonnée à l'origine :
      * calculer $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_{i.}^{\top}\tilde{\boldsymbol{\beta}})$;
      * calculer

$$\hat{\beta}_0^{CEN} = \tilde{\beta}_0 - \frac{\delta}{2}\frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i}{n};$$

      * poser $\tilde{\beta}_0 = \hat{\beta}_0^{C}$.

4.6 Relation avec la méthode HHSVM-EN

Dans le cas d'un seul groupe $K = 1$, notre méthode HHSVM-CEN est équivalente à la méthode HHSVM-EN. Ainsi, on peut montrer qu'elles sont mêmes égales si p est très grand.

En effet, si p est très grand et on a un seul groupe $K = 1$, alors pour des valeurs de λ_1 et λ_2 fixes, on a

$$\hat{\beta}_j^{CEN} = \frac{S(-\frac{\sum_{i=1}^n \phi_c'(r_i) y_i x_{ij}}{n} + \frac{2\lambda_2}{|C_k|}\sum_{l\neq j}\rho_{jl}\tilde{\beta}_l + \frac{2}{\delta}\tilde{\beta}_j, \lambda_1)}{(\frac{2}{\delta} + 2\lambda_2\rho_{jj}\frac{|C_k|-1}{|C_k|})}$$

puisque

$$\frac{2\lambda_2\sum_{l\neq j}\rho_{jl}\tilde{\beta}_l}{|C_k|} \simeq 0$$

et

$$\frac{|C_k|-1}{|C_k|} = 1 - \frac{1}{|C_k|} \simeq 1,$$

car $|C_k| = p$ est très grand.

Ainsi, on trouve presque la même forme de $\hat{\beta}_j^C$ que nous avons trouvé pour la méthode HHSVM-EN. En effet, rappelons que

$$\hat{\beta}_j^C = \frac{S(\frac{2}{\delta}\tilde{\beta}_j - \frac{\sum_{i=1}^n \phi_c'(r_i) y_i x_{ij}}{n}, \lambda_1)}{\frac{2}{\delta} + \lambda_2}.$$

Par conséquent, nous pouvons dire que la méthode HHSVM-CEN est une généralisation de la méthode HHSVM-EN lorsque il y a présence de groupes inconnus dans les prédicteurs.

4.7 Implémentation de l'algorithme de la méthode HHSVM-CEN

Notre algorithme contient deux parties qu'il faut itérer jusqu'à convergence. Une partie pour l'algorithme K-means (à préciser, c'est notre implémentation en Fortran) pour estimer les groupes à partir de $\mathbf{x}_1\beta_1, \mathbf{x}_2\beta_2, ..., \mathbf{x}_p\beta_p$. Pour l'autre partie,

nous gardons les groupes fixes, et nous faisons une descente par coordonnée pour calculer les chemins de solutions. Nous procédons comme dans Freidman *et al.* (2010). On fixe d'abord un λ_2, ensuite, on calcule les solutions pour une grille de valeurs de λ_1. Nous commençons d'abord par une valeur de λ_1 que l'on note λ_{max} et qui désigne la valeur de λ_1 pour laquelle tous les β_j sont nuls. Pour calculer λ_{max}, nous avons besoin de calculer $\hat{y}$, pour le modèle sans prédicteurs, en effet,

$$\hat{y} = arg\min_{y} \frac{1}{n}\sum_{i=1}^{n} \phi_c(y_i y),$$

avec les conditions de Karush-Kuhn-Tucker(KKT), on a

$$\lambda_{max} = \frac{1}{n}\max_{j} |\sum_{i=1}^{n} \phi_c^{'}(\hat{y} y_i x_{ij})|.$$

Comme dans Yang et Zou (2013), on pose $\lambda_{min} = \tau\lambda_{max}$, avec $\tau = 10^{-2}$ comme valeur par défaut de τ si $n < p$ et $\tau = 10^{-4}$ pour $n \geq p$. Entre λ_{max} et λ_{min}, on place 98 valeurs en échelle algorithmique pour obtenir ainsi 100 valeurs de λ_1. Ainsi, nous avons

$$\lambda_1[1] = \lambda_{max}, \quad \lambda_1[100] = \lambda_{min}.$$

On a $\hat{\boldsymbol{\beta}} = 0_p$, pour la première valeur de λ_1 qui est $\lambda_1[1]$. Pour calculer la solution pour $\lambda_1[k+1]$, on initialise l'algorithme par la solution obtenue pour $\lambda_1[k]$. Ainsi, pour calculer la solution pour chaque λ_1, on obtient d'abord une solution qui contient des valeurs nulles et des valeurs non nulles des β_j. Nous prenons ainsi seulement les valeurs non nulles et on répète le processus jusqu'à convergence. On itère les deux parties (2a) et (2b) de notre algorithme jusqu'à convergence. Le critère de convergence utilisé dans l'algorithme est

$$||\hat{\boldsymbol{\beta}}^k - \hat{\boldsymbol{\beta}}^{k-1}||/||\hat{\boldsymbol{\beta}}^k||^2 < 10^{-5}.$$

Remarque: Approximation de la méthode *SVM*

L'algorithme 1 est valable pour n'importe quelle valeur de δ. On remarque que pour $\delta \simeq 0$ la méthode HHSVM-CEN est équivalente à la méthode SVM-CEN,

nous allons utiliser cet avantage pour obtenir une approximation pour la solution de la méthode *SVM* avec la pénalité CEN. Cette intuition vient du fait que $\lim_{\delta \to 0} \phi_c(t) = [1-t]_+$. Une illustration de la comparaison entre ces deux fonctions de perte est donnée dans la figure 4.1.

4.8 La descente par coordonnée et le principe MM

Dans la section 3, nous avons parlé de la descente par coordonnée pour faire une mise à jour pour chaque coordonnée en minimisant la fonction $F(.)$, et on a souligné que cette minimisation n'admet pas une forme explicite. Ainsi, pour remédier à un tel problème, nous avions majoré $F(.)$ par une fonction quadratique $Q(.)$, et à la place de minimiser $F(.)$, on minimisait $Q(.)$. Ceci est connu sous le nom *minimisation-maximisation* (MM) (voir Hunter et Lange, 2004). Pour assurer la convergence de $F(.)$ à un point stationnaire. La propriété de la descente de la fonction $F(.)$ à chaque itération est garantie par l'algorithme MM. Elle est montrée dans la proposition suivante.

Proposition 4.8.1. *Soit $F(.)$ la fonction objective à minimiser pour chaque β_j. $Q(.)$ la fonction quadratique qui majore $F(.)$. Alors*

$$F(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) \leq Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}), \tag{4.13}$$

$$F(\tilde{\beta}_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = Q(\tilde{\beta}_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}). \tag{4.14}$$

Preuve. Nous avons déjà montré l'équation (4.13) dans la proposition 4.4.1. L'équation (4.14) est triviale, en effet, il suffit de remplacer dans les deux fonctions β_j par $\tilde{\beta}_j$, pour obtenir l'égalité. □

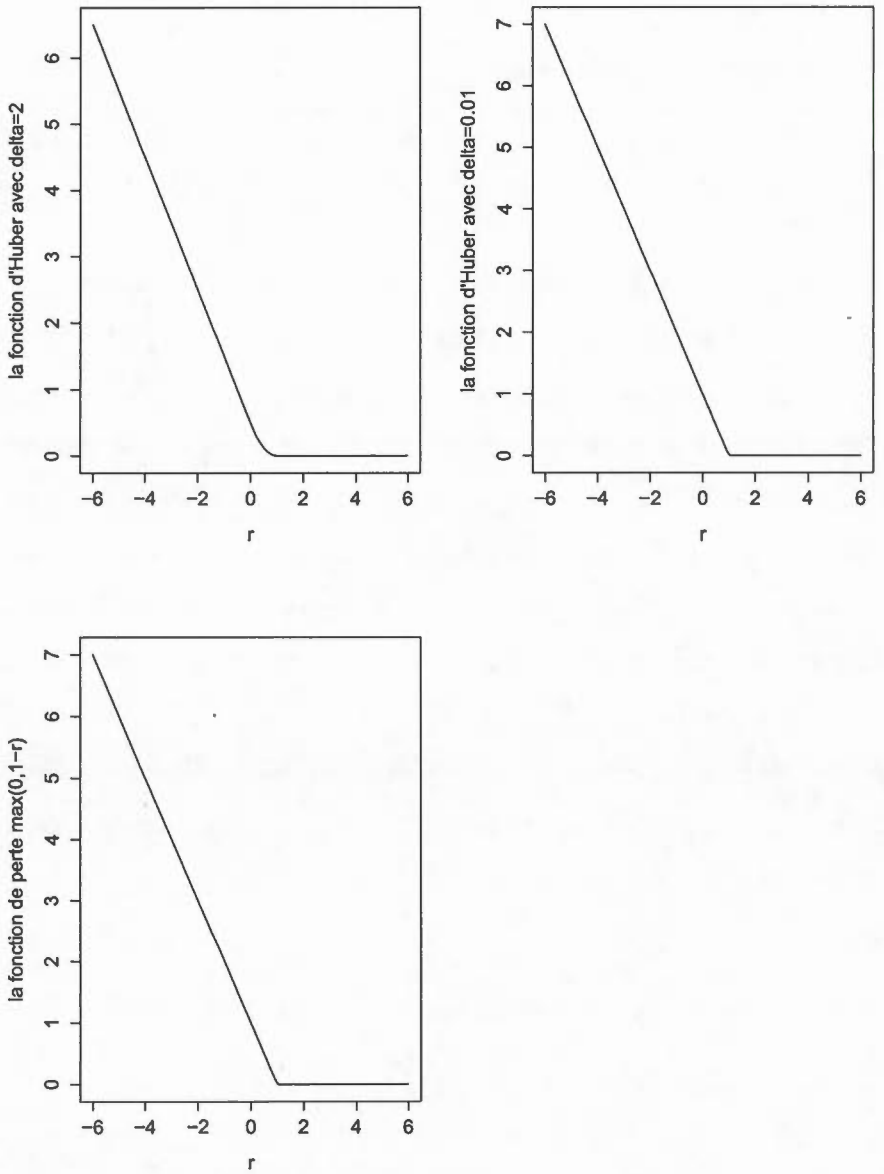

Figure 4.2: (a) La fonction d'Huber ϕ_c avec $\delta = 2$, (b) La fonction d'Huber ϕ_c avec $\delta = 0.01$, et (c) la fonction de perte de la méthode SVM standard, $[1 - r]_+$.

4.9 Une extension de l'algorithme HHSVM-CEN pour d'autres fonctions de perte

Dans cette section, comme dans Yang et Zou (2013), nous allons généraliser l'algorithme 1 pour des fonctions de perte autres que la fonction de perte définie dans l'équation (4.12). Autrement dit, nous allons montrer que la pénalité CEN peut s'adapter à autres fonctions de perte que la fonction de perte *SVM*, entre autres, la fonction logistique.

Le problème d'optimisation que nous avons proposé peut s'écrire sous une forme générale comme

$$\min_{(\beta_0,\beta_j)} F\{(\beta_j|\tilde{\beta}_0,\tilde{\boldsymbol{\beta}})\} = \min_{(\beta_0,\beta_j)} \frac{1}{n}\sum_{i=1}^{n} L\{r_i + y_i x_{ij}(\beta_j - \tilde{\beta}_j\} + \boldsymbol{p_{CEN}}(\beta_j), \qquad (4.15)$$

tel que $L(.)$ est une fonction de perte convexe et $\boldsymbol{p_{CEN}}$ est la pénalité CEN. Dans la section 4.5, nous avons trouvé que la fonction quadratique Q qui majore la fonction F est donnée par

$$Q(\beta_j|\tilde{\beta}_0,\tilde{\boldsymbol{\beta}}) = \frac{\sum_{i=1}^{n}\phi_c(r_i)}{n} + \frac{\sum_{i=1}^{n}\phi_c'(r_i)y_i x_{ij}(\beta_j - \tilde{\beta}_j)}{n} + \frac{(\beta_j - \tilde{\beta}_j)^2}{\delta} + \boldsymbol{p_{CEN}}(\beta_j). \qquad (4.16)$$

Si nous posons $z_{ij} = y_i x_{ij}(\beta_j - \tilde{\beta}_j)$ alors $\frac{1}{n}\sum_{i=1}^{n} z_{ij}^2 = \frac{1}{n}\sum_{i=1}^{n} x_{ij}^2(\beta_j - \tilde{\beta}_j)^2 = (\beta_j - \tilde{\beta}_j)^2$ parce que les données sont centrées et réduites. L'équation (4.16) devient

$$Q(\beta_j|\tilde{\beta}_0,\tilde{\boldsymbol{\beta}} = \frac{1}{n}\sum_{i=1}^{n}[\phi_c(r_i) + \phi_c'(r_i)z_{ij} + \frac{z_{ij}^2}{\delta}] + \boldsymbol{p_{CEN}}(\beta_j). \qquad (4.17)$$

Nous avons montré que la fonction de perte ϕ_c vérifie la condition de majoration définie dans l'équation (4.10). L'équation (4.10) peut s'écrire comme

$$\phi_c(r_i + z_{ij}) \leq \phi_c(r_i) + \phi_c'(r_i)z_{ij} + \frac{M}{2}z_{ij}^2, \qquad (4.18)$$

tel que $M = \frac{2}{\delta}$. Ainsi, nous pouvons conclure que c'est l'équation (4.18) qui a permis de majorer la fonction $F(.)$ par la forme quadratique définie dans l'équation

(4.16). Le lemme suivant nous montre que toute fonction de perte dérivable et que sa dérivée première est M_1-lipschitzienne ou que sa dérivée seconde est bornée, alors elle vérifie la condition de majoration de l'équation (4.18). Par conséquence, elle admet une fonction quadratique $Q(.)$ qui majore sa fonction $F(.)$ donnée dans 4.15.

Lemme 4.9.1. *{ Yang et Zou, 2013)}*

a. *Toute fonction L dérivable pour laquelle la dérivée première est M_1-lipschitzienne c.-à-d.*

$$|L^{'}(a) - L^{'}(b)| \leq k|a - b| \quad \forall a, b, \tag{4.19}$$

alors L satisfait à la condition de majoration avec $M = 2M_1$.

b. *Pour toute fonction de perte L pour laquelle sa dérivée seconde est bornée c.-à-d. : il existe M_2 tel que pour tout $t \in \mathbb{R}$ on a*

$$L^{''}(t) \leq M_2,$$

alors L vérifie la condition de majoration avec $M = M_2$.

La démonstration du lemme 4.9.1 est triviale et elle est donc laissée au lecteur.

Exemple

- La fonction de perte logistique :

  La fonction de perte logistique vue dans le chapitre 3 vérifie la condition de majoration (4.18). Pour montrer cela, nous allons montrer que la fonction de perte logistique, L, vérifie la condition (b) du lemme 4.9.1. En effet, $L(t) = \log\{1 + \exp(-t)\}$ et sa dérivée seconde est donnée par $L^{''}(t) = \frac{e^{-t}}{(e^{-t}+1)^2}$. Nous savons que pour tout $a, b \in \mathbb{R}$, on a

$$(a + b)^2 \geq 2ab,$$

  alors

$$L^{''}(t) \leq \frac{1}{2}.$$

Donc, L vérifie le lemme 4.9.1 avec $M = M_2 = \frac{1}{2}$.

L'algorithme qui permet de résoudre le problème d'optimisation (4.15) pour une fonction de perte quelconque qui satisfait a) ou b) du lemme est donné ci-dessous.

Algorithme 2 : Une descente par cordonnée pour le problème d'optimisation (4.13) avec la pénalité CEN.

- Initialiser $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$.
- Itérer de (2a) et (2b) jusqu'à convergence :
  - (2a) Faire une K-means pour $\mathbf{x}_1\tilde{\beta}_1, ..., \mathbf{x}_p\tilde{\beta}_p$;
  - (2b) une descente par coordonnée pour $j = 1, ..., p$;
    - (2b1) mettre à jour les $\hat{\beta}_j$,
      * calculer $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_{i.}^\top \tilde{\boldsymbol{\beta}})$,
      * calculer

$$\hat{\beta}_j^{CEN} = \frac{S(-\frac{\sum_{i=1}^n L_c'(r_i)y_i x_{ij}}{n} + \frac{2\lambda_2}{|C_k|}\sum_{l\neq j}\rho_{jl}\tilde{\beta}_l + M\tilde{\beta}_j, \lambda_1)}{(M + 2\lambda_2\frac{|C_k|-1}{|C_k|})},$$

      * poser $\tilde{\beta}_j = \hat{\beta}_j^{CEN}$.

    . (2b2) mettre à jour l'ordonnée à l'origine :
      * calculer $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_{i.}^\top \tilde{\boldsymbol{\beta}})$,
      * calculer

$$\hat{\beta}_0^C = \tilde{\beta}_0 - \frac{1}{M}\frac{\sum_{i=1}^n L_c'(r_i)y_i}{n};$$

      * poser $\tilde{\beta}_0 = \hat{\beta}_0^{CEN}$.

CHAPITRE V

SIMULATION DES DONNÉES ET ANALYSE DE DONNÉES RÉELLES

Afin d'illustrer la méthodologie développée dans le chapitre 4, nous allons présenter, dans ce chapitre, une étude de simulation ainsi qu'une analyse sur données réelles.

5.1 Étude de simulation

Dans cette étude de simulation, nous allons générer les données selon deux scénarios, un scénario avec $n > p$ et un deuxième scénario avec $p >> n$. Dans les deux scénarios nous allons générer trois groupes de prédicteurs C_1, C_2 et C_3 avec $K = 3$. Notons que C_1 et C_2 sont associés à la variable réponse.

5.1.1 Scénario 1

On génère une matrice $\boldsymbol{X}$, $n \times p$ avec $n = 50$ et $p = 30$. Les lignes de la matrice $\boldsymbol{X}$ sont indépendantes et identiquement distribuées selon la loi normale multidimensionnelle $N_p(\mathbf{0}_p, \Sigma)$ avec Σ est une matrice $p \times p$ diagonale par blocs définie

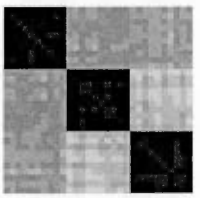

Figure 5.1: La carte de chaleur de la matrice Σ.

comme suit

$$\Sigma_{ij} = \begin{cases} 1, & \text{si} \quad i = j \\ 0.8, & \text{si} \quad i \leq 10, j \leq 10 \text{ et } i \neq j \\ 0.8, & \text{si} \quad 11 \leq i \leq 20, 11 \leq j \leq 20 \text{ et } i \neq j \\ 0, & \text{sinon.} \end{cases}$$

La figure (5.1) illustre la carte de chaleur (de l'anglais *heatmap*) de la matrice Σ. Le vecteur β des coefficients est donné par

$$\beta = (\underbrace{1, ..., 1}_{10}, \underbrace{-1, ..., -1}_{10}, \underbrace{0..., 0..., 0}_{10})^{\top}.$$

Cela veut dire que les 10 premiers prédicteurs $\mathbf{x}_1, ..., \mathbf{x}_{10}$ sont corrélés entre eux et ils sont associés positivement à la variable réponse et les prédicteurs $\mathbf{x}_{11}, .., \mathbf{x}_{20}$ sont corrélés entre eux et ils sont associés négativement à la variable réponse. Les 10 derniers prédicteurs sont corrélés entre eux, mais ils ne sont pas liés à la variable réponse. Ainsi, on génère une variable réponse z selon le modèle de l'équation (2.1) et on convertit z à une variable y selon $P(y = -1) = \frac{1}{1+\exp(-z)}$ et $P(y = 1) = \frac{1}{1+\exp(z)}$. La figure 5.2 montre les chemins de solutions des coefficients des six techniques de discrimination : HHSVM-CEN avec $\delta = 2$, HHSVM-CEN

avec $\delta = 0.01$, HHSVM-EN avec $\delta = 2$, HHSVM-EN avec $\delta = 0.01$, la logistique-EN et HHSVM-lasso.

Nous pouvons constater que notre nouvelle méthode sépare d'avantage les groupes de prédicteurs. Ainsi, elle fournit de meilleurs résultats que les quatre autres méthodes surtout pour $\delta = 2$. Pour $||\beta||_1$ grand, la méthode HHSVM-EN avec $\delta = 0.01$ fournit des solutions non nulles pour le groupe bleu. Ceci peut s'expliquer par la nouvelle fonction de pénalité CEN. En effet, contrairement aux autres, dans notre problème d'optimisation nous cherchons à la fois le meilleur regroupement des prédicteurs ainsi que les groupes qui sont fortement associés à la variable réponse. Notons que pour que notre méthode fonctionne, en ignorant la structure des groupes, il suffit de rentrer le nombre de groupes à l'avance. Pour le scénario 1 nous avons fixé $K = 3$. Nous allons signalé que le temps de convergence de notre algorithme lorsque $\delta = 0.01$ est plus grand que le temps de convergence lorsque $\delta = 2$. A priori un choix adapté pour δ est 2 comme dans le cas de la méthode HHSVM-EN.

5.1.2 Scénario 2

Dans cette section, nous allons étudier un deuxième scénario afin d'illustrer le cas de données de grandes dimensions. Ainsi, nous allons simuler les données comme dans le scénario 1. En effet, les $\epsilon_i, i = 1, ..., n$ sont indépendantes et de même $N(0, 2.5^2)$. Les observations (c.-à-d. les lignes de $\boldsymbol{X}$) sont indépendantes et de même $N_p(\mathbf{0}_p, \Sigma)$, avec $n = 50$. La matrice Σ est $p \times p$ diagonale par blocs avec

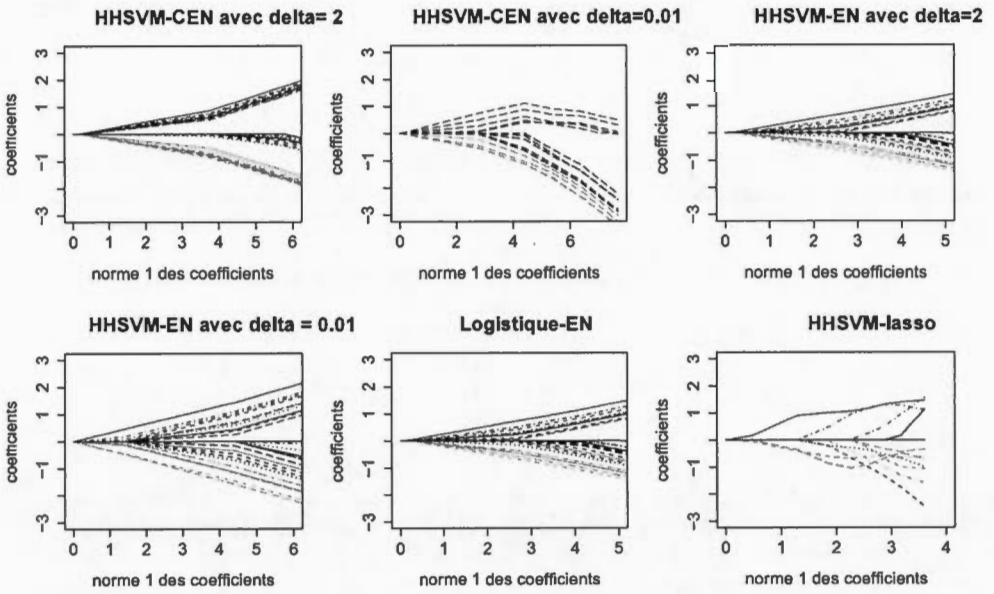

Figure 5.2: Les chemins de solutions des coefficients pour les six techniques. L'axe des abscisses désigne la norme ℓ_1 des coefficients et l'axe des ordonnées désigne les coefficients $\hat{\beta}$.

$p = 1000$, avec la forme suivante

$$\Sigma_{ij} = \begin{cases} 1 & \text{si} \quad i = j \\ \rho & \text{si} \quad i \leq 50, j \leq 50 \text{ et } i \neq j \\ \rho & \text{si} \quad 51 \leq i \leq 100, 51 \leq j \leq 100 \text{ et } i \neq j \\ 0 & \text{sinon.} \end{cases}$$

Nous simulons également le vecteur $\boldsymbol{\beta}$ de la façon suivante

$$\begin{cases} \beta_j \sim Unif[0.9, 1.1], & \text{si} \quad 1 \leq j \leq 25 \\ \beta_j \sim Unif[-1.1, -0.9], & \text{si} \quad 51 \leq j \leq 75 \\ \beta_j = 0 & \text{sinon.} \end{cases}$$

Ainsi, nous avons choisi trois groupes C_1, C_2 et C_3, avec C_1 et C_2 contenant 50 prédicteurs chacun. Les prédicteurs dans chaquun des deux groupes C_1 et C_2 sont corrélés entre eux mais 25 prédicteurs dans chaque groupe ne sont pas associés à la variable réponse. Les prédicteurs du groupe C_3 ne sont pas corrélés entre eux également et ils ne sont pas liés à la variable réponse. La figure 5.3 montre les chemins de solutions de $\hat{\boldsymbol{\beta}}$ pour ce scénario. Nous constatons que notre nouvelle méthode regroupe mieux les groupes qui sont associés à la variable réponse, par contre la méthode HHSVM-EN ne distingue pas entre les groupes qui sont associés avec la variable réponse et qui ne les sont pas. Nous allons maintenant comparer la performance des approches suivantes :

[1] HHSVM-EN avec $\delta = 2$;

[2] HHSVM-EN avec $\delta = 0.01$;

[3] HHSVM-CEN avec $\delta = 2$;

[4] HHSVM-CEN avec $\delta = 0.01$;

[5] HHSVM-Lasso ;

[6] Logistique-EN.

Afin de faire une comparaison rigoureuse entre notre méthode HHSVM-CEN et les autres méthodes, nous allons calculer les statistiques suivantes pour chaque méthode :

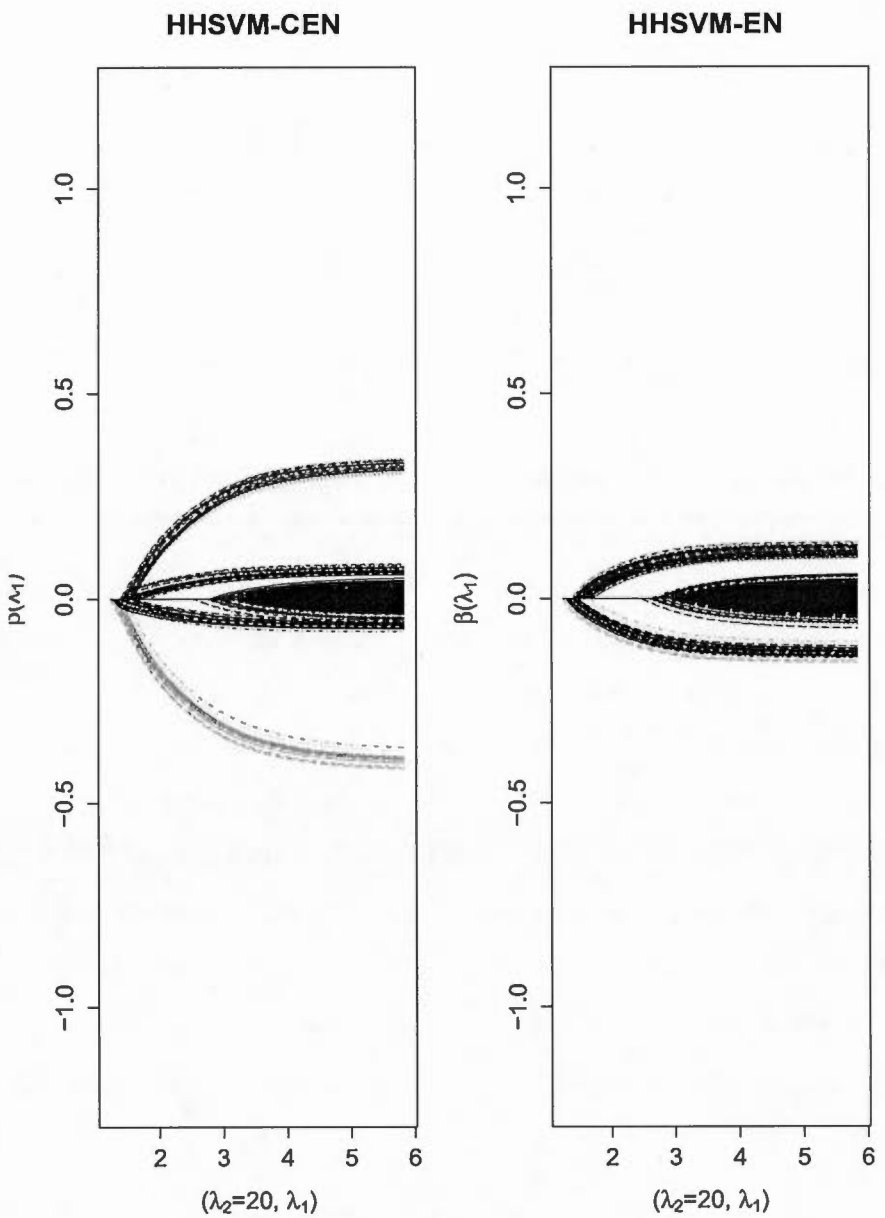

Figure 5.3: Les chemins de solutions de $\hat{\boldsymbol{\beta}}$ pour notre nouvelle méthode HHSVM-CEN et la méthode HHSVM-EN. L'axe des abscisses désigne la grille de valeurs pour le paramètre de régularisation λ_1 et l'axe des ordonnées désigne les coefficients $\hat{\boldsymbol{\beta}}$.

[1] Erreur de classement : elle désigne le nombre d'individus mal classés ;

[2] Nom.non-zero désigne le nombre d'éléments non nuls dans $\hat{\boldsymbol{\beta}}$ (c.-à-d. $\hat{\beta}_j \neq 0$) ;

[3] RIs (de l'anglais *Rand Index*), qui mesure la concordance entre le groupe estimé et le vrai groupe, (voir Rand, 1971) ;

[4] Auc (de l'anglais *Area under the curve*), Fawcett (2006).

Pour calculer ces statistiques nous avons besoin de produire des groupes de prédicteurs pour chaque méthode. Ainsi, pour les méthodes HHSVM-EN, HHSVM-Lasso et Logistique-EN, nous avons déterminé les groupes de prédicteurs en procédant en deux étapes :

1) nous calculons $\hat{\boldsymbol{\beta}}$ en utilisant l'algorithme discuté dans la section 3 ;
2) nous faisons une K-means basée sur $[\mathbf{x}_1\hat{\beta}_1, ..., \mathbf{x}_p\hat{\beta}_p]$ pour estimer les groupes de prédicteurs.

Notre méthode estime les $\hat{\boldsymbol{\beta}}$ et regroupe les prédicteurs en même temps. Ainsi, les groupes de prédicteurs par cette dernière sont donnés directement par l'algorithme 1 après convergence. Notons que, pour calculer ces statistiques, nous avons pris une grille de valeurs pour λ_1 et une autre pour λ_2. Ainsi, nous avons fait une validation croisée par paquet de 5 pour choisir les valeurs λ_{1min} et λ_{2min} pour lesquelles l'erreur de classement est minimale.

Le tableau 5.1 présente les résultats de cette étude. Les résultats sont obtenus par rééchantillonnage avec 10 réplications et nous avons pris la moyenne pour chaque statistique.

Comme nous avons souligné dans le chapitre 4, notre méthode tient compte de la structure des groupes de prédicteurs et l'association de ces groupes avec la variable réponse. Le tableau 5.1 montre que notre approche est meilleure en termes de regroupement des prédicteurs associés à la variable réponse que les autres méthodes.

Tableau 5.1: Résultats de simulation pour le scénario 2.

Méthode	Erreur de classement	Nom.Non-zéros	RIs	Auc
HHSVM-CEN avec $\delta = 2$	2.721	87.6	0.912	0.98
HHSVM-CEN avec $\delta = 0.01$	3.074	190.8	0.91	0.99
HHSVM-EN avec $\delta = 2$	2.803	109.7	0.909	0.978
HHSVM-EN avec $\delta = 0.01$	2.816	98.4	0.912	0.971
Logistique-EN	2.793	109.5	0.912	0.988
HHSVM-Lasso	3.571	27.8	0.912	0.987

5.1.3 Comment choisir le nombre de groupes K

Dans le cas de données simulées, on connaît le nombre de groupes $K = 3$. Cependant, dans le cas de données réelles, une telle information n'est pas disponible. Notre algorithme a besoin d'une spécification du nombre de classes de prédicteurs au départ. Afin de voir l'impact du nombre de groupes sur l'efficacité de classement des prédicteurs par notre méthode, nous avons mené une étude de simulation avec les paramètres du scénario 2, mais à chaque fois nous changeons le nombre $K = \{2, 3, 5, 7\}$ qu'il faut spécifier pour notre méthode.

Nous avons évalué la performance de la méthode avec différentes valeurs de K en utilisant les mêmes statistiques du tableau 5.1, avec une validation croisée par paquet de 5 pour le choix des valeurs optimales de λ_1 et λ_2.

Les résultats sont illustrés dans le tableau 5.2. Nous pouvons constater que les meilleures résultats sont donnés pour $K = 3$. Les résultats obtenus pour les autres valeurs de K sont bons également. Le tableau 5.2 nous montre également que la mauvaise spécification du nombre de groupes à priori pour notre méthode a très peu d'impact sur la performance de notre méthode pour ce scénario.

Tableau 5.2: Les résultats de simulation pour le scénario 2 pour différentes valeurs de K en utilisant une validation croisée avec cinq groupes.

Nombre de groupes	Erreur de classement	Nombre de $\hat{\beta}$ non zero	RIs	auc
K = 2	2.88	114.5	0.908	0.97
K = 3	2.721	87.6	0.912	0.98
K = 5	2.85	124	0.910	0.97
K = 7	2.88	119.8	0.908	0.97

5.2 Application à des données réelles : Le cancer de la prostate

Le cancer de la prostate est une tumeur qui se développe dans les cellules de la prostate, ce petit organe de l'appareil reproductif chez les hommes. Afin d'illustrer notre nouvelle méthodologie sur des données réelles, nous allons étudier le jeu de données *Prostate Cancer*, de Efron (2010). C'est une base de données qui contient des mesures sur 102 patients, dont 52 sont atteint du cancer et 50 sont des contrôles. Ces données sont recueillies par un groupe de chercheurs des États-Unis, qui ont pris également les mesures de 6033 expressions de gènes pour les 102 patients. Notre objectif est de voir s'il y a une structure de groupe dans les 6033 expressions de gènes, qui peut aider à expliquer l'architecture biologique de cette maladie. En effet, une telle structure peut aider à comprendre comment les gènes interagissent pour causer la maladie. Ainsi, le regroupement des gènes peut aider à comprendre la structure biologique sous-jacente du cancer de la prostate. On considère y la variable réponse (individu malade, non malade). Nous avons ainsi 102 individus avec $y = 1$ individu normal et $y = -1$ pour un individu atteint du cancer. Les prédicteurs sont les expressions de gènes avec $p = 6033$.

Notons $\boldsymbol{X}$ la matrice des données, elle est $n \times p$ avec $n = 102$ et $p = 6033$. On

Tableau 5.3: Résultats pour les données sur le cancer de la prostate.

Méthode	Erreur de classement	Nom.Non-zéros	Auc
HHSVM-CEN($\delta = 2$)	6.80	54.975	0.622
HHSVM-CEN($\delta = 0.01$)	6.51	55	0.66
HHSVM-EN($\delta = 2$)	6.90	49.71	0.622
HHSVM-EN($\delta = 0.01$)	6.50	76.39	0.620
Logistique-EN	6.87	43.69	0.622
HHSVM-lasso	7	10.55	0.62

prend une grille de valeurs pour λ_1 et une autre pour λ_2. On fait une validation croisée par paquet de 5 pour choisir λ_{1min} et λ_{2min}, qui conviennent à la valeur minimale de l'erreur de classement. On prend ces valeurs pour calculer les quatre statistiques comme dans le cas du scénario 2 de la section précédente. Notons que, pour les données réelles, nous ne connaissons pas les vrais groupes, donc nous ne pouvons pas calculer la statistique RIs.

Nous avons souligné que notre méthode a besoin de spécifier le nombre de groupes au départ de notre algorithme. Ainsi, nous allons mené une étude de classification hiérarchique en utilisant la méthode de Ward, afin d'obtenir un nombre de groupes approprié. Cette méthode nous a donnés que $K = 3$ est un bon choix. Nous avonss répété le processus 10 fois et faire une moyenne pour les trois statistiques : Erreur de classification, Nom.non zéros et Auc. Les résultats sont présentés dans le tableau 5.4.

Nous constatons que notre méthode HHSVM-CEN avec $\delta = 2$ est meilleure que les autres techniques, car elle donne la valeur minimale pour l'erreur de classification. Pour $\delta = 0.01$, les résultats obtenus pour l'erreur de classification pour les deux

techniques HHSVM-CEN et HHSVM-EN sont presque égaux. Pour la prédiction, notre méthode HHSVM-CEN avec $\delta = 0.01$ donne les meilleurs prédictions avec une moyenne de l'aire en dessous de la courbe du classificateur égale à 0.66.

La méthode HHSVM-Lasso comme d'habitude nous ne rentre pas beaucoup de $\hat{\beta}_j \neq 0$, mais demeure une bonne technique de classement.

CONCLUSION

Après le grand succès de l'approche de Yang et Zou (2013) pour le calcul des solutions des coefficients de la méthode HHSVM-EN, nous avons présenté la méthode HHSVM-CEN qui s'inscrit dans le cadre de l'apprentissage supervisé. Plus précisément, nous avons présenté une approche de classification pour des données de grandes dimensions. L'objectif principal de ce mémoire était de proposer une nouvelle technique de l'apprentissage statistique, afin d'améliorer la méthode HHSVM-EN. En effet, notre méthode tient compte de la structure des groupes de prédicteurs et de leur associations avec la variable réponse. Ainsi, notre algorithme est composé de deux parties : une partie où l'algorithme K-means est utilisé pour estimer les groupes de prédicteurs et une seconde partie où l'on fait appel à l'algorithme de la descente par coordonnée pour estimer les coefficients du modèle. Nos résultats sur les données simulées illustrent le gain de notre méthode comparée à celle de Yang et Zou (2013). Nos résultats sur les données réelles montrent également une bonne performance de notre méthode.

Notre problème d'optimisation n'étant pas convexe, nous l'avons séparer en deux parties pour faciliter la convergence de notre algorithme. Nous avons montré également que la fonction de perte dans le problème d'optimisation de la première partie de notre algorithme n'admet pas une dérivée seconde. Ainsi, comme dans le cas de HHSVM-EN, nous avons proposé une technique de majoration-minimisation de la fonction objective par une autre fonction objective deux fois dérivable.

Bien que notre méthode fournit de bons résultats au niveau de la prédiction comparée à HHSVM-EN, elle est coûteuse en terme de temps de calcul en la comparant

à la méthode HHSVM-EN. Cette dernière compte une seule partie pour la descente par coordonnée. Notre méthode par contre est composée de deux parties : une partie pour l'algorithme K-means afin d'estimer les groupes des prédicteurs et l'autre partie pour la descente par coordonnée pour estimer les coefficients de la méthode.

Nous avons présenté dans la section 4.9 un deuxième algorithme qui permet de résoudre les problèmes d'optimisation pour d'autres fonctions de perte qui vérifient la condition de majoration. Nous avons prouvé que la fonction logistique est une fonction de perte pour laquelle nous pouvons appliquer cet algorithme. Notre prochain objectif pour des travaux futurs serait d'implémenter notre algorithme pour résoudre le modèle logistique avec la pénalité CEN.

Une autre extension de nos travaux sera l'adaptation de l'algorithme de Tseng (2009) à notre approche. En effet, Tseng (2009) a présenté une descente par gradient pour résoudre des problèmes similaires au problème d'optimisation présenté dans ce mémoire. Ainsi, appliquer une telle technique pourra aboutir à de meilleures résultats, et pourra réduire le temps de calcul de notre programme.

APPENDICE A

EXEMPLE D'APPLICATION POUR DES MÉTHODES DE DISCRIMINATION AVEC LE LOGICIEL R

La base de données (*Smarket*) de James et al. (2013) est un ensemble de données qui contient les taux de rendements en pourcentage de l'indice boursier *S&P* 500 pour 1250 jours, du début de l'année 2001 à la fin de l'année 2005. Pour chaque jour, on enregistre les taux de rendement pour les 5 jours précédents, ces taux sont *lag1*,..., *lag5*. On enregistre également le *Volume* (le nombre de titres échangés le jour qui précède en milliards), *Today* (le rendement en ce jour-là). La variable *Direction* est une variable qui montre si le marché est à la hausse ou à la baisse ce jour-là.

La matrice de corrélation des variables est donnée comme suit

```
> cor(Smarket [,-9])
             Year         Lag1         Lag2         Lag3         Lag4      Lag5
Year    1.00000000  0.029699649  0.030596422  0.033194581  0.035688718  0.02978
Lag1    0.02969965  1.000000000 -0.026294328 -0.010803402 -0.002985911 -0.00567
Lag2    0.03059642 -0.026294328  1.000000000 -0.025896670 -0.010853533 -0.00355
Lag3    0.03319458 -0.010803402 -0.025896670  1.000000000 -0.024051036 -0.01880
Lag4    0.03568872 -0.002985911 -0.010853533 -0.024051036  1.000000000 -0.02708
Lag5    0.02978799 -0.005674606 -0.003557949 -0.018808338 -0.027083641  1.00000
```

```
Volume 0.53900647  0.040909908 -0.043383215 -0.041823686 -0.048414246 -0.02200
Today  0.03009523 -0.026155045 -0.010250033 -0.002447647 -0.006899527 -0.03486
             Volume        Today
Year     0.53900647  0.030095229
Lag1     0.04090991 -0.026155045
Lag2    -0.04338321 -0.010250033
Lag3    -0.04182369 -0.002447647
Lag4    -0.04841425 -0.006899527
Lag5    -0.02200231 -0.034860083
Volume   1.00000000  0.014591823
Today    0.01459182  1.000000000
```

Cette matrice est donnée dans l'appendice (Annexe1) à titre illustratif. Nous constatons qu'il n'y a pas de corrélations entre les rendements journaliers (rendement en ce jour-là) et les variables *lag*. La seule corrélation est entre l'année *Year* et *Volume*. Nous allons appliquer ce que l'on a vu avant dans ce chapitre pour faire une régression logistique pour ces données afin de prédire la direction du marché chaque jour en utilisant les variables *lag1*, *lag2*,*lag3*,*lag4*,*lag5*, et *Volume*.

Ainsi, nous allons utiliser la fonction *glm* du langage R.

```
> glm.fit=glm(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume ,
      data=Smarket ,family =binomial )

> summary (glm.fit )

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
   Volume, family = binomial, data = Smarket)
```

```
Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.446  -1.203   1.065   1.145   1.326

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
Lag5         0.010313   0.049511   0.208    0.835
Volume       0.135441   0.158360   0.855    0.392

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1731.2  on 1249  degrees of freedom
Residual deviance: 1727.6  on 1243  degrees of freedom
AIC: 1741.6
```

Les coefficients de l'analyse linéaire discriminante sont donnés par

```
Coefficients of linear discriminants:
            LD1
Lag1 -0.6420190
Lag2 -0.5135293
```

Après avoir estimé les paramètres β_0 et β_1, on peut calculer la probabilité $\hat{p}(\mathbf{y}|\mathbf{x})$. En effet,

$$\hat{p}(\mathbf{x}) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1\mathbf{x}}}{1+e^{\hat{\beta}_0+\hat{\beta}_1\mathbf{x}}}.$$

La variable *Direction* est une variable qui montre si le marché est à la hausse ou à la baisse ce jour-là, elle est qualitative. Ainsi, elle faut la convertir en une variable binaire afin de lui ajuster un modèle de régression logistique. Ainsi, nous pouvons écrire

$$Direction = \begin{cases} 0 & \text{si, marché est à la baisse,} \\ 1 & \text{si, marché est à la hausse.} \end{cases}$$

Nous allons utiliser la fonction *glm*() pour ajuster une régression logistique aux données, mais il faut choisir la famille *binomiale* comme option dans cette fonction. La plus petite probabilitécritique est associée à la variable *lag1*, cette variable a un coefficient négatif, ce qui signifie que si l'indice a un rendement positif hier, il a moins de chance d'aller à la hausse aujourd'hui. La fonction *predict()* nous permet de prédire la probabilité d'une hausse ou d'une baisse du marché, étant donnés les valeurs des prédicteurs. Par exemple, $P(y = 1)$ donne la probabilité que le marché soit à la hausse. Les résultats des probabilités pour 20 jours consécutifs sont donnés comme suit

```
 > glm.probs = predict(glm.fit , type ="response" )
> glm.probs[1:20]
        1         2         3         4         5         6
0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565
7         8         9
0.4926509 0.5092292 0.5176135

       10        11        12        13        14        15
```

0.4888378	0.4965211	0.5197834	0.5183031	0.4963852	0.4864892
16	17	18		19	20
0.5153660	0.5053976	0.5319322		0.5167163	0.4983272

APPENDICE B

LA PREUVE DE LA PROPOSITION 4.4.2

Nous allons montrer que

$$\hat{\beta_j^C} = \arg\min_{\beta_j} Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{S(\frac{2}{\delta}\tilde{\beta}_j - \frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}, \lambda_1)}{\frac{2}{\delta} + \lambda_2}.$$

En effet,

$$Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{\sum_{i=1}^n \phi_c(r_i)}{n} + \frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}(\beta_j - \tilde{\beta}_j)}{n} + \frac{(\beta_j - \tilde{\beta}_j)^2}{\delta} + p_{\lambda_1,\lambda_2}(\beta_j).$$

Alors

$$\arg\min_{\beta_j} Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \arg\min_{\beta_j}\{\frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}(\beta_j - \tilde{\beta}_j) + \frac{1}{\delta}(\beta_j - \tilde{\beta}_j)^2 + p_{\lambda_1,\lambda_2}(\beta_j)\},$$

ensuite,

$$\arg\min_{\beta_j} Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \arg\min_{\beta_j}\{\frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}\beta_j - \frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}\tilde{\beta}_j + \frac{1}{\delta}\beta_j^2 - \frac{2}{\delta}\tilde{\beta}_j\beta_j + \frac{1}{\delta}\tilde{\beta}_j^{\,2} + p_{\lambda_1,\lambda_2}(\beta_j)\}. \quad \text{(B.1)}$$

Sachant que $\tilde{\boldsymbol{\beta}}$ et $\tilde{\beta}_0$ sont connus et que r_i est défini comme combinaison des deux quantités constantes, alors $\frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}\tilde{\beta}_j$ et $\frac{1}{\delta}\tilde{\beta}_j^{\,2}$ sont des constantes. Ce qui implique que

$$\begin{aligned}\arg\min_{\beta_j} Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) &= \arg\min_{\beta_j}\{\frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}\beta_j - \frac{1}{\delta}\beta_j^2 - \frac{2}{\delta}\tilde{\beta}_j\beta_j + p_{\lambda_1,\lambda_2}(\beta_j)\} \\ &= \arg\min_{\beta_j}\{\frac{1}{\delta}\beta_j^2 - [\frac{2}{\delta}\tilde{\beta}_j - \frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}]\beta_j + p_{\lambda_1,\lambda_2}(\beta_j)\}.\end{aligned}$$

Soit

$$Z = \frac{2}{\delta}\tilde{\beta}_j - \frac{\sum_{i=1}^{n} \phi_c'(r_i) y_i x_{ij}}{n}.$$

Pour minimiser la fonction $Q(.)$, il suffit de minimiser la fonction L qui est définie par

$$L(\beta_j) = \frac{1}{\delta}\beta_j^2 - Z\beta_j + p_{\lambda_1,\lambda_2}(\beta_j).$$

En effet, on a

$$p_{\lambda_1,\lambda_2}(\beta_j) = \lambda_1|\beta_j| + \frac{\lambda_2}{2}\beta_j^2,$$

alors

$$\begin{aligned} L(\beta_j) &= \frac{1}{\delta}\beta_j^2 - Z\beta_j + \lambda_1|\beta_j| + \frac{\lambda_2}{2}\beta_j^2 \\ &= (\frac{1}{\delta} + \frac{\lambda_2}{2})\beta_j^2 + \lambda_1|\beta_j| - Z\beta_j. \end{aligned}$$

Nous allons étudier trois cas :

1) Si $\beta_j > 0$ alors $|\beta_j| = \beta_j$.

   Donc, nous avons

$$L(\beta_j) = (\frac{1}{\delta} + \frac{\lambda_2}{2})\beta_j^2 + (\lambda_1 - Z)\beta_j.$$

   La dérivée première de L par rapport à β_j est donnée par

$$\frac{\partial L}{\partial \beta_j} = 2(\frac{1}{\delta} + \frac{\lambda_2}{2})\beta_j + (\lambda_1 - Z).$$

   Alors, nous avons

$$\frac{\partial L}{\partial \beta_j} = 0$$

   implique que si $Z > \lambda_1$, nous obtenons

$$\hat{\beta}_j = \frac{Z - \lambda_1}{\frac{2}{\delta} + \lambda_2}.$$

2) Si $\beta_j < 0$ alors $|\beta_j| = -\beta_j$.

Alors, la fonction L dans ce cas s'écrit comme suit

$$L(\beta_j) = (\frac{1}{\delta} + \frac{\lambda_2}{2})\beta_j^2 - (\lambda_1 + Z)\beta_j$$

et sa dérivée première est donnée par

$$\frac{\partial L}{\partial \beta_j} = 2(\frac{1}{\delta} + \frac{\lambda_2}{2})\beta_j - (\lambda_1 + Z).$$

Donc, si

$$\frac{\partial L}{\partial \beta_j} = 0,$$

et $Z < -\lambda_1$, nous aurons

$$\hat{\beta}_j = \frac{Z + \lambda_1}{\frac{2}{\delta} + \lambda_2}.$$

3) Si $\beta_j = 0$ alors $|Z| \leq \lambda_1$, ce qui donne

$$\hat{\beta}_j = 0.$$

Ainsi, nous allons résumer le tout

$$\hat{\beta_j^C} = \arg\min Q(\beta_j|\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = \frac{S(\frac{2}{\delta}\tilde{\beta}_j - \frac{\sum_{i=1}^n \phi_c'(r_i)y_i x_{ij}}{n}, \lambda_1)}{\frac{2}{\delta} + \lambda_2},$$

avec S est la fonction définie par l'équation (2.6).

APPENDICE C

COMPARAISON ENTRE LES MÉTHODES LASSO, RIDGE ET ÉLASTIQUE NET

C.1 Présentation de la base de données : le cancer de la prostate

Dans cette section, nous allons comparer les différentes techniques : *Lasso*, *Ridge*, Élastique Net et la régression multiple standard (de l'anglais *OLS*). Cette base de données est tirée d'une étude faite par Stamey *et al.* (1989). Cette étude examine le niveau de l'antigène spécifique de la prostate (*psa*) et un certain nombre de mesures cliniques chez des hommes qui étaient sur le point de recevoir une prostatectomie radicale. Cette base de données est disponible sur le paquet *lasso2* sur le logiciel R, les variables sont :

- *lcavol* = *log* (cancer volume) ;
- *lweight* = *log* (prostate weight) ;
- *age* = *age* ;
- *lcp* = *log* (capsular penetration) ;
- *lbph* = *log* (benign prostatic hyperplasia amount invasion) ;
- *svi* = seminal vesicle ;
- *gleason* = gleason score ;
- *pgg45* = percentage gleason scores 4 or 5 ;
- *lpsa* = *log* (prostate specific antigin).

C.2 Résultats de la comparaison

Nous allons utiliser le paquet *glmnet* qui est disponible sur R pour estimer le modèle. Notre but est de comparer les techniques suivantes

- la méthode *Lasso* : $\alpha = 1$;
- la méthode *Ridge* : $\alpha = 0$;
- élastique Net : $\alpha = 0.5$;
- la régression multiple standard (*OLS*).

Pour calculer les coefficients du modèle, nous avons fait une validation croisée avec 5 groupes pour choisir le coefficient de régularisation λ_1, qui nous donne l'erreur minimale en utilisant la fonction *cv.glmnet* disponible dans le paquet *glmnet* de R, la méthode est décrite dans l'article de Freidman *et al.* (2010). On trouve λ_{1min} (c.-à.d. la valeur de λ_1 qui correspond à la plus petite valeur de l'erreur de classement). Ensuite, on utilise la fonction *coef* pour calculer les coefficients qui convient à λ_{1min}. Les résultats de cette comparaison sont donnés par les graphiques C.1, C.2, C.3 et le tableau C.1.

Nous remarquons que les coefficients sont proches pour les différentes techniques. Nous constatons également que la technique Lasso pénalise les coefficients des variables qui ne contribuent pas dans le modèle par des zéros. Les variables *age*, *lcp* et *gleason*, sont des variables nuisibles.

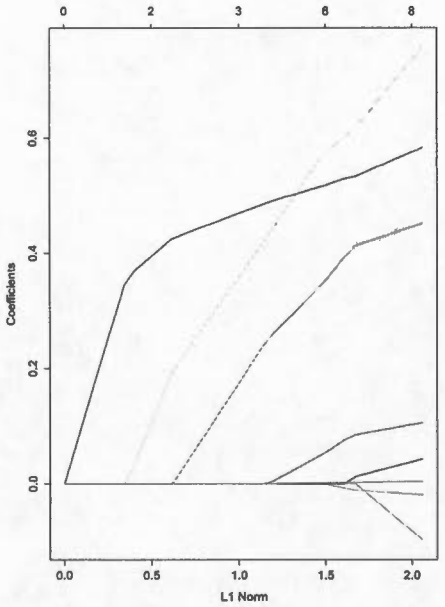

Figure C.1: Les chemins de solutions $\hat{\boldsymbol{\beta}}$ pour la méthode lasso. L'axe des abscisses est la norme $L1$ des coefficients $\hat{\boldsymbol{\beta}}$ et l'axe des ordonnées pour les coefficients $\hat{\boldsymbol{\beta}}$.

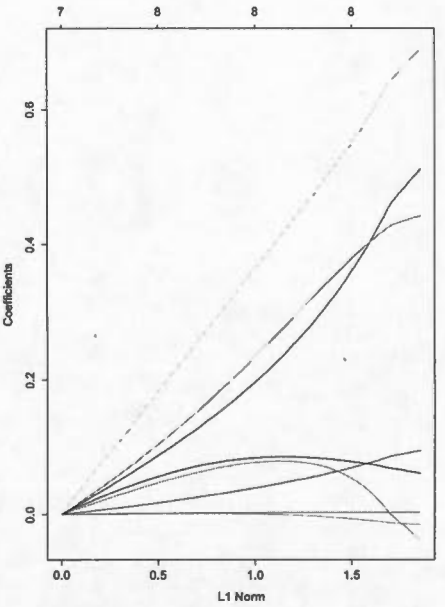

Figure C.2: Les chemins de solutions $\hat{\boldsymbol{\beta}}$ pour la méthode Ridge.

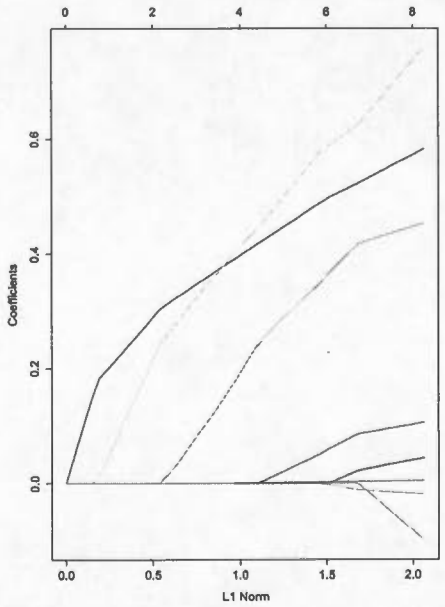

Figure C.3: Les chemins de solutions $\hat{\boldsymbol{\beta}}$ pour la méthode Élastique Net.

Tableau C.1: Les coefficients $\hat{\beta}$ pour les différentes techniques : Lasso, Ridge, Élastique Net et OLS.

Technique	Lasso	Ridge	Élastique Net	OLS
Intercept	0.4012	0.6617	0.4655	0.6
lcavol	0.5133	0.5792	0.5058	0.5784
lweigth	0.3359	0.4501	0.4409	0.4958
age	0.0000	-0.0152	-0.0148	-0.0175
lbph	0.0457	0.10	0.0942	0.0969
svi	0.55834	0.75	0.6850	0.774
lcp	0.0000	-0.0924	-0.0335	-0.1077
gleason	0.0000	0.0425	0.0624	0.1074
pgg45	0.0013	0.00434	0.0034	0.0034

RÉFÉRENCES

Antoniadis, A., Fan, J. et Gijbels, I. (2001). A wavelet method for unfolding sphere size distributions. *Canadian Journal of Statistics*, *29*(2), 251–268.

Bradley, P. et Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. Dans *Machine Learning Proceedings of the Fifteenth International Conference(ICML '98)*, pp. 82–90.

Breiman, L. (1995). Better rubset regression using the nonnegative garrote. *Technometrics*, *37*(4), 373–384.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. *et al.* (2004). Least angle regression. *The Annals of Statistics*, *32*(2), 407–499.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Friedman, J., Hastie, T. et Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Berlin : Springer Series in Statistics Springer.

Friedman, J., Hastie, T. et Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1.

James, G., Witten, D., Hastie, T. et Tibshirani, R. (2013). *An introduction to statistical learning*, volume 6. New York : Springer.

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. Dans *Proceeding of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 1065–1076.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*(336), 846–850.

Stamey, T. A., Kabalin, J. N., McNeal, J. E., Johnstone, I. M., Freiha, F., Redwine, E. A. et Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients. *The Journal of Urology*, *141*(5), 1076–1083.

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

Tseng, P. (2001). Convergence of a block coordinate descent method for non-differentiable minimization. *Journal of Optimization Theory and Applications*, *109*(3), 475–494.

Tseng, P. et Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, *117*(1-2), 387–423.

Wang, L., Zhu, J. et Zou, H. (2006). The doubly regularized support vector machine. *Statistica Sinica*, 589–615.

Wang, L., Zhu, J. et Zou, H. (2008). Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics*, *24*(3), 412–419.

Witten, A. S. et Zhang, F. (2014). The cluster elastic net for high-dimensional regression with unknown variable grouping. *Technometrics*, *33*(1), 112–121.

Yang, Y. et Zou, H. (2013). An efficient algorithm for computing the hhsvm and its generalizations. *Journal of Computational and Graphical Statistics*, *22*(2), 396–415.

Zou, H. et Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, *67*(2), 301–320.