Statistique en grande dimension

Frédéric Lavancier

Université de Nantes
M2 Ingénierie Statistique

2020/2021

Références

- ESL : "The elements of statistical learning", T. Hastie, R. Tibshirani, J. Friedman.
- "An introduction to statistical learning with applications in R", G. James, D. Witten, T. Hastie, R. Tibshirani.
- "Régression avec R", P-A. Cornillon, E. Matzner-Løber
- "Introduction to high-dimensional statistics", C. Giraud.

La plupart des figures de ce document sont tirées du premier ouvrage ESL ci-dessus, avec la permission des auteurs.

Contents

1 Introduction **4**

2 Rappels sur la régression linéaire **7**
2.1 La régression linéaire est une projection 7
2.2 Modélisation statistique 10
2.3 Qualité de l'estimation et validation du modèle 11
2.4 Aspects en grande dimension 14

3 Choix de modèles **17**
3.1 Erreurs de prévision 18
3.2 Critères usuels 22
3.2.1 C_p de Mallows 22
3.2.2 Critère AIC 23
3.2.3 Critère BIC 25
3.2.4 Comparaison des critères 26
3.3 Validation croisée 27
3.3.1 Hold out 27
3.3.2 Leave-one-out (LOO) 28
3.3.3 Leave-k-out 30
3.3.4 K-fold 31
3.4 Sélection automatique d'un sous-modèle 32
3.5 Estimation du modèle retenu 34

4 Réduction de dimension et régression sous contraintes **35**
4.1 Réduction de dimension : PCR et PLS 36
4.1.1 Régression sur composantes principales (PCR) 36
4.1.2 Régression des moindres carrés partiels (PLS) 39
4.2 Régression ridge 41

4.3 Régression Lasso 43
4.3.1 Principe 43
4.3.2 Résolution 46
4.3.3 Aparté `R` : comparaison entre `glmnet` et `lars` 52
4.3.4 Aspects théoriques 55
4.4 Quelques généralisations des régressions Ridge et Lasso 60
4.4.1 Elastic net 61
4.4.2 Gauss-Lasso 63
4.4.3 Adaptive Lasso 63
4.4.4 Logistic Lasso 64
4.4.5 Group-Lasso 65

5 Tests multiples **67**
5.1 Présentation du problème et notations 68
5.2 Principe des tests multiples : contrôler le $FWER$ ou le FDR 71
5.2.1 Procédure de Bonferroni 72
5.2.2 Procédure de Benjamini-Hochberg 73
5.2.3 p-values ajustées 78

Chapter 1

Introduction

Lors d'une étude statistique, on dispose généralement de n individus pour lesquels on a observé/relevé p variables (quantitatives ou qualitatives).

Qu'est-ce que la grande dimension? Plusieurs situations sont possibles selon que n et/ou p sont grands.

n grand, p de taille raisonnable

Exemple 1: Les clients d'une banque. Leur nombre n est très grand, mais leurs caractéristiques individuelles sont en nombre limité.

Exemple 2: "Data Lake" Renault. Chaque concessionnaire du monde entier saisit ses factures du jour en ligne. Le "Data Lake" de Renault contient ainsi les factures de chaque véhicule, pour chaque concessionnaire et chaque jour de l'année. Si on s'intéresse par exemple aux factures, leur nombre n est immense, mais les variables les caractérisant (nom du véhicule, type d'intervention, montant, etc) sont en nombre p limité (quelques dizaines au maximum).

C'est en général une situation favorable d'un point de vue théorique, car plus il y a d'individus et meilleures sont les estimations. De plus, si n est très grand, on peut utiliser sans risque les résultats asymptotiques disponibles, principalement le théorème limite central, pour quantifier la qualité des estimations ou des prévisions via des intervalles de confiance.

En fait, le principal problème lorsque n est très grand est d'ordre informatique. Le stockage peut ne plus être possible sur un simple ordinateur, ce qui complique l'accès aux données, et les calculs sont infaisables sur toute la population en un temps raisonnable. Des solutions informatiques de type

"Big Data" sont alors nécessaires (clusters de serveurs, données distribuées, technologie Hadoop) mais ce n'est pas le sujet de ce cours.

n de taille raisonnable, p grand

Exemple 1: Text Mining. Pour l'analyse des textes (par exemple issus de la satisfaction clients ou de réponses à un questionnaire), on liste tous les "mots" utilisés dans l'ensemble des textes, et on forme un tableau récapitulant pour chaque texte le nombre d'apparitions de chaque mot. Chaque colonne du tableau correspond à un mot et chaque ligne à un texte analysé. Dans ce contexte, n correspond au nombre de textes analysés et p au nombre total de mots, qui est généralement très grand.

Exemple 2: En génomique. Pour chaque individu, on dispose de "l'expression" de certains de ses gènes. Il s'agit d'une valeur numérique par gène. Le nombre p de gènes "exprimés" est de l'ordre de 10^5 ou 10^6. Etant donné n individus observés, l'objectif peut-être d'essayer de former des groupes homogènes en terme d'expression des gènes (classification non supervisée), ou en supposant qu'une partie des individus est malade et l'autre non, de tenter de trouver les gènes s'exprimant différemment pour les individus malades.

Exemple 3: En analyse de signaux. Lorsqu'on observe le résultat d'une analyse par spectrométrie, la réponse est une valeur de spectre par longueur d'ondes. Il y a en général plusieurs milliers de longueurs d'onde mesurées par individu.

Dans ces situations, on dispose d'un très grand nombre de variables p par rapport à n, par exemple $p > n/2$, voire même $p >> n$. Cette situation pose problème pour les analyses statistiques classiques, notamment pour la régression linéaire. En effet, supposons que l'on souhaite déterminer le lien entre une variable d'intérêt et les autres variables à disposition, on ne peut raisonnablement pas espérer estimer un modèle ayant plus de variables que d'observations disponibles. Deux éléments clés permettent néanmoins l'analyse statistique:

1. On suppose que parmi les nombreuses variables explicatives à disposition, seule une petite partie d'entre elles est en réalité utile pour le modèle (sans que l'on sache lesquelles a priori). Il s'agit de l'hypothèse fondamentale de parcimonie.

2. Des méthodes adaptées (choix de modèles, modèles contraints) permettent d'exploiter la parcimonie et d'estimer convenablement le modèle.

Ce contexte forme ce que l'on appelle généralement la statistique en grande dimension, et il s'agit de la situation dont traite principalement ce cours.

n et p grand

Il s'agit bien évidemment de la situation la plus compliquée : chaque individu est associé à un très grand nombre de variables, et il y a un très grand nombre d'individus observés. On peut penser aux exemples précédents (text mining, génomique, spectres) en imaginant dans chaque cas n très grand.

Dans cette situation, on utilise à la fois des outils informatiques adaptés au Big Data pour le stockage et l'accès aux données, et des outils statistiques de grande dimension pour intégrer le grand nombre de variables dans les modèles.

Chapter 2

Rappels sur la régression linéaire

L'objectif de ce chapitre n'est pas d'introduire la régression linéaire, que l'on suppose avoir déjà été étudiée, mais de rappeler ses principes de base, avec un regard particulier pour la situation où le nombre de variables explicatives est important, voire supérieur au nombre d'observations.

2.1 La régression linéaire est une projection

On dispose de n observations $y_1, \ldots, y_n$ de la variable à expliquer et de n observations $x_1, \ldots, x_n$ des variables explicatives. Pour tout i, $y_i \in \mathbb{R}$ est une valeur numérique réelle et $x_i \in \mathbb{R}^p$ est un vecteur de taille p.

On note

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \text{et} \quad X = \begin{pmatrix} x_1' \\ \vdots \\ x_n' \end{pmatrix} = \begin{pmatrix} x_{11} & \ldots & x_{p1} \\ \vdots & \vdots & \vdots \\ x_{1n} & \ldots & x_{pn} \end{pmatrix}.$$

Les p variables explicatives, observées sur les n individus, sont donc

$$X_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \end{pmatrix}, \quad \cdots \quad , \; X_p = \begin{pmatrix} x_{p1} \\ \vdots \\ x_{pn} \end{pmatrix}.$$

On suppose que X, de dimension (n, p), est de plein rang, c'est à dire $rang(X) = \min(n, p)$. Lorsque $p \leq n$, cela signifie qu'il n'existe aucune combinaison linéaire entre les variables explicatives. Lorsque $p > n$, cela

signifie que la famille des p variables explicatives génère $\mathbb{R}^n$, autrement dit il existe au moins n variables parmi les p qui sont non linéairement liées entre elles.

On note $[X] = \{X\alpha,\, \alpha \in \mathbb{R}^p\} = \{v \in \mathbb{R}^n,\, \exists \alpha \in \mathbb{R}^p, v = X\alpha\}$ l'espace vectoriel engendré par les colonnes de X. Il est de dimension $rang(X)$.

- Si $p < n$, $[X]$ est un sous espace vectoriel de $\mathbb{R}^n$ de dimension p.
- Si $p \geq n$, $[X] = \mathbb{R}^n$.

On note $[X]^\perp$ l'espace vectoriel orthogonal à $[X]$ dans $\mathbb{R}^n$, c'est à dire $[X]^\perp = \{v \in \mathbb{R}^n,\, X'v = 0\}$.

- Si $p < n$, il est de dimension $n - p$.
- Si $p \geq n$, $[X]^\perp = \{0\}$.

Régresser $Y \in \mathbb{R}^n$ sur les colonnes de X consiste à projeter Y sur $[X]$. On note $P_{[X]}$ la matrice de cette projection. On note également $P_{[X]^\perp}$ la matrice de projection sur $[X]^\perp$. On a la relation $P_{[X]} + P_{[X]^\perp} = I_n$, où I_n désigne la matrice identité de taille n

- Si $p < n$, $P_{[X]} = X(X'X)^{-1}X'$. Dans ce cas, $X'X$, matrice carrée de taille p, est de rang p et est donc bien inversible. De plus, $P_{[X]^\perp} = I_n - P_{[X]} = I_n - X(X'X)^{-1}X'$.
- Si $p \geq n$, $P_{[X]} = I_n$ et $P_{[X]^\perp} = I_n - P_{[X]} = 0_n$ (la matrice nulle de taille n).

Le projeté est noté $\hat{Y}$, i.e. $\hat{Y} = P_{[X]}Y$. Il est obtenu par la méthode des moindres carrés ordinaires (MCO) au sens où

$$\hat{Y} = \text{argmin}_{Z \in [X]} \|Y - Z\|^2$$

ou de façon équivalent $\hat{Y} = X\hat{\beta}$ avec

$$\hat{\beta} = \text{argmin}_{\beta \in \mathbb{R}^p} \|Y - X\beta\|^2.$$

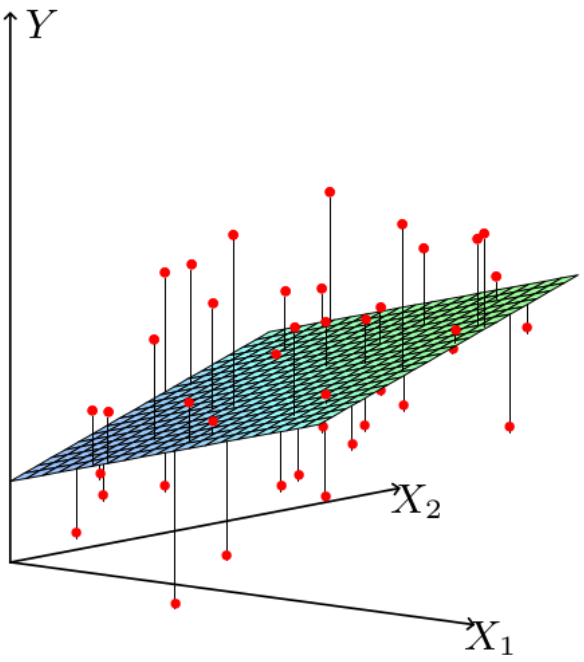

Figure 2.1: Figure extraite de l'ouvrage ESL montrant le nuage des points (x_i, y_i) en rouge lorsque x_i est de dimension $p = 2$, autrement dit $X = (X_1, X_2)$. Le plan représenté est celui des moindres carrés, auquel appartient les $\hat{y}_i$.

- Si $p < n$, $\hat{Y} = X\hat{\beta}$ où $\hat{\beta} = (X'X)^{-1}X'Y$. Cette solution est unique.
- Si $p \geq n$, $\hat{Y} = Y = X\hat{\beta}$ où $\hat{\beta}$ représente simplement les coordonnées de Y dans la famille génératrice de $\mathbb{R}^n$ formée des colonnes de X. Si $p = n$, $\hat{\beta}$ est unique car cette famille est alors une base de $\mathbb{R}^n$. Si $p > n$, il y a une infinité de façon de choisir $\hat{\beta}$ (par exemple en annulant $p - n$ coordonnées).

Les figures 2.1 et 2.2 illustrent le principe de la régression linéaire dans le cas $p = 2$.

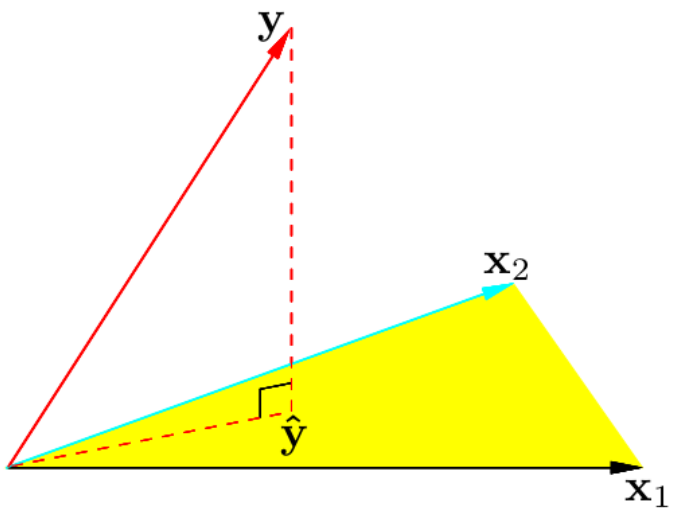

Figure 2.2: Figure extraite de l'ouvrage ESL. Le plan en jaune représente $[X]$ lorsque $p = 2$. Le vecteur Y est projeté sur $[X]$ pour donner $\hat{Y}$.

2.2 Modélisation statistique

Dans la partie précédente, il n'est pas question de loi de probabilité ni de statistique : étant donné un vecteur Y, on se contente de le projeter sur l'espace vectoriel $[X]$.

Le cadre statistique est le suivant : on suppose que chaque observation y_i est la réalisation d'une variable aléatoire ayant pour espérance $x_i'\beta$ pour un certain $\beta \in \mathbb{R}^p$ et ayant même variance σ^2. De plus les y_i sont supposés non corrélées entre eux. Ces hypothèses reviennent à affirmer que

$$Y = X\beta + \epsilon \tag{2.1}$$

où $\epsilon = Y - X\beta$ est un vecteur aléatoire d'espérance nulle et de covariance $\sigma^2 I_n$. Dans ce formalisme les variables explicatives X sont supposées non aléatoires.

Dit autrement, on suppose que $\mathbb{E}(Y) \in [X]$ mais Y n'appartient pas à $[X]$ (sauf si $[X] = \mathbb{R}^n$, le cas $p \geq n$). Supposer que $Y \in [X]$ serait une hypothèse extrêmement forte et peu réaliste. Dans ce cadre, régresser Y sur $[X]$ revient donc à estimer les coordonnées de $\mathbb{E}(Y)$ dans $[X]$.

L'intérêt d'une telle modélisation peut être de deux types :

- Descriptif : comprendre le lien entre $\mathbb{E}(Y)$ et les variables explicatives (quelles sont les variables qui, en moyenne, influencent le plus Y et comment?). Cela nécessite une estimation précise de β dans le sens où $\mathbb{E}\|\hat{\beta} - \beta\|^2$ doit être la plus petite possible.

- Prédictif : étant donné un nouvel individu o dont on connait les valeurs x_o des variables explicatives, on prédit y_o via l'estimation de $\mathbb{E}(y_o)$ c'est à dire $\hat{y}_o = x_o'\hat{\beta}$. Une bonne prédiction doit minimiser $\mathbb{E}(y_o - \hat{y}_o)^2$. Il est à noter qu'une prévision naïve de $\mathbb{E}(y_o)$ aurait pu être simplement la moyenne empirique $\frac{1}{n}\sum_{i=1}^n y_i$. Cette prévision est en général fortement biaisée car les y_i n'ont pas même espérance.

Remarque 2.2.1. *Si la relation* (2.1) *n'est pas vérifiée, la régression de Y sur $[X]$ peut toujours être effectuée pour obtenir $\hat{Y} = P_{[X]}Y$. Mais dans ce cas $\hat{Y}$ est généralement un mauvais estimateur de $\mathbb{E}(Y)$. Trouver la bonne matrice X (c'est à dire choisir les bonnes variables explicatives) telle que* (2.1) *est vérifiée est donc primordial. C'est l'objet de la sélection de modèles abordée en section 2.3 et discutée plus en détail dans le chapitre suivant.*

2.3 Qualité de l'estimation et validation du modèle

On suppose dans cette partie que la relation (2.1) est vérifiée avec ϵ centré de variance $\sigma^2 I_n$. On suppose de plus que $p < n$ et X est de plein rang, i.e. $rang(X) = p$. Ces hypothèses sont le cadre d'étude classique (favorable) de la régression linéaire. On résume les principales propriétés qui en découlent ci-dessous. Le cas $p \geq n$ est discuté en section 2.4.

Dans le cadre précédent, on a les propriétés suivantes:

- $\hat{\beta} = (X'X)^{-1}X'Y$ est sans biais, i.e. $\mathbb{E}(\hat{\beta}) = \beta$, et de variance $Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$.

- $\hat{\beta}$ est le meilleur estimateur linéaire sans biais de β au sens du coût quadratique (Théorème de Gauss-Markov).

- Si l'on suppose de plus que ϵ suit une loi normale, alors $\hat{\beta}$ correspond à l'estimateur du maximum de vraisemblance de β.

Les résidus de la régression sont $\hat{\epsilon} = Y - \hat{Y}$. Leur norme correspond à la somme des carrés des résidus : $SCR = \|\hat{\epsilon}\|^2$. Elle permet d'estimer sans biais la variance σ^2 de ϵ :

$$\hat{\sigma}^2 = \frac{SCR}{n-p} = \frac{1}{n-p}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

Les résidus sont à la base de toutes les procédures de test et d'analyse de la qualité de la modélisation, résumées ci-dessous.

Les principaux tests :

- Le test de Student de significativité du paramètre β_j consiste à tester $H_0 : \beta_j = 0$ contre $H_1 : \beta_j \neq 0$. Il utilise la statistique

$$T = \frac{\hat{\beta}_j}{\hat{\sigma}_{\hat{\beta}_j}}$$

où $\hat{\sigma}_{\hat{\beta}_j}$ désigne l'écart-type de $\hat{\beta}_j$ qui vaut $\hat{\sigma}_{\hat{\beta}_j} = \hat{\sigma}\sqrt{(X'X)^{-1}_{jj}}$, c'est à dire la racine carrée du j-ème terme de la diagonale de $Var(\hat{\beta})$.

La région critique du test au niveau α est $RC_\alpha = \{|T| > t_{n-p}(1-\alpha/2)\}$ où $t_{n-p}(1-\alpha/2)$ est le quantile d'ordre $1-\alpha/2$ d'une loi de Student de degré de liberté $n-p$. Le niveau du test est exactement α si la loi de ϵ est une loi normale, et vaut asymptotiquement α (i.e. lorsque $n \to \infty$) sinon (et sous certaines hypothèses de régularité).

- Le test de Fisher de contraintes linéaires sur les paramètres consiste à tester $H_0 : R\beta = 0$ contre $H_1 : R\beta \neq 0$ où R est une matrice de q contraintes de taille (q, p). Un cas particulier courant est le test des modèles emboités où les contraintes sont la nullité simultanée de q paramètres dans le modèle général. La statistique de test est

$$F = \frac{n-p}{q}\frac{SCR_c - SCR}{SCR},$$

où SCR désigne la SCR du modèle sans contrainte et SCR_c celle du modèle contraint. Si ϵ suit une loi normale, une région critique au niveau α est $RC_\alpha = \{F > f_{q,n-p}(1-\alpha)\}$ où $f_{q,n-p}(1-\alpha)$ est le quantile d'ordre $1-\alpha$ d'une loi de Fisher de degré de liberté $(q, n-p)$. De même le niveau vaut asymptotiquement α si ϵ ne suit pas une loi normale (et sous certaines hypothèses de régularité).

- Sans entrer dans les détails, nous mentionnons le test d'homoscédasticité de Breusch-Pagan (la matrice de variance de ϵ est-elle bien constante sur la diagonale?), et les tests d'auto-corrélation de Durbin-Watson et de Breusch-Godfrey (la matrice de variance de ϵ est-elle bien diagonale?).

Les mesures usuelles de la qualité de modélisation :

- Le coefficient de corrélation multiple (ou coefficient de détermination), dans le cas où X contient la colonne constante égale à 1 (autrement dit si le modèle contient une constante) :

$$R^2 = \frac{SCE}{SCT} = 1 - \frac{SCR}{SCT}$$

où $SCE = \|\hat{Y} - \bar{Y}\|^2$ est la somme des carrés expliqués et $SCT = \|Y - \bar{Y}\|^2$ est la somme des carrés totaux. Lorsque le modèle ne contient pas de constante, $R^2 = 1 - SCR/\|Y\|^2$. Plus le R^2 est proche de 1, plus le modèle s'ajuste bien aux données.

- Le R^2 corrigé (ajusté) :

$$R^2_a = 1 - \frac{n-1}{n-p}\frac{SCR}{SCT}$$

dans le cas où le modèle contient une constante. Le R^2_a s'interprète comme le R^2 sans souffrir de son principal défaut, qui est que le R^2 croît nécessairement lorsqu'on ajoute une variable au modèle même si cette dernière est non significative.

Les critères usuels de choix de modèles : si on hésite entre plusieurs modèles, le but étant de trouver le modèle le plus parcimonieux qui vérifie (2.1), on peut comparer les critères suivants (voir le chapitre suivant pour plus de détails)

- R_a^2 : on privilégie le modèle ayant le R_a^2 le plus élevé.
- Le C_p de Mallows
$$C_p = \frac{SCR}{\hat{\sigma}^2} - n + 2p$$
où $\hat{\sigma}^2$ est un estimateur sans biais de σ^2 (en général celui calculé sur le plus gros modèle) et on privilégie le modèle ayant le C_p le plus faible.
- le critère AIC
$$AIC = n \log \frac{SCR}{n} + 2(p+1)$$
et on privilégie le modèle ayant le AIC le plus faible.
- le critère BIC
$$BIC = n \log \frac{SCR}{n} + (p+1) \log n$$
et on privilégie le modèle ayant le BIC le plus faible.

Enfin, on peut procéder à un choix automatique basé sur l'un des critères précédents à l'aide d'une procédure pas à pas ascendante (forward stepwise selection) de la façon suivante :

1. On part du modèle le plus petit (contenant uniquement la constante)
2. On ajoute la variable conduisant à la meilleure amélioration possible selon le critère choisi (par exemple la plus grande diminution du BIC)
3. On recommence jusqu'à ce qu'aucun nouvel ajout de variable n'améliore plus le critère.

De la même manière, on peut effectuer une procédure pas à pas descendante (backward stepwise selection) en partant du plus gros modèle et en enlevant successivement la "pire" variable.

2.4 Aspects en grande dimension

Il est intéressant d'observer ce qui se passe en grande dimension : si $p > n$, le modèle linéaire
$$Y = X\beta + \epsilon$$

est mal défini puisqu'il existe une infinité de coefficients β conduisant au même vecteur $v = X\beta$. Par exemple v peut s'exprimer en fonction des n premières colonnes de X, ou des n dernières, etc.

Chercher à estimer β n'a donc pas vraiment de sens. D'ailleurs, $\hat{\beta}$ obtenu par les MCO n'est pas non plus unique, voir la section 2. Aborder l'analyse d'un tel modèle en toute généralité parait donc vain.

La situation devient plus raisonnable si l'on suppose que dans la relation linéaire précédente, la plupart des coefficient β_j sont en fait nuls. Pour fixer les idées, supposons que seuls p^* coefficients β_j sont non nuls, avec $p^* << n$. Il s'agit d'une **hypothèse de parcimonie (sparsity) cruciale** permettant l'analyse statistique en grande dimension. Elle revient à supposer que $X\beta$ appartient en réalité à un sous-espace vectoriel $[X^*]$ de $[X]$ de dimension $p^* << n$, où X^* désigne la sous matrice de X ne contenant que les p^* colonnes associées aux coefficients β_j non-nuls.

Si on connaissait X^*, on serait dans la situation favorable de la section 2.3. La difficulté réside dans le fait que l'on ne connait pas p^* et encore moins X^*. Il convient donc d'utiliser des outils pour trouver X^* à partir de X et estimer le lien linéaire associé.

Concernant les critères standards de sélection de modèles présentés dans la partie précédente, ils s'appuient tous sur les résidus. Mais si l'on part d'un modèle ayant $p \geq n$ variables, alors $\hat{Y} = Y$ et donc $\hat{\epsilon} = 0$, et ce même en absence de lien réel entre Y et X. En conséquence, pour un tel modèle :

- $R^2 = 1$,
- R^2_a n'a plus de sens (car $n - p < 0$),
- C_p n'a plus de sens (car à la fois le SCR du modèle estimé et $\hat{\sigma}^2$ calculé sur le modèle le plus gros sont nuls),
- $AIC = -\infty$,
- $BIC = -\infty$.

Toutes les méthodes de sélection de modèles présentées précédemment ne sont donc plus utilisables.

Il y a néanmoins une exception : la sélection forward. En effet cette dernière part du plus petit modèle, pour lequel les critères de sélection sont calculables (à part le C_p de Mallows qui s'appuie sur l'estimation de $\hat{\sigma}^2$ à partir du plus gros modèle). Ces critères restent calculables et comparables

pour les premières étapes de l'algorithme pour lesquelles un nombre limité de variables a été ajouté. Par l'hypothèse de parcimonie, on peut espérer que cet algorithme s'arrête avant d'avoir inclus un nombre trop important de variables dans le modèle.

Ainsi en présence d'un grand nombre de variables explicatives ($p > n$ mais également $p \leq n$ avec p grand), le modèle étant supposé par ailleurs parcimonieux, il est nécessaire d'introduire des méthodes alternatives aux MCO afin d'estimer au mieux le modèle linéaire. C'est ce que nous verrons au chapitre 4. Mais quelle que soit la méthode utilisée, il conviendra d'adopter une stratégie de sélection de modèles. Cette étape est commune à toute modélisation statistique, qu'elle concerne un modèle linéaire, logistique, ou autre. Le chapitre suivant clarifie les méthodes usuelles pour y parvenir.

Chapter 3

Choix de modèles

Pour fixer les idées on se place dans le cadre d'un modèle de régression linéaire. Comme on l'a vu dans le premier chapitre, la présence de nombreuses variables explicatives (p grand par rapport à n) rend l'estimateur par moindres carrés peu fiable car sa variance explose (en effet, si X de dimension (n, p) est de rang $n < p$, alors $X'X$ est une matrice (p, p) de rang n et est donc non inversible). En particulier les prévisions basées sur l'estimateur par MCO sont mauvaises. Ce phénomène n'est pas propre à la régression linéaire mais s'observe dans la plupart des modèles statistiques en grande dimension. Il est assez naturel dans ce contexte de supposer de la parcimonie : un faible nombre de variables sont en réalité pertinentes dans le modèle, mais on ne sait pas a priori lesquelles.

La façon la plus naturelle de procéder dans ce contexte est de choisir au mieux un sous-modèle. Cela nécessite d'estimer pour chaque modèle l'erreur associée, ce qui est détaillée dans la partie 3.1. Différentes approches existent pour estimer cette erreur : des critères, comme AIC ou BIC, dont la construction est rappelée dans la partie 3.2, et la validation croisée discutée dans la partie 3.2.4. Les méthodes de sélection consistent à parcourir un grand nombre de sous-modèles possibles (idéalement tous) et de retenir celui dont l'erreur estimée est la plus faible. La partie 3.4 résume les approches classiques.

De façon alternative à la sélection d'un sous-modèle, d'autres techniques permettent d'aborder l'estimation de modèles en grande dimension en exploitant la parcimonie. Le chapitre suivant traitera de la réduction de dimension et des estimations sous contraintes.

3.1 Erreurs de prévision

Pour fixer les idées on se place dans le cadre d'un modèle de régression linéaire, mais les notions développées sont valables pour la plupart des modèles considérés en statistique.

On suppose que le vrai modèle (inconnu) expliquant Y est

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

où X^* est une matrice non aléatoire de taille (n, p^*) contenant p^* variables explicatives, $\beta^* \in \mathbb{R}^{p^*}$ est le paramètre inconnu et $\epsilon \in \mathbb{R}^n$ est l'erreur de modélisation, aléatoire, supposée centrée et de variance $\sigma^2 I_n$.

On ne connait pas a priori les "vraies" variables explicatives et on effectue donc la régression linéaire de Y sur p variables explicatives regroupées dans la matrice X de taille (n, p). En toute généralité, p n'est pas nécessairement égale à p^*, et même lorsque c'est le cas les variables explicatives dans X et X^* ne sont pas nécessairement les mêmes. On obtient donc

$$\hat{Y} = P_{[X]}Y = X\hat{\beta}.$$

Il est à noter que $\hat{\beta}$ n'a pas nécessairement la même taille que β^*. Pour évaluer la qualité d'estimation on s'intéresse donc à $X\hat{\beta} - X^*\beta^*$ dont les deux termes ont la même dimension. Mais dans une optique de prévision on s'intéresse plutôt aux résidus $Y - \hat{Y}$. Plusieurs erreurs quadratiques moyennes peuvent être considérées.

1. L'erreur sur l'échantillon d'apprentissage (la MSE usuelle) est

$$R_{\text{train}} = \frac{1}{n}\mathbb{E}(\|Y - \hat{Y}\|^2) = \frac{1}{n}\mathbb{E}(\|Y - X\hat{\beta}\|^2).$$

   Cette erreur nous informe de la qualité de l'ajustement sur les données utilisées, mais ne nous dit rien sur ses qualités de prévision sur de nouvelles données (échantillon test). Or c'est ce dernier point qui est généralement intéressant.

2. L'erreur test "in" est l'erreur de prévision sur un nouvel échantillon construit ainsi : on considère les mêmes valeurs des variables explicatives que dans l'échantillon d'apprentissage, mais un nouveau vecteur $\tilde{Y}$, indépendant de Y et de même loi, est associé. Autrement dit,

$\tilde{Y} = X^*\beta^* + \tilde{\epsilon}$ où $\tilde{\epsilon}$ est indépendant et de même loi que ϵ. Comme précédemment, on ne connait pas X^* et la prévision se base sur les mêmes variables explicatives X que dans l'échantillon d'apprentissage. La prévision de $\tilde{Y}$ est donc la même que celle de Y, i.e. $X\hat{\beta}$, la différence importante étant que $\hat{\beta}$, construit à partir de Y, est indépendant de $\tilde{Y}$. L'erreur "in" est donc

$$R_{\text{in}} = \frac{1}{n}\mathbb{E}(\|\tilde{Y} - X\hat{\beta}\|^2),$$

où $\tilde{Y}$ est indépendant et de même loi que Y.

3. Enfin l'erreur de prévision la plus générale, appelée erreur test, s'intéresse à la qualité de prévision en présence à la fois de nouvelles valeurs $\tilde{X}^*$ et $\tilde{X}$ pour les variables explicatives, et d'une réalisation $\tilde{Y}$ indépendante de Y. Les matrices $\tilde{X}^*$ et $\tilde{X}$, de taille respective (n, p^*) et (n, p), contiennent les mêmes variables explicatives que X^* et X mais pour n valeurs différentes. Autrement dit $\tilde{Y} = \tilde{X}^*\beta^* + \tilde{\epsilon}$ où $\tilde{\epsilon}$ est indépendant et de même loi que ϵ, et la prévision se base sur $\tilde{X}$. La prévision de $\tilde{Y}$ est alors $\tilde{X}\hat{\beta}$ et l'erreur associée

$$R_{\text{test}} = \frac{1}{n}\mathbb{E}(\|\tilde{Y} - \tilde{X}\hat{\beta}\|^2).$$

Cette erreur dépend de $\tilde{X}$ et $\tilde{X}^*$, i.e. $R_{\text{test}} = R_{\text{test}}(\tilde{X}, \tilde{X}^*)$, et on remarque que $R_{\text{in}} = R_{\text{test}}(X, X^*)$.

Proposition 3.1.1. *On a les relations suivantes.*

$$\begin{aligned} R_{\text{test}} &= \sigma^2 + \frac{1}{n}\|\tilde{X}^*\beta^* - \mathbb{E}(\tilde{X}\hat{\beta}))\|^2 + \frac{1}{n}tr(Var(\tilde{X}\hat{\beta})) \\ &= \frac{1}{n}\sum_{i=1}^{n}(\sigma^2 + B_i^2 + V_i) \end{aligned}$$

où $B_i = \tilde{x}_i^{*\prime}\beta^* - \mathbb{E}(\tilde{x}_i'\hat{\beta})$ et $V_i = Var(\tilde{x}_i'\hat{\beta})$ sont respectivement un terme de biais et un terme de variance pour la prévision de $\tilde{y}_i$.

Si $p \leq n$,

$$R_{\text{train}} = \frac{1}{n}\|P_{[X]^\perp}X^*\beta^*\|^2 + \frac{n-p}{n}\sigma^2, \quad R_{\text{in}} = \sigma^2 + \frac{1}{n}\|P_{[X]^\perp}X^*\beta^*\|^2 + \frac{p}{n}\sigma^2$$

Si $p \geq n$,

$$R_{\text{train}} = \frac{1}{n}\|P_{[X]^\perp} X^* \beta^*\|^2, \quad R_{\text{in}} = 2\sigma^2 + \frac{1}{n}\|P_{[X]^\perp} X^* \beta^*\|^2.$$

Ainsi

$$R_{\text{in}} = R_{\text{train}} + 2\frac{\min(n,p)}{n}\sigma^2. \tag{3.1}$$

Ces résultats conduisent aux observations suivantes.

- Le risque le plus général R_{test} est la somme de trois termes : une erreur incompressible σ^2, due à la présence du bruit $\tilde{\epsilon}$ dans les nouvelles données, qui est imprévisible; un terme de biais dû au fait qu'on a effectué la régression sur les mauvaises variables X au lieu de $\tilde{X}$ (ce terme disparaît si le bon choix a été effectué); et un terme de variance dû à l'estimateur $\hat{\beta}$. Le meilleur modèle satisfait donc un bon compromis biais-variance.

- Le risque R_{in}, qui est un cas particulier de R_{test} lorsque $\tilde{X} = X$ et $\tilde{X}^* = X^*$, fait intervenir de la même manière un terme de biais et de variance, qui sont rendus plus explicites dans ce contexte.

- En analysant R_{in}, on s'aperçoit qu'un gros modèle, dans le sens où $X^*\beta^* \in [X]$, a un biais nul, car alors $P_{[X]^\perp} X^* \beta^* = 0$. En revanche, il aura une forte variance car p est alors grand. Au contraire, un petit modèle (p petit) aura une faible variance mais un biais potentiellement élevé.

- Pour choisir un bon modèle, il conviendrait d'estimer R_{in} ou mieux R_{test}. Mais la quantité naturelle à laquelle on a accès est le risque empirique $SCR = \|Y - \hat{Y}\|^2$. Ce dernier est un estimateur sans biais de nR_{train}, i.e. $\mathbb{E}(SCR/n) = R_{\text{train}}$ par définition, et non des risques de prévision plus réalistes R_{in} et R_{test}.

- Le risque R_{train} n'a pas la même dynamique que R_{in} et R_{test} : son terme de biais disparaît bien lorsque le modèle est gros ($X^*\beta^* \in [X]$), mais son terme de variance aussi (il décroit avec p). Ainsi, baser le choix du modèle sur R_{train} conduirait à choisir le plus gros modèle possible, celui qui interpole les données et implique $SCR = 0$.

- Grâce à la relation (3.1), on peut proposer un estimateur sans biais de R_{in} : $SCR/n + 2\min(n,p)\,\hat{\sigma}^2/n$, pourvu que $\hat{\sigma}^2$ soit un estimateur sans biais de σ^2. C'est la motivation à la base du C_p de Mallows, voir la section suivante. Le risque R_{test} est par contre trop général pour être estimé convenablement à partir de SCR.

Remarque 3.1.2. *L'égalité* (3.1) *est vraie pour toute matrice X fixée déterministe et motive le C_p de Mallows. Néanmoins, une fois le modèle choisi grâce au C_p de Mallows (ou par tout autre critère basé sur les données), l'égalité* (3.1) *devient fausse, autrement dit l'estimation de R_{in} a posteriori (c'est à dire après avoir sélectionné le modèle) en utilisant $SCR/n + 2p\hat{\sigma}^2/n$ devient biaisé, même si $\hat{\sigma}^2$ est un estimateur sans biais. En effet, choisir un modèle, c'est à dire X, par un critère de sélection, rend le choix de X aléatoire et dépendant de $\hat{\beta}$ (puisque les critères en dépendent). Ainsi dans le calcul de R_{in} conduisant à* (3.1)*, la propriété $\mathbb{E}(P_{[X]^\perp}Y) = P_{[X]^\perp}\mathbb{E}(Y)$ devient fausse si X a été choisie a posteriori. En fait, l'estimation de R_{in} a posteriori en utilisant le C_p de Mallows sous estime R_{in}. Ce phénomène est appelé biais de sélection.*

Proof. On a

$$\begin{aligned}
nR_{\text{train}} &= \mathbb{E}(\|P_{[X]^\perp}Y\|^2) \\
&= \mathbb{E}(\|P_{[X]^\perp}X^*\beta^* + P_{[X]^\perp}\epsilon\|^2) \\
&= \|P_{[X]^\perp}X^*\beta^*\|^2 + \mathbb{E}(\|P_{[X]^\perp}\epsilon\|^2) \\
&= \|P_{[X]^\perp}X^*\beta^*\|^2 + \mathbb{E}((P_{[X]^\perp}\epsilon)'(P_{[X]^\perp}\epsilon)) \\
&= \|P_{[X]^\perp}X^*\beta^*\|^2 + \mathbb{E}(tr((P_{[X]^\perp}\epsilon)(P_{[X]^\perp}\epsilon)')) \\
&= \|P_{[X]^\perp}X^*\beta^*\|^2 + tr(P_{[X]^\perp}\mathbb{E}(\epsilon\epsilon')P_{[X]^\perp}) \\
&= \|P_{[X]^\perp}X^*\beta^*\|^2 + \sigma^2 tr(P_{[X]^\perp}) \\
&= \begin{cases} \|P_{[X]^\perp}X^*\beta^*\|^2 + (n-p)\sigma^2 & \text{si} \quad p \le n, \\ \|P_{[X]^\perp}X^*\beta^*\|^2 & \text{si} \quad p \ge n. \end{cases}
\end{aligned}$$

$$nR_{\text{test}} = \mathbb{E}(\|(\tilde{Y} - \mathbb{E}(\tilde{Y})) + (\mathbb{E}(\tilde{Y}) - \mathbb{E}(\tilde{X}\hat{\beta})) + (\mathbb{E}(\tilde{X}\hat{\beta}) - \tilde{X}\hat{\beta})\|^2)$$

En développant le carré, on observe que les termes croisés s'annulent car par indépendance de $\tilde{Y}$ et $\hat{\beta}$

$$\mathbb{E}[(\tilde{Y} - \mathbb{E}(\tilde{Y}))'(\mathbb{E}(\tilde{X}\hat{\beta}) - \tilde{X}\hat{\beta})] = [\mathbb{E}(\tilde{Y} - \mathbb{E}(\tilde{Y}))]'[\mathbb{E}(\mathbb{E}(\tilde{X}\hat{\beta}) - \tilde{X}\hat{\beta})] = 0$$

et les autres termes croisés sont le produit d'un terme déterministe par l'espérance d'une variable centrée. Ainsi

$$\begin{aligned} nR_{\text{test}} &= \mathbb{E}(\|\tilde{Y} - \mathbb{E}(\tilde{Y})\|^2) + \|\mathbb{E}(\tilde{Y}) - \mathbb{E}(\tilde{X}\hat{\beta}))\|^2 + \mathbb{E}(\|\mathbb{E}(\tilde{X}\hat{\beta}) - \tilde{X}\hat{\beta}\|^2) \\ &= n\sigma^2 + \|\tilde{X}^*\beta^* - \mathbb{E}(\tilde{X}\hat{\beta}))\|^2 + tr(Var(\tilde{X}\hat{\beta})). \end{aligned}$$

En choisissant $\tilde{X} = X$ et $\tilde{X}^* = X^*$ dans l'expression de R_{test}, on obtient R_{in} et on en déduit

$$\begin{aligned} nR_{\text{in}} &= n\sigma^2 + \|X^*\beta^* - \mathbb{E}(X\hat{\beta}))\|^2 + tr(Var(X\hat{\beta})) \\ &= n\sigma^2 + \|\mathbb{E}(Y) - \mathbb{E}(P_{[X]}Y)\|^2 + tr(Var(P_{[X]}Y)) \\ &= n\sigma^2 + \|\mathbb{E}(P_{[X]^\perp}Y)\|^2 + tr(P_{[X]}Var(Y)P'_{[X]}) \\ &= n\sigma^2 + \|P_{[X]^\perp}X^*\beta^*\|^2 + \sigma^2 tr(P_{[X]}) \\ &= \begin{cases} n\sigma^2 + \|P_{[X]^\perp}X^*\beta^*\|^2 + p\sigma^2 & \text{si} \quad p \leq n, \\ n\sigma^2 + \|P_{[X]^\perp}X^*\beta^*\|^2 + n\sigma^2 & \text{si} \quad p \geq n. \end{cases} \end{aligned}$$

□

3.2 Critères usuels

3.2.1 C_p de Mallows

Pour choisir un bon modèle de prévision, l'idéal est de minimiser l'erreur test R_{test}. Cette erreur très générale est difficile à estimer. De façon plus raisonnable, on peut estimer R_{in} qui correspond au cas particulier de R_{test} lorsqu'on considère les mêmes valeurs des variables explicatives X. D'après (3.1), on utilise l'estimateur naturel

$$\hat{R}_{\text{in}} = \frac{SCR}{n} + 2\frac{\min(n,p)}{n}\hat{\sigma}^2$$

qui est un estimateur sans biais dès que $\hat{\sigma}^2$ est sans biais. Pour cette raison, $\hat{\sigma}^2$ n'est pas calculé à partir du modèle testé (d'où est issue la SCR) mais provient généralement du plus gros modèle disponible $X_{\max}$. Le point clé est que ce dernier vérifie $X^*\beta^* \in [X_{\max}]$. En supposant que ce modèle contient $p_{\max}$ variables avec $p_{\max} < n$, et en notant $SCR_{\max}$ la SCR correspondante, cela donne

$$\hat{\sigma}^2 = \frac{SCR_{\max}}{n - p_{\max}}.$$

Cet estimateur est bien sans biais car $\mathbb{E}(\hat{\sigma}^2) = nR_{\text{train}}/(n - p_{\max})$ et d'après la proposition 3.1.1, $nR_{\text{train}} = \|P_{[X_{\max}]^\perp} X^*\beta^*\|^2 + (n - p_{\max})\sigma^2 = (n - p_{\max})\sigma^2$ car $X^*\beta^* \in [X_{\max}]$.

Dans le cas $p_{\max} < n$ (et donc $p < n$), minimiser $\hat{R}_{\text{in}}$ revient à minimiser

$$C_p = \frac{SCR}{\hat{\sigma}^2} - n + 2p$$

qui est l'expression usuelle du C_p de Mallows implémenté dans les logiciels. Si $p_{\max} \geq n$, l'estimation de σ^2 à partir du plus gros modèle ne peut pas se faire comme précédemment, car alors $SCR_{\max} = 0$. Le C_p n'est pas défini dans ce cas.

Le défaut principal du C_p de Mallows est la nécessité de trouver un estimateur sans biais de σ^2. L'approche précédente requiert $p_{\max} < n$, ce qui empêche le calcul du critère en grande dimension $p_{\max} \geq n$, même si le modèle testé ne contient qu'un nombre p faible de variables. De plus, si $p_{\max} < n$ mais $p_{\max}$ est grand, l'estimateur $\hat{\sigma}^2$ sera certes sans biais, mais sa variance sera élevée ce qui impliquera une mauvaise estimation de R_{in} et donc un mauvais choix de modèle. Le C_p de Mallows est donc à utiliser uniquement en présence d'un nombre total modéré de variables explicatives.

3.2.2 Critère AIC

L'AIC (Akaike Infomation Criterion) est un critère construit à partir de la vraisemblance. Supposons que le modèle soit Gaussien (ϵ suit une loi Gaussienne). Pour un modèle donné faisant intervenir comme variables explicatives la matrice X, l'estimateur du maximum de vraisemblance maximise la log-vraisemblance $\log V_X(Y;\theta)$ où V_X désigne la vraisemblance du modèle considéré et $\theta = (\beta, \sigma^2)$ est le vecteur des paramètres inconnus du modèle. On obtient comme solution dans le cas Gaussien l'estimateur des MCO $\hat{\beta}$ et $\hat{\sigma}^2 = SCR/n$. Ainsi $-\log V_X(Y;\hat{\theta})$ est minimal pour ce modèle.

Pour mesurer la qualité du modèle, l'idée est de considérer un échantillon test indépendant $\tilde{Y}$ et d'observer ce que vaut en moyenne $-\log V_X(\tilde{Y};\hat{\theta})$, où $\hat{\theta}$ est l'estimateur calculé sur l'échantillon d'apprentissage Y. C'est une façon de mesurer la qualité de l'estimation sur un échantillon test. On retiendra finalement le modèle (c'est à dire la matrice X) pour lequel $\mathbb{E}(-\log V_X(\tilde{Y};\hat{\theta}))$ est minimal. Il s'agira en pratique d'estimer $\mathbb{E}(-\log V_X(\tilde{Y};\hat{\theta}))$, ce qui est l'objectif du critère AIC.

Cette approche est dans le même esprit que la minimisation de R_{in} ayant conduit au C_p de Mallows, mais ici c'est $\mathbb{E}(-\log V_X(\tilde{Y};\hat{\theta}))$ qui joue le rôle de la fonction de perte et non R_{in}. En fait lorsque σ^2 est supposé connu (hypothèse peu réaliste), le critère AIC et le C_p de Mallows coincident.

Proposition 3.2.1. *Dans le modèle de régression Gaussien, si σ^2 est connu, alors le critère AIC est équivalent au C_p de Mallows.*

Proof. Puisque σ^2 est connu, le seul paramètre inconnu est β. Dans le cas Gaussien $-\log V_X(\tilde{Y};\hat{\beta}) = n\log(\sigma\sqrt{2\pi}) + \|\tilde{Y} - X\hat{\beta}\|^2/(2\sigma^2)$ de telle sorte que le modèle qui minimise $\mathbb{E}(-\log V_X(\tilde{Y};\hat{\theta}))$ minimise de façon équivalente R_{in}. Le critère AIC et le critère C_p coincident : il suffit de minimiser un estimateur sans biais de R_{in}, c'est à dire $SCR/n + 2\min(n,p)\sigma^2/n$, où ici σ^2 n'a pas à être estimé puisqu'il est supposé connu. □

Dans le cas général (σ^2 inconnu), H. Akaike a montré dans un article de 1973 que sous certaines conditions, lorsque n est grand,

$$\mathbb{E}(-\log V_X(\tilde{Y};\hat{\theta})) \approx -\frac{2}{n}\mathbb{E}(-\log V_X(Y;\hat{\theta})) + 2\,\frac{p+1}{n}, \tag{3.2}$$

où $p+1$ correspond à la dimension du vecteur $\theta = (\beta, \sigma^2)$. Le critère AIC consiste donc à minimiser $-2\log V_X(Y;\hat{\theta}) + 2(p+1)$ qui est (à un facteur n près qui ne change rien à la minimisation) un estimateur asymptotiquement sans biais du risque. Dans le cas Gaussien,

$$\begin{aligned} -\log V_X(Y;\hat{\theta}) &= \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\hat{\sigma}^2) + \frac{\|Y - X\hat{\beta}\|^2}{2\hat{\sigma}^2} \\ &= \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(SCR/n) + \frac{SCR}{2SCR/n} \end{aligned}$$

de telle sorte que le critère AIC revient à minimiser

$$AIC = n\log\frac{SCR}{n} + 2(p+1).$$

Comme cela a été remarqué dans le chapitre précédent, lorsque $p \geq n$ $SCR = 0$ donc $AIC = -\infty$. Le critère AIC n'est donc pas adapté à la grande dimension.

3.2.3 Critère BIC

Le critère BIC (Bayesian Information Criterion) est motivé différemment que par la minimisation de l'erreur test. Comme son nom l'indique, il s'agit d'une approche bayésienne. Dans ce contexte on associe à chaque modèle une probabilité a priori et on choisit le modèle ayant la probabilité a posteriori (sachant les observations) maximale.

Formellement, on note

- $\mathcal{M}_X$ le modèle linéaire associée à la matrice X
- $\mathbb{P}(\mathcal{M}_X)$: la probabilité a priori que ce modèle soit valide, c'est à dire que X contiennent les bonnes variables explicatives, sans considération de la valeur du paramètre associé. On supposera par la suite que cette probabilité est identique pour chaque modèle, de telle sorte qu'aucun modèle n'est privilégié a priori.
- $\mathbb{P}(\theta|\mathcal{M}_X)$: la loi a priori du paramètre θ dans le modèle $\mathcal{M}_X$.
- $\mathbb{P}(Y|\mathcal{M}_X, \theta)$: la loi de Y sachant que le modèle est $\mathcal{M}_X$ associé au paramètre θ. Vu comme une fonction de θ, il s'agit simplement de la vraisemblance, notée $V_X(Y;\theta)$ dans la partie précédente.

La probabilité a posteriori de $\mathcal{M}_X$ sachant les observations Y vaut d'après la formule de Bayes

$$\mathbb{P}(\mathcal{M}_X|Y) = \frac{\mathbb{P}(Y|\mathcal{M}_X)\mathbb{P}(\mathcal{M}_X)}{\mathbb{P}(Y)},$$

où $\mathbb{P}(Y)$ est la loi de Y intégrée sur tous les modèles possibles. Puisqu'on a supposé $\mathbb{P}(\mathcal{M}_X)$ constant, maximiser $\mathbb{P}(\mathcal{M}_X|Y)$ en $\mathcal{M}_X$ revient à maximiser $\mathbb{P}(Y|\mathcal{M}_X)$. Cette quantité vaut d'après la formule des probabilités totales

$$\mathbb{P}(Y|\mathcal{M}_X) = \int \mathbb{P}(Y|\mathcal{M}_X, \theta)\mathbb{P}(\theta|\mathcal{M}_X)d\theta.$$

Lorsque n est grand, et sous certaines conditions, G. Schwarz (1978) a montré que l'on peut approcher le logarithme de l'intégrale précédente de la façon suivante

$$\log \mathbb{P}(Y|\mathcal{M}_X) \approx \log \mathbb{P}(Y|\mathcal{M}_X, \hat{\theta}) - \frac{p+1}{2}\log n,$$

où $\hat{\theta}$ est l'estimateur du maximum de vraisemblance de θ dans le modèle $\mathcal{M}_X$ et $(p+1)$ est sa dimension. Un résultat remarquable est que cette approximation ne dépend pas du choix de la loi a priori $\mathbb{P}(\theta|\mathcal{M}_X)$ sur les paramètres.

Le critère BIC est défini comme valant -2 fois la quantité précédente, qu'il s'agit donc de minimiser parmi les modèles testés. On remarque que la forme est similaire au critère AIC puisque c'est la somme de la log-vraisemblance calculée en $\hat{\theta}$ plus un terme de pénalité, qui fait intervenir ici $\log n$ au lieu de 2 dans AIC.

Si l'on suppose enfin que le modèle est Gaussien, la log-vraisemblance dans cette formule se simplifie comme pour le AIC et on obtient

$$BIC = n \log \frac{SCR}{n} + (p+1) \log n.$$

Pour les mêmes raisons que pour AIC, le critère BIC n'est pas adapté à la grande dimension.

3.2.4 Comparaison des critères

Tous les critères précédents consistent à minimiser une expression de la forme

$$f(SCR) + c(n)p$$

où f est une fonction croissante de SCR et $c(n)p$ est un terme pénalisant les modèles ayant beaucoup de variables. La fonction $c(n)$ est appelée la pénalité. Elle vaut 2 pour C_p et AIC et elle vaut $\log(n)$ pour BIC. D'autres pénalités ont été proposées, privilégiant de façon plus ou moins forte les modèles les plus parcimonieux (p faible).

On observe en particulier que dès que $\log(n) > 2$, le critère BIC pénalise davantage la complexité du modèle que AIC. Le C_p de Mallows est quant à lui très similaire à AIC, comme l'atteste la proposition 3.2.1.

Lors de la sélection de variables dans un modèle de régression linéaire, les critères usuelles s'ordonnent de la manière suivante en fonction de leur propension à sélectionner le modèle le plus parcimonieux :

$$BIC < F\,test < C_p \approx AIC < R_a^2 < R^2$$

3.3 Validation croisée

Le principe de la validation croisée est d'estimer le risque test R_{test} en confrontant le modèle à un échantillon test qui n'a pas été utilisé pour l'ajustement du modèle. Il y a différentes manières de construire un échantillon test, décrites ci-après.

Les avantages de la validation croisée par rapport à l'utilisation des critères de sélection est double :

- La validation croisée estime $R_{\text{test}}(\tilde{X})$ pour de nouvelles valeurs $\tilde{X}$ et pas seulement $R_{\text{in}} = R_{\text{test}}(X)$. (L'estimation est néanmoins limitée aux valeurs $\tilde{X}$ présentes dans l'échantillon test et il faut donc veiller à ne pas trop généraliser son interprétation).

- La procédure est plus générale puisque l'estimation de R_{test} ne s'appuie pas sur l'hypothèse que les données suivent un modèle particulier (contrairement par exemple à AIC ou BIC qui s'appuient pour leur construction sur la vraisemblance du modèle).

Le principal inconvénient de la validation croisée est son temps de calcul, qui peut parfois être prohibitif.

L'objectif est d'estimer R_{test}. Comme toute estimation, les procédures suivantes souffrent d'un biais et d'une variance, que l'on souhaite les plus petits possibles. Un biais trop élevé signifie que l'on sous-estime ou sur-estime le risque associé. Par ailleurs, une procédure d'estimation trop variable rend peu fiable la comparaison des risques entre deux modèles : le fait qu'un risque estimé est inférieur à un autre peut être dû aux fluctuations des estimateurs. Pour choisir la bonne méthode de validation croisée, on veillera à ce que le biais et la variance soient les moins élevés possibles, tout en respectant un temps de calcul raisonnable.

3.3.1 Hold out

Le principe est simple : on sépare l'échantillon en deux parties, une pour estimer le modèle, l'autre pour estimer l'erreur de prévision. Concernant la taille de chaque échantillon, on rencontre deux choix courants : soit la même taille $n/2$ pour les deux échantillons, soit $3n/4$ pour l'échantillon d'apprentissage et $n/4$ pour l'échantillon test.

Avantages :

- Très simple à mettre en oeuvre
- Un seul ajustement à effectuer par modèle.

Inconvénients :

- La taille de l'échantillon d'apprentissage ($n/2$ ou $3n/4$) rend l'estimation moins précise que si on utilisait les n observations. Cela conduit à une sur-estimation de l'erreur test par rapport à celle issue d'une estimation sur n individus.
- L'évaluation sur un seul échantillon test rend l'estimation de l'erreur très variable (la variance est élevée).

Pour minimiser ce dernier inconvénient, on peut recommencer la procédure Hold-out plusieurs fois (une dizaine de fois), en considérant plusieurs séparations aléatoires de l'échantillon, toujours de même taille. Le risque final, obtenu par la moyenne des risques issus de chaque découpage, est alors moins variable. Cette généralisation fait néanmoins perdre les avantages initiaux, en particulier le coût est supérieur puisqu'il faut estimer et évaluer chaque modèle autant de fois qu'il y a de découpages différents.

L'idée d'itérer les découpages est en fait à la base des méthodes suivantes.

3.3.2 Leave-one-out (LOO)

La validation croisée Leave-one-out consiste à mettre de côté une observation, d'estimer le modèle sur les $n-1$ autres observations, puis de calculer le risque de prévision sur l'observation mise de côté. Ce processus est répété pour chaque observation de l'échantillon, de telle sorte que n erreurs de prévision sont finalement calculées. Leur moyenne est une estimation de l'erreur test.

De façon plus formelle, soit $\hat{y}_i^{(-i)}$ la prévision de y_i à partir du modèle estimé sur toutes les observations sauf i. Le risque estimé par LOO est

$$CV_{LOO} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{(-i)})^2.$$

Avantages :

- Chaque échantillon d'apprentissage a une taille $n-1$, très proche de la vraie taille n de l'échantillon total. Ainsi le risque estimé sera peu biaisé.

Inconvénients :

- Peut-être très couteux à mettre en oeuvre car il nécessite en toute généralité n ajustements de chaque modèle testé. Il y a une exception notable : dans le cas de la régression linéaire, CV_{LOO} s'exprime directement grâce à la seule estimation du modèle sur les n observations, cf la proposition 3.3.1.
- Tous les échantillons d'apprentissage sont très semblables (ils différent que par une observation) et donc le risque final par LOO ne moyenne pas suffisamment de situations différentes. En conséquences l'estimation du risque a une forte variance.

Proposition 3.3.1. *L'estimation de l'erreur test par LOO d'un modèle de régression linéaire $Y = X\beta + \epsilon$ est simplement*

$$CV_{LOO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2,$$

où $\hat{y}_i$ est la prévision de y_i à partir du modèle estimé sur toutes les observations et h_i est l'effet levier, défini par le $i$ème élément de la diagonale de $P_{[X]}$, c'est à dire $h_i = x_i'(X'X)^{-1}x_i$.

Proof. On note $Y_{(-i)}$, respectivement $\epsilon_{(-i)}$, le vecteur Y, resp. ϵ, privé de son i-ème élément y_i, resp. ϵ_i, et $X_{(-i)}$ la matrice X privée de sa i-ème ligne x_i'. Le modèle sans l'observation i s'écrit donc $Y_{(-i)} = X_{(-i)}\beta + \epsilon_{(-i)}$. On note enfin $\hat{\beta}_{(-i)}$ l'estimation de β par les MCO dans ce modèle.

Il s'agit de montrer que $y_i - \hat{y}_i^{(-i)} = \hat{\epsilon}_i/(1-h_i)$. Ceci est une conséquence du lemme suivant.

Lemme 3.3.2.

$$\hat{\beta}_{(-i)} = \hat{\beta} - \frac{1}{1-h_i}(X'X)^{-1}x_i\hat{\epsilon}_i.$$

En effet, on a alors

$$y_i - \hat{y}_i^{(-i)} = y_i - x_i'\hat{\beta}_{(-i)} = y_i - x_i'\hat{\beta} + \frac{x_i'(X'X)^{-1}x_i}{1-h_i}\hat{\epsilon}_i = \hat{\epsilon}_i + \frac{h_i}{1-h_i}\hat{\epsilon}_i = \frac{\hat{\epsilon}_i}{1-h_i}.$$

□

Preuve du lemme. On sait que

$$\hat{\beta}_{(-i)} = (X'_{(-i)}X_{(-i)})^{-1}X'_{(-i)}Y_{(-i)}.$$

On peut facilement vérifier que $X'_{(-i)}X_{(-i)} = X'X - x_i x'_i$ et que $X'_{(-i)}Y_{(-i)} = X'Y - x_i y_i$. Par ailleurs la formule de Sherman-Morrison donne l'inverse de $M + uv'$ pour toute matrice inversible M de taille p et tous vecteurs u et v dans $\mathbb{R}^p$:

$$(M + uv')^{-1} = M^{-1} - \frac{M^{-1}uv'M^{-1}}{1 + u'M^{-1}v}.$$

On en déduit que

$$(X'_{(-i)}X_{(-i)})^{-1} = (X'X)^{-1} + \frac{1}{1-h_i}(X'X)^{-1}x_i x'_i (X'X)^{-1}.$$

Ainsi

$$\begin{aligned}
\hat{\beta}_{(-i)} &= \left((X'X)^{-1} + \frac{1}{1-h_i}(X'X)^{-1}x_i x'_i (X'X)^{-1}\right)(X'Y - x_i y_i) \\
&= (X'X)^{-1}X'Y - (X'X)^{-1}x_i y_i + \frac{(X'X)^{-1}x_i x'_i \hat{\beta}}{1-h_i} - \frac{(X'X)^{-1}x_i h_i y_i}{1-h_i} \\
&= \hat{\beta} - \frac{(X'X)^{-1}x_i}{1-h_i}((1-h_i)y_i - x'_i\hat{\beta} + h_i y_i) \\
&= \hat{\beta} - \frac{1}{1-h_i}(X'X)^{-1}x_i\hat{\epsilon}_i.
\end{aligned}$$

□

3.3.3 Leave-k-out

L'idée est la même que le Leave-one-out, sauf qu'au lieu de mettre de côté une seule observation, on en met k et on calcule le risque de prévision sur les k observations mises de côté. On réitère le processus pour tous les découpages possibles, il y en a $\binom{n}{k} = n!/(k!(n-k)!)$, et on moyenne tous les risques pour obtenir l'estimation finale.

Si $k = 1$, on retrouve évidemment le LOO. Pour k plus grand, l'estimation du risque est un peu plus biaisé qu'avec le LOO (car on ajuste le modèle sur moins d'observations), mais la variance est plus faible car les échantillons

d'apprentissage sont plus diverses. Ainsi il existe un choix de k garantissant un bon compromis biais-variance (ce choix dépend de n et le consensus est de le prendre de l'ordre de $n/10$ ou $n/5$). Cependant, le défaut majeur de cette démarche est son coût : elle requiert d'estimer le modèle sur un nombre prohibitif d'échantillons d'apprentissage. A titre d'exemple, pour $n = 100$ et $k = 5$, ce qui est loin de constituer un cas de grande dimension, cela représente plus de 75 millions d'échantillons.

Avantage :

- Bonne estimation du risque test pour un choix de k approprié.

Inconvénient :

- Généralement impossible à mettre en oeuvre en pratique.

3.3.4 K-fold

On coupe l'échantillon en K parties égales (ou à peu près si n n'est pas multiple de K) de taille respective n/K. On apprend le modèle en utilisant $K-1$ parties de l'échantillon et on le teste sur la partie restante pour fournir une estimation du risque. En répétant cette opération pour tous les cas possibles, cela représente K risques à calculer, que l'on moyenne pour fournir l'estimation du risque final.

Dans cette procédure, les K échantillons d'apprentissage ont donc une taille de $n - n/K$ et le risque final est la moyenne des K risques tests, chacun calculé sur un échantillon test de n/K individus.

Lorsque $K = n$, il s'agit de la procédure LOO.

Lorsque $K = 2$, cela ressemble à une procédure Hold out dans laquelle l'échantillon d'apprentissage et l'échantillon test ont même taille. Ce n'est cependant pas tout à fait équivalent car ici on inverse le rôle joué par chaque partie pour fournir deux estimations du risque que l'on moyenne, contrairement à Hold out où une seule estimation du risque est effectuée.

Si K est choisi trop grand, il y aura un faible biais car les échantillons d'apprentissage auront une taille très proche de n, mais par contre l'estimation du risque souffrira d'une grande variance comme pour le LOO. Le désavantage majeure dans ce cas est similaire au LOO : le calcul peut-être très long car il faut estimer K modèles. A l'inverse, si K est petit, le calcul est rapide, la variance d'estimation est moindre car on moyenne des situations où les échantillons sont très différents, mais l'estimation risque d'être

biaisée car la taille de l'échantillon d'apprentissage est beaucoup plus petite que la vraie taille n.

En pratique, les valeurs $K = 5$ à $K = 10$ sont recommandées.

Avantage :

- Bonne estimation du risque test pour un choix de K approprié ($K = 5$ à $K = 10$)
- Relativement rapide à mettre en oeuvre pour les choix de K précédents.

Inconvénient :

- L'estimation du risque n'est pas aussi bonne qu'avec une procédure leave-k-out optimale.

3.4 Sélection automatique d'un sous-modèle

L'idée est de parcourir un ensemble de sous-modèles, d'estimer pour chacun d'entre eux le risque de prévision associé à l'aide d'une des méthodes discutées précédemment, puis de retenir le sous-modèle dont l'erreur estimée est minimale. Il existe quatre façons classiques de parcourir un ensemble de sous-modèles :

- La recherche exhaustive. On parcourt tous les 2^p sous-modèles possibles. Cette approche est impossible en grande dimension. Par exemple, pour $p = 20$ (ce qui reste raisonnable), cela représente plus d'un million de modèles à tester.
- La sélection stepwise backward. On part du plus gros modèle possible à p variables et on élimine la variable la moins importante, dans le sens où le modèle sans cette dernière a le risque le plus faible parmi les modèles à $p - 1$ variables. On continue jusqu'à ce que l'élimination d'une variable n'améliore plus le risque. Il y a au maximum $1 + p(p + 1)/2$ modèles testés. Par exemple, pour $p = 20$, cela représente 211 modèles, ce qui est beaucoup plus raisonnable que la recherche exhaustive. Par contre cette démarche ne garantit pas de trouver le sous-modèle optimal. En effet une variable éliminée en début d'algorithme n'a plus aucune chance d'être réintégrée par la suite, alors qu'il est possible qu'elle appartienne en réalité au sous-modèle optimal. Cet algorithme

est en général inutilisable en grande dimension car les plus gros modèles sont trop coûteux à estimer, et les critères de sélection lorsque $p > n$ deviennent dégénérés ($R^2 = 1$, $AIC = -\infty$, etc).

- La sélection stepwise forward. On part du plus petit modèle ne contenant que la constante et on ajoute la variable la plus importante. On continue jusqu'à ce que l'ajout d'une variable n'améliore plus le modèle. Comme pour la sélection backward, il y a au maximum $1 + p(p+1)/2$ modèles testés, sans garantie de trouver le sous-modèle optimal (une variable introduite en début d'algorithme y reste jusqu'à la fin). Contrairement à la sélection backward, la sélection forward est généralement possible en grande dimension car on part des plus petits modèles dont l'estimation est faisable.

- La sélection stepwise hybride (backward ou forward). Pour commencer on part du plus petit modèle (pour la hybride forward) et on ajoute la variable la plus importante. Dans les étapes suivantes, on estime l'erreur associée à l'ajout de chaque variable (comme en forward classique) mais on estime également l'erreur associée au retrait de chaque variable déjà présente dans le modèle. Ainsi le modèle suivant retenu peut contenir une variable de plus ou une variable de moins. On continue ainsi jusqu'à ce que l'ajout ou le retrait d'une variable n'améliore plus la modélisation. La hybride backward est similaire sauf que l'on part du plus gros modèle. Le coût est un peu plus élevé que les sélections backward et forward car à chaque étape les p variables sont testées. Mais cette méthode parcourt davantage de modèles.

En présence d'un grand nombre de variables, la méthode hybride forward est la plus recommandée, car elle part des plus petits modèles dont l'estimation est faisable et elle a un coût raisonnable tout en parcourant une grande variété de sous-modèles possibles.

Mise en oeuvre sous R (exemple avec `lm` et les critères AIC et BIC) :

En supposant que le tableau de données se nomme `tab`, on peut utiliser la fonction `regsubsets` de la librairie `leaps` pour effectuer une sélection exhaustive ou backward ou forward dans un modèle de régression linéaire, de la façon suivante.

```
res=regsubsets(y~.,data=tab)
res=regsubsets(y~.,data=tab,method="backward")
```

```
res=regsubsets(y~.,data=tab,method="forward")
```

La valeur des SCR, BIC, C_p pour chaque modèle testé est disponible dans `summary(res)`, voir aussi `plot(res,scale="Cp")` (par exemple).

De façon alternative, on peut utiliser la fonction `step` qui s'applique à des modèles plus généraux (issus de la fonction `glm` notamment). Pour cela on commence par estimer le plus gros modèle et le plus petit modèle.

```
fit_full=lm(y~.,data=tab)
fit0=lm(y~1,data=tab)
```

Pour la sélection backward, en supposant qu'il y a n observations :

```
step(fit_full,direction="backward") #critère AIC
step(fit_full,direction="backward",k=log(n)) #critère BIC
```

Pour la sélection backward hybride (par défaut sous R) :

```
step(fit_full) #critère AIC
step(fit_full,k=log(n)) #critère BIC
```

Pour la sélection forward :

```
step(fit0,scope=formula(fit_full),direction="forward") #critère AIC
step(fit0,scope=formula(fit_full),direction="forward",k=log(n)) #critère BIC
```

Pour la sélection forward hybride :

```
step(fit0,scope=formula(fit_full)) #critère AIC
step(fit0,scope=formula(fit_full),k=log(n)) #critère BIC
```

3.5 Estimation du modèle retenu

Une fois le modèle sélectionné par l'une des approches précédentes, il convient de l'estimer sur les n individus de l'échantillon total avant de l'utiliser pour des prévisions.

Cette précision concerne surtout un modèle retenu par validation croisée : son estimation s'est faite sur un sous-échantillon d'apprentissage afin de le confronter à un sous-échantillon test. Maintenant que ce modèle est retenu, il faut l'estimer sur tout l'échantillon à disposition, ce qui ne peut qu'améliorer la qualité d'estimation et donc les prévisions associées.

Chapter 4

Réduction de dimension et régression sous contraintes

En présence de nombreuses variables explicatives, on suppose généralement que peu d'entre elles sont pertinentes pour modéliser Y. On peut relever trois familles de méthodes qui permettent d'exploiter cette parcimonie:

- Les méthodes de sélection, abordées dans le chapitre précédent. L'idée est de choisir le sous-modèle dont l'estimation du risque de prévision est minimale.

- La réduction de dimension. L'idée est de projeter les p variables, évoluant en toute généralité dans un espace vectoriel de dimension p, dans un sous-espace vectoriel de dimension beaucoup plus petit. On peut alors régresser Y sur ce sous-espace afin d'éviter les écueils de la grande dimension. C'est l'objet de la section 4.1 qui présente les méthodes PCR et PLS.

- Les estimations contraintes. L'idée est d'utiliser une méthode d'estimation qui contraint les paramètres à ne pas exploser (contrairement aux MCO en grande dimension). Ainsi l'estimation est moins variable et les prévisions plus fiables. Parmi ces méthodes, les techniques de type "Lasso" conduisent à estimer certains coefficients par 0, auquel cas une sélection de variables s'opère dans le même temps. Ces méthodes sont abordées dans les sections 4.2-4.4.

4.1 Réduction de dimension : PCR et PLS

4.1.1 Régression sur composantes principales (PCR)

L'analyse en composantes principales (ACP ou PCA en anglais) consiste à trouver une base orthogonale de l'espace vectoriel [X] dont les vecteurs $Z_1, \dots, Z_p$, appelés composantes principales ou axes principaux, sont construits de telle sorte à garder le plus "d'information" possible contenue dans les vecteurs initiaux $X_1, \dots, X_p$. L'information est quantifiée à l'aide de la variance des coordonnées des n individus sur chaque nouvel axe : plus la variance est élevée et plus le nuage de points se répartit bien sur l'axe, gardant ainsi le maximum de la diversité initiale du nuage, contrairement au cas extrême inverse où tous les points seraient projetés au même endroit (conduisant à une variance nulle).

Un exemple de construction des 2 axes principaux d'un nuage de points en 2D est donné dans la figure 4.1. L'intérêt de considérer des projections sur les premiers axes principaux pour résumer un nuage de points en illustré dans la figure 4.2. Les données sont l'image 3D d'un chameau. Le premier axe principal est celui qui traverse le chameau sur sa longueur, le second sur sa hauteur et le troisième sur sa largeur. La projection sur le plan formé par les deux premiers axes (à droite de la figure) suffit à bien représenter les données. Cette projection est plus informative que celle sur les deux derniers axes (à gauche de la figure), à partir de laquelle il est plus difficile de deviner la nature des données initiales. Travailler à partir du plan factoriel de droite permet donc une réduction de dimension (de 3 à 2 dans cet exemple) sans grande perte d'information.

Cette approche est sensible à l'échelle : si une variable X_j est d'un ordre de grandeur 1000 fois supérieur aux autres variables, l'essentiel de la variance du nuage de points reposera sur X_j et le premier axe sera donc très proche de X_j. Pour éviter ce phénomène, les variables initiales sont centrées et réduites. On suppose donc dans la suite que chaque variable X_j est centrée et de variance 1.

Concrètement, étant donnée une matrice X de variables centrées réduites, le premier axe Z_1 est choisi comme étant la combinaison linéaire de $X_1, \dots, X_p$ de variance maximale : $Z_1 = X\alpha_1$ avec $\|\alpha_1\| = 1$ et $\mathrm{Var}(X\alpha_1)$ maximale parmi tous les vecteurs de la forme $X\alpha$. Dans ce formalisme, $\alpha_1 \in \mathbb{R}^p$ représente la direction du premier axe principal et $X\alpha_1 \in \mathbb{R}^n$ est l'ensemble des coordonnées du nuage de points sur cet axe. Le second axe

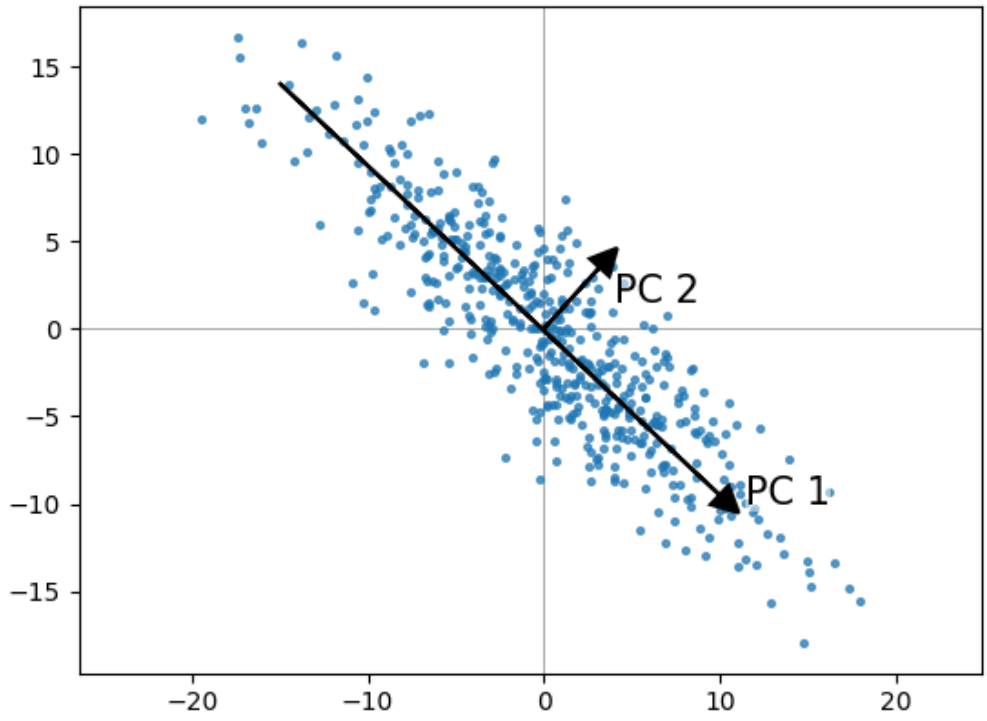

Figure 4.1: Construction des deux axes principaux d'un nuage de points en 2D.

Figure 4.2: Chameau (en 3D) observé via sa projection sur le plan formé des axes principaux 2 et 3 de l'ACP (à gauche) et sur le plan formé des deux premiers axes de l'ACP (à droite). La projection 2D de droite est plus informative et suffit à comprendre l'objet 3D.

est choisi de la même manière, avec la contrainte supplémentaire d'être orthogonal à Z_1. Cette dernière contrainte s'écrit à l'aide du produit scalaire $(X\alpha_2)'X\alpha_1 = 0$, soit $\alpha_2' X'X\alpha_1 = 0$. De façon générale, pour $j = 1, \dots, p$, le j-ème axe est $Z_j = X\alpha_j$ où

$$\alpha_j = \underset{\alpha \in \mathbb{R}^p}{argmax}\ \mathrm{Var}(X\alpha) \tag{4.1}$$

sous les contraintes $\|\alpha\| = 1$ et $\alpha' X'X\alpha_l = 0$ pour tout $l = 1, \dots, j-1$. Dans (4.1), puisque les variables sont supposées centrées, on a $\mathrm{Var}(X\alpha) = \frac{1}{n}\alpha' X'X\alpha$.

Par construction tous les axes sont orthogonaux et ils sont ordonnés du plus "informatif" Z_1, au moins informatif Z_p, au sens où la variance des coordonnées des individus sur les axes décroit. L'algorithme pour résoudre le problème d'optimisation précédent est simple : les directions α_j correspondent aux vecteurs propres (normalisés) de la matrice $X'X$, ordonnés par ordre décroissant de leur valeur propre associée.

Le principe de la régression sur composantes principales (PCR) est le suivant : plutôt que de régresser Y sur $X_1, \dots, X_p$, on régresse Y sur les

premiers axes principaux $Z_1, \dots, Z_M$ (en incluant de plus une constante), où le nombre de composantes M retenues est à choisir. Cette approche permet de réduire la dimension du problème de p à M, tout en garantissant l'orthogonalité des variables explicatives retenues. Le modèle s'écrit ainsi

$$Y = \gamma_0 + \sum_{j=1}^{M} \gamma_j Z_j + \epsilon$$

et l'on peut montrer que les estimateurs par moindres carrés associés sont

$$\hat{\gamma}_0 = \bar{Y}, \qquad \hat{\gamma}_j = \frac{Y' Z_j}{Z_j' Z_j}. \tag{4.2}$$

L'interprétation des M composantes principales retenues et des coefficients estimés $\hat{\gamma}_j$ de la PCR n'est pas toujours aisée. Néanmoins, il est possible de revenir aux variables initiales puisque $Z_j = X\alpha_j$. On obtient

$$\hat{Y} = \bar{Y} + X\hat{\beta} \tag{4.3}$$

avec $\hat{\beta} = \sum_{j=1}^{M} \hat{\gamma}_j \alpha_j$.

La mise en oeuvre d'une PCR requiert le choix du nombre M de composantes retenues. Ceci se fait par validation croisée (voir les différentes versions dans le chapitre précédent) : on estime l'erreur de prévision de la PCR lorsque M varie de 1 à p (ou une valeur maximale inférieure à p), et on retient la valeur de M associée à l'erreur minimale.

Mise en oeuvre sous R : fonction `pcr` de la libraire `pls`. La sélection par validation croisée est proposée en option de la fonction.

4.1.2 Régression des moindres carrés partiels (PLS)

Les composantes d'une ACP sont construites pour représenter au mieux l'information contenue dans les variables $X_1, \dots, X_p$. Cette construction ne tient pas compte du lien avec Y, qui est pourtant l'objectif premier d'un modèle de régression. La régression PLS remédie à ce défaut en construisant les composantes de telle sorte qu'elles représentent au mieux les variables $X_1, \dots, X_p$ et qu'elles soient dans le même temps le plus possible corrélées avec Y. Ainsi, contrairement à (4.1), les composantes d'une PLS sont

$Z_1, \dots, Z_p$ où pour tout j, $Z_j = X\alpha_j$ et

$$\alpha_j = \underset{\alpha \in \mathbb{R}^p}{argmax}\ \mathrm{Cov}(X\alpha, Y) \tag{4.4}$$

sous les contraintes $\|\alpha\| = 1$ et $\alpha X'X\alpha_l = 0$ pour tout $l = 1, \dots, j-1$. Dans cette expression $\mathrm{Cov}(X\alpha, Y) = \alpha' X'Y/n$ car les variables sont supposées centrées. Puisque $\mathrm{Cov}(X\alpha, Y)^2 = \mathrm{Corr}(X\alpha, Y)^2 \mathrm{Var}(X\alpha)\mathrm{Var}(Y)$, où Corr désigne la corrélation, (4.4) peut aussi se récrire

$$\alpha_j = \underset{\alpha \in \mathbb{R}^p}{argmax}\ \mathrm{Corr}(X\alpha, Y)^2 \mathrm{Var}(X\alpha),$$

ce qui montre bien que chaque composante est choisie comme le compromis qui d'une part maximise la variance de la projection du nuage de points formé par les variables explicatives $X_1, \dots, X_p$ (comme pour l'ACP) et d'autre part maximise la corrélation entre la composante et Y.

L'algorithme pour construire les axes de la PLS est le suivant : partant de la matrice $X_{(1)} = X$ centrée réduite et de $j = 1$, tant que $j \leq p$ et $\alpha_j \neq 0$

1. $\alpha_j = X'_{(j)}Y / \|X'_{(j)}Y\|$.
2. $Z_j = X_{(j)}\alpha_j$
3. $X_{(j+1)} \leftarrow X_{(j)} - Z_j Z'_j X_{(j)}/(Z'_j Z_j)$ et retour à 1 avec $j \leftarrow j+1$.

Si $\alpha_j = 0$ pour un certain j, $Z_k = 0$ pour tout $k \geq j$. La première étape est la solution de (4.4) sachant que la contrainte d'orthogonalité ($\alpha X'X\alpha_l = 0$ pour tout $l = 1, \dots, j-1$) est nécessairement vérifiée grâce à l'étape 3. En effet cette opération effectue la projection $P_{[Z_j]^\perp} X_{(j)}$, autrement dit elle construit de nouvelles variables (regroupées dans la nouvelle matrice $X_{(j+1)}$) orthogonales à l'axe Z_j. Ainsi tout vecteur $X_{(j+1)}\alpha$ considéré dans l'itération suivante sera nécessairement orthogonal aux axes déjà construits.

Une fois les axes obtenus, la régression PLS est similaire à la PCR : on effectue la régression sur les M premiers axes, en choisissant M par validation croisée. De part l'orthogonalité des axes construits, les coefficients de régression s'obtiennent de la même manière que pour la PCR, voir (4.2). De même, puisque chaque axe est combinaison linéaire des variables $X_1, \dots, X_p$, on peut exprimer $\hat{Y}$ en fonction de X, comme dans (4.3) pour la PCR.

Mise en oeuvre sous R : fonction `plsr` de la libraire `pls`. La sélection par validation croisée est proposée en option de la fonction.

4.2 Régression ridge

En présence d'un grand nombre de variables, la matrice $X'X$ n'est plus forcément de rang p et n'est donc plus inversible, rendant l'estimateur des moindres carrés incalculable. De même, en présence de variables fortement corrélées entre elles (problème de multicolinéarité), la matrice peut être inversible mais son inverse, et par suite l'estimateur des MCO, sont très instables, dans le sens où une légère modification des données (par exemple la suppression d'un individu, c'est à dire d'une ligne de X) peut conduire à une matrice inverse radicalement différente, et de même pour la valeur de l'estimateur. Ce phénomène est clairement reflété dans la variance de l'estimateur par MCO qui vaut $\sigma^2(X'X)^{-1}$.

Lorsque $X'X$ est non inversible ou d'inverse instable, cela se lit dans ses valeurs propres : certaines sont nulles ou quasiment nulles. Une manière de régulariser l'inversion consiste à ajouter une petite valeur $\lambda > 0$ aux valeurs propres. On remplace donc $X'X$ par $X'X + \lambda I_p$ où I_p est la matrice identité de taille p. Il s'agit de la régularisation de Tikhonov, connue en statistique sous le nom de régression ridge.

L'estimateur ridge de β dans un modèle de régression est ainsi

$$\hat{\beta}_{\text{ridge}} = (X'X + \lambda I_p)^{-1} X'Y.$$

La proposition suivante montre que l'introduction du paramètre λ biaise l'estimation, mais en contre-partie diminue sa variance (c'est l'effet de la stabilisation), de telle sorte que pour des valeurs petites de λ, l'estimateur ridge est préférable à l'estimateur par MCO au sens du coût quadratique.

Proposition 4.2.1. *Si $Y = X\beta + \epsilon$ avec $\mathbb{E}(\epsilon) = 0$ et $\text{Var}(\epsilon) = \sigma^2 I_n$:*

- *l'estimateur ridge est biaisé, précisément $\mathbb{E}(\hat{\beta}_{\text{ridge}}) = (X'X+\lambda I_p)^{-1}X'X\beta$*
- *sa variance vaut $\text{Var}(\hat{\beta}_{\text{ridge}}) = \sigma^2(X'X + \lambda I_p)^{-1}X'X(X'X + \lambda I_p)^{-1}$*
- *sa matrice EQM est inférieure à celle de l'estimateur par MCO (dans le sens où la différence est définie positive) dès que $0 < \lambda < 2\sigma^2/\|\beta\|^2$.*

Proof. Le calcul de l'espérance et de la variance est immédiat. On en déduit après quelques calculs

$$\begin{aligned} EQM(\hat{\beta}_{\text{ridge}}) &= (\mathbb{E}(\hat{\beta}_{\text{ridge}}) - \beta)(\mathbb{E}(\hat{\beta}_{\text{ridge}}) - \beta)' + \text{Var}(\hat{\beta}_{\text{ridge}}) \\ &= (X'X + \lambda I_p)^{-1}(\sigma^2 X'X + \lambda^2 \beta\beta')(X'X + \lambda I_p)^{-1}. \end{aligned}$$

Si $X'X$ n'est pas inversible, l'estimateur par MCO n'est pas défini de façon unique et sa variance est infinie. L'EQM de $\hat{\beta}_{\text{ridge}}$ est donc toujours inférieure dans ce cas. On suppose dans la suite que $X'X$ est inversible. En notant $\hat{\beta}$ l'estimateur des MCO de β, on a

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

qui peut se récrire

$$EQM(\hat{\beta}) = \sigma^2(X'X + \lambda I_p)^{-1}(X'X + 2\lambda I_p + \lambda^2(X'X)^{-1})(X'X + \lambda I_p)^{-1}.$$

En notant $A = (X'X + \lambda I_p)^{-1}$, qui est symétrique, on en déduit que

$$EQM(\hat{\beta}) - EQM(\hat{\beta}_{\text{ridge}}) = A(2\lambda\sigma^2 I_p + \lambda^2\sigma^2(X'X)^{-1} - \lambda^2\beta\beta')A'.$$

Cette matrice est définie positive si et seulement si $(2\lambda\sigma^2 I_p + \lambda^2\sigma^2(X'X)^{-1} - \lambda^2\beta\beta')$ l'est. Ceci peut se vérifier en utilisant le fait que A est inversible et qu'une matrice M est définie positive ssi il existe N inversible telle que $M = N'N$ (décomposition de Choleski). De plus la somme de deux matrices définies positives est définie positive. Puisque $\lambda^2\sigma^2(X'X)^{-1}$ est définie positive, une condition suffisante est donc que la matrice $(2\lambda\sigma^2 I_p - \lambda^2\beta\beta')$ soit définie positive.

Or la matrice $\beta\beta'$ est de rang 1, son image étant engendrée par β, et son unique valeur propre non nulle vaut $\|\beta\|^2$. En effet tout vecteur v de l'image s'écrit $v = \beta\beta' u = (\beta' u)\beta$ pour un certain $u \in \mathbb{R}^p$ et on vérifie que $(\beta\beta')\beta = \beta\|\beta\|^2$. Cela implique que les valeurs propres de $(2\lambda\sigma^2 I_p - \lambda^2\beta\beta')$ valent $2\lambda\sigma^2$ (pour $p-1$ d'entre elles) et $2\lambda\sigma^2 - \lambda^2\|\beta\|^2$. Ces valeurs propres sont strictement positives si et seulement si $0 < \lambda < 2\sigma^2/\|\beta\|^2$, qui est donc une condition suffisante pour que $EQM(\hat{\beta}) - EQM(\hat{\beta}_{\text{ridge}})$ soit définie positive. □

L'estimateur ridge peut-être vu d'une façon alternative. Il est solution du problème de minimisation sous contrainte suivant

$$\hat{\beta}_{\text{ridge}} = \text{argmin}_{\beta\in\mathbb{R}^p}\|Y - X\beta\|^2 \quad \text{sous la contrainte } \sum_{j=1}^{p}\beta_j^2 \leq \kappa$$

pour un certain $\kappa > 0$. Autrement dit, l'estimateur ridge est construit comme l'estimateur par MCO mais sous la contrainte que sa norme (donc ses coefficients) n'explose pas. Le Lagrangien associé à ce problème s'écrit

$$\|Y - X\beta\|^2 + \lambda(\|\beta\|^2 - \kappa)$$

où λ est le multiplicateur de Lagrange. En annulant le gradient par rapport à β on retrouve la solution

$$\hat{\beta}_{\text{ridge}} = (X'X + \lambda I_p)^{-1} X'Y.$$

Le paramètre λ est lié à κ par l'identité $\|\hat{\beta}_{\text{ridge}}\|^2 = \kappa$ (en annulant la dérivée du Lagrangien par rapport à λ), mais cette relation ne conduit pas à une écriture simple de $\hat{\beta}_{\text{ridge}}$ en fonction de κ. L'utilisation en pratique de l'estimateur $\hat{\beta}_{\text{ridge}}$ nécessite de choisir le paramètre κ ou de façon équivalente λ. Etant donné la forme explicite en fonction de λ, c'est cette paramétrisation qui est retenue, et le problème en pratique consiste donc à trouver la meilleure valeur de λ possible.

La proposition 4.2.1 assure que pour λ suffisamment petit, $\hat{\beta}_{\text{ridge}}$ est préférable à l'estimateur des MCO. Néanmoins si on choisit λ trop petit, l'intérêt est nul puisque pour $\lambda = 0$, $\hat{\beta}_{\text{ridge}}$ coincide avec l'estimateur des MCO. Par ailleurs la borne supérieure $2\sigma^2/\|\beta\|^2$ est inutilisable en pratique puisqu'on ne connait ni σ^2 ni β. Le choix de λ se fait donc par validation croisée (voir le chapitre précédent pour les différentes stratégies possibles) : on choisit le paramètre λ qui conduit à une estimation de l'erreur test minimale.

Mise en oeuvre sous R :

Fonction `glmnet` de la librairie du même nom en choisissant $\alpha = 0$ dans les options de la fonction. La validation croisée pour choisir λ peut se faire avec la fonction `cv.glmnet`.

Ces fonctions commencent par centrer et réduire les variables X avant d'estimer β sous contraintes. Cette démarche est naturelle car si les variables ne sont pas à la même échelle, la contrainte $\sum_{j=1}^p \beta_j^2 \leq \kappa$ impactera différemment chaque variable. Les estimations retournées par les fonctions sont néanmoins dans l'échelle initiale.

4.3 Régression Lasso

4.3.1 Principe

La régression Lasso (pour Least Absolute Shrinkage and Selection Operator) suit la même idée que la régression ridge, mais au lieu de contraindre la

norme ℓ^2 de β à ne pas exploser, elle contraint sa norme ℓ^1:

$$\hat{\beta}_{\text{Lasso}} = \text{argmin}_{\beta\in\mathbb{R}^p}\|Y - X\beta\|^2 \quad \text{sous la contrainte } \sum_{j=1}^{p} |\beta_j| \leq \kappa.$$

Cette modification semble minime mais elle a des conséquences importantes : alors que l'estimateur Ridge a tendance à réduire la valeur des coefficients (effet shrinkage) pour satisfaire la contrainte, l'estimateur Lasso en annule carrément certains tout en réduisant les autres. Ainsi, contrairement au Ridge qui garde toutes les variables, le Lasso effectue indirectement une sélection des variables. Cet effet, dû à la forme de la contrainte Lasso, est illustré dans la figure 4.3 et est rendu explicite par la résolution du problème exposée dans la partie suivante.

Le Lagrangien associé au problème d'optimisation s'écrit

$$\|Y - X\beta\|^2 + 2\lambda(\sum_{j=1}^{p} |\beta_j| - \kappa) \tag{4.5}$$

où 2λ est le multiplicateur de Lagrange lié à κ par la contrainte $\sum_{j=1}^{p} |\beta_j| = \kappa$ (obtenu en annulant la dérivée par rapport à λ). Le choix 2λ plutôt que λ n'a aucune incidence et est effectué pour simplifier quelques calculs dans la résolution du problème (voir partie suivante). Comme pour l'estimateur Ridge, on a le choix de paramétrer le problème par κ ou par λ et c'est ce dernier choix qui est préféré pour des raisons pratiques.

Le point négatif du Lasso est qu'il n'y a pas de formule explicite pour la solution $\hat{\beta}_{\text{Lasso}}$ à $\lambda > 0$ donné. En particulier, on ne peut pas dériver le Lagrangien par rapport à β puisque la présence des valeurs absolues le rend non dérivable sur $\mathbb{R}^p$. La partie suivante présente néanmoins des algorithmes très efficaces pour la résolution du problème d'optimisation.

D'un point de vue pratique, comme pour les méthodes précédentes (PCR, PLS, Ridge), on commence par centrer et réduire les variables pour éviter les effets d'échelle, puis on choisit le paramètre de régularisation λ par validation croisée.

Mise en oeuvre sous R :

Fonction `glmnet` de la librairie du même nom en choisissant $\alpha = 1$ dans les options de la fonction (choix par défaut). La validation croisée pour choisir λ peut se faire avec la fonction `cv.glmnet`. La résolution du Lasso se

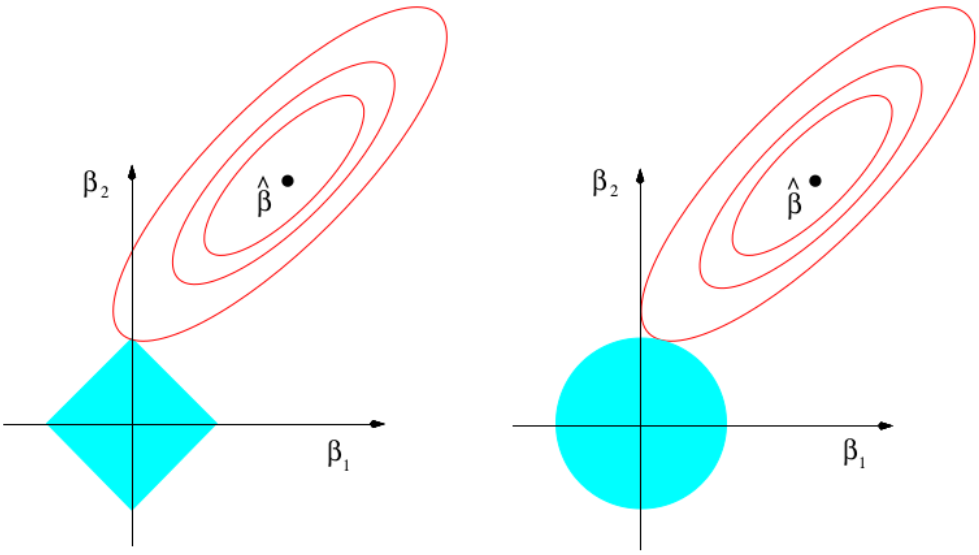

Figure 4.3: Figure extraite de l'ouvrage ESL. Parmi l'ensemble des valeurs possibles (β_1, β_2) lorsque $p = 2$, on observe le point $\hat{\beta}$ représentant l'estimateur des MCO qui minimise $SCR(\beta) = \|\hat{Y} - X\beta\|^2$. Les ellipses rouges sont les lignes de niveau de $SCR(\beta)$, dont la valeur augmente au fur et à mesure que l'on s'éloigne de $\hat{\beta}$. La solution Lasso (à gauche) est le point de contact de ces ellipses avec l'ensemble vérifiant la contrainte souhaitée $|\hat{\beta}_1| + |\hat{\beta}_2| \leq \kappa$. De même à droite le point de contact est la solution ridge vérifiant $\hat{\beta}_1^2 + \hat{\beta}_2^2 \leq \kappa$. La forme de la contrainte Lasso facilite des solutions sur les "coins" de l'ensemble, pour lesquels une coordonnée est nulle.

fait dans `glmnet` par l'algorithme de descente par coordonnée (voir la partie suivante).

Fonction `lars` de la librairie du même nom. La validation croisée se fait avec `cv.lars`. La résolution dans `lars` se fait par l'algorithme LARS-Lasso (voir la partie suivante).

Les fonctions précédentes normalisent les variables automatiquement par défaut. La partie 4.3.3 donne plus de précisions concernant ces deux fonctions et leur comparaison.

4.3.2 Résolution

D'après l'expression (4.5) du Lagrangien, on en déduit que pour $\lambda > 0$ donné, l'estimateur Lasso $\hat{\beta}$ est le vecteur qui minimise

$$L(\beta) = \frac{1}{2}\|Y - X\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{4.6}$$

En d'autres termes $\hat{\beta} = \text{argmin}_{\beta \in \mathbb{R}^p} L(\beta)$.

Proposition 4.3.1. *Pour tout $\lambda > 0$, il existe toujours une solution Lasso $\hat{\beta}$ au problème d'optimisation précédent. Cette dernière n'est pas nécessairement unique mais la prévision $X\hat{\beta}$ est unique.*

Proof. L'existence d'une solution est garantie par la convexité de $L(\beta)$. Mais comme cette fonction n'est pas différentiable, la solution n'est pas nécessairement unique. Soit deux solutions différentes $\hat{\beta}_1$ et $\hat{\beta}_2$ et posons $\bar{\beta} = (\hat{\beta}_1 + \hat{\beta}_2)/2$ leur moyenne. Par convexité de l'application $x \mapsto \|x\|^2$, on sait que pour tout x, y, $\|(x+y)/2\|^2 \leq (\|x\|^2 + \|y\|^2)/2$ et l'inégalité est stricte ssi $x \neq y$. Si $X\hat{\beta}_1 \neq X\hat{\beta}_2$, en utilisant le résultat précédent et l'inégalité triangulaire pour la norme ℓ_1 on obtient

$$L(\bar{\beta}) < (L(\hat{\beta}_1) + L(\hat{\beta}_2))/2.$$

Or par définition de la solution Lasso, pour tout $\beta \in \mathbb{R}^p$, $L(\hat{\beta}_1) \leq L(\beta)$ et de même $L(\hat{\beta}_2) \leq L(\beta)$. En particulier pour $\beta = \bar{\beta}$, on aboutit à la contradiction $L(\bar{\beta}) < L(\bar{\beta})$, ce qui montre que $X\hat{\beta}_1 = X\hat{\beta}_2$. □

La proposition suivante établit deux propriétés importantes de l'estimateur Lasso qui sont à la base des algorithmes de résolution présentés ensuite.

Proposition 4.3.2. *On a les deux propriétés suivantes.*

1. *$\hat{\beta}$ minimise* (4.6) *si et seulement si, pour $j = 1, \dots, p$,*

$$X_j'(Y - X\hat{\beta}) = \lambda \, sign(\hat{\beta}_j) \quad si \ \hat{\beta}_j \neq 0 \tag{4.7}$$

et

$$|X_j'(Y - X\hat{\beta})| \leq \lambda \quad si \ \hat{\beta}_j = 0. \tag{4.8}$$

Ceci implique que pour les variables actives (associées à un $\hat{\beta}_j \neq 0$),

$$|X_j'(Y - X\hat{\beta})| = \lambda \tag{4.9}$$

tandis que pour les variables non actives (associées à un $\hat{\beta}_j = 0$), seule l'inégalité (4.8) *est vérifiée.*

2. *En supposant que chaque variable X_j est centrée et réduite (standardisation classique lors d'une procédure Lasso), la j-ème coordonnée $\hat{\beta}_j$ de la solution Lasso s'exprime en fonction des autres coordonnées de la façon suivante*

$$\hat{\beta}_j = \begin{cases} 0 & si \ |R_j| \leq \lambda, \\ (R_j - \lambda)/n & si \ R_j > \lambda, \\ (R_j + \lambda)/n & si \ R_j < -\lambda, \end{cases} \tag{4.10}$$

où $R_j = X_j'(Y - \sum_{k \neq j} X_k \hat{\beta}_k)$.

Proof. On rappelle les notions d'optimisation suivantes. Soit f une fonction convexe de $\mathbb{R}^d$ dans $\mathbb{R}$. Un sous-gradient $s \in \mathbb{R}^d$ de f au point $x \in \mathbb{R}^d$ vérifie

$$f(x + h) \geq f(x) + s'h, \quad \forall h \in \mathbb{R}^d.$$

Si la fonction f est différentiable en x, alors le sous-gradient de f en x est unique et correspond à son gradient en x. Dans le cas contraire, on note $\partial f(x)$ l'ensemble (éventuellement vide) des sous-gradients possibles de f en x. Selon la règle de Fermat (ou condition d'optimalité du premier ordre), le point x^* est le minimum d'une fonction convexe f ssi $0 \in \partial f(x^*)$.

La solution Lasso minimise la fonction convexe $L(\beta)$. On cherche donc son sous-gradient. Il vaut

$$\partial L(\beta) = -X'Y + X'X\beta + \lambda \, \partial \left(\sum_{j=1}^{p} |\beta_j| \right).$$

Or le sous-gradient de $x \mapsto |x|$, pour $x \in \mathbb{R}$, vaut $sign(x)$ si $x \neq 0$ et correspond à l'intervalle $[-1, 1]$ si $x = 0$. On en déduit que la j-ème composante de $\partial L(\beta)$ vaut

$$\partial L(\beta)_j = \begin{cases} X_j'(X\beta - Y) + \lambda sign(\beta_j) & \text{si } \beta_j \neq 0, \\ \left[X_j'(X\beta - Y) - \lambda;\, X_j'(X\beta - Y) + \lambda\right] & \text{si } \beta_j = 0. \end{cases}$$

Donc $\hat{\beta}$ est une solution Lasso ssi $0 \in \partial L(\hat{\beta})$ ssi les conditions (4.7) et (4.8) sont réalisées pour $j = 1, \ldots, p$. La condition nécessaire (mais non suffisante) (4.9) s'en déduit immédiatement.

Concernant le second point de la proposition, on remarque que

$$X_j'(Y - X\hat{\beta}) = X_j'Y - X_j'X_j\hat{\beta}_j - X_j' \sum_{k \neq j} X_k\hat{\beta}_k = R_j - X_j'X_j\hat{\beta}_j = R_j - n\hat{\beta}_j,$$

où la dernière égalité provient de la normalisation $X_j'X_j = n$. Ainsi lorsque $\hat{\beta}_j = 0$, $|R_j| = |X_j'(Y - X\hat{\beta})| \leq \lambda$ d'après (4.8), ce qui montre la première égalité dans (4.10). Lorsque $\hat{\beta}_j \neq 0$, $n\hat{\beta}_j = R_j - \lambda sign(\hat{\beta}_j)$ en utilisant (4.7). Si $\hat{\beta}_j > 0$, on déduit de la première relation que $R_j = n\hat{\beta}_j + \lambda > 0$ et de même si $\hat{\beta}_j < 0$, $R_j < 0$, donc $\hat{\beta}_j$ et R_j ont même signe. Ainsi si $R_j \geq \lambda > 0$, $n\hat{\beta}_j = R_j - \lambda sign(\hat{\beta}_j) = R_j - \lambda sign(R_j) = R_j - \lambda$ et si $R_j < 0$, $n\hat{\beta}_j = R_j + \lambda$. □

Algorithme de descente par coordonnée

Cet algorithme utilise la seconde propriété de la proposition 4.3.2 : si on connait toutes les coordonnées de $\hat{\beta}$ sauf la j-ème $\hat{\beta}_j$, cette dernière s'obtient explicitement grâce à (4.10). On initialise donc $\hat{\beta}$ à une certaine valeur $\hat{\beta}^{(0)}$ puis on met à jour chaque coordonnée itérativement jusqu'à convergence (c'est à dire jusqu'à ce que la fonction à minimiser $L(\beta)$ ne décroît plus à une précision ϵ près) :

1. Pour $j = 1, \ldots, p$, on calcule $R_j = X_j'(Y - \sum_{k \neq j} X_k\hat{\beta}_k^{(0)})$ et on met à jour $\hat{\beta}_j$ par (4.10).

2. Si $|L(\hat{\beta}) - L(\hat{\beta}^{(0)})| < \epsilon$, la convergence est atteinte et l'algorithme s'arrête, sinon retour à 1 avec $\hat{\beta}^{(0)} \leftarrow \hat{\beta}$.

Puisque la fonction $L(\beta)$ est convexe, la descente par coordonnée converge nécessairement vers le minimum global de $L(\beta)$ (ou l'un d'entre eux s'il n'est pas unique). Il s'agit de la méthode utilisée dans `glmnet` pour la résolution du Lasso.

Algorithme LARS-Lasso

Cet algorithme fournit toutes les solutions $\hat{\beta}(\lambda)$ pour tout $\lambda \geq 0$, ce que l'on appelle le "chemin Lasso", pour un coût algorithmique faible. Il exploite les propriétés suivantes :

- pour $\lambda = \infty$, $\hat{\beta}(\infty) = 0$,
- pour $\lambda = 0$, $\hat{\beta}(0) = \hat{\beta}^{MCO}$ où $\hat{\beta}^{MCO}$ est l'estimateur par MCO,
- entre ces deux extrêmes, (4.7) montre que $\hat{\beta}(\lambda)$ évolue de façon linéaire par morceaux en fonction de λ. Précisément l'évolution est linéaire tant que les variables actives restent les mêmes et un changement de dynamique s'opère uniquement lorsqu'une nouvelle variable devient active (ou une active devient inactive dans le cas de la version LARS-Lasso).

L'algorithme LARS (Least angle regression) construit une solution $\hat{\beta}(\lambda)$, pour tout $\lambda \geq 0$, vérifiant (4.9) et (4.8). Puisque seul (4.9) est assuré et non (4.7), il ne garantit pas d'obtenir la solution Lasso (mais une solution très proche). L'algorithme LARS-Lasso, présenté dans un second temps, est une modification de l'algorithme LARS permettant d'obtenir la solution Lasso.

Algorithme LARS. On construit $\hat{\beta}(\lambda)$, pour tout $\lambda \geq 0$, en faisant décroitre λ de $+\infty$ à 0.

Soit $\lambda_1 = max_j |X_j' Y|$.

Pour tout $\lambda \geq \lambda_1$, $\hat{\beta}(\lambda) = 0$.

Pour $\lambda < \lambda_1$, $\hat{\beta}(\lambda)$ est obtenu par l'algorithme suivant. On l'initialise en posant $k = 1$ et en définissant la première variable active comme étant celle réalisant $|X_j' Y| = \lambda_1$.

1. Partant de $\lambda = \lambda_k$, tant que pour toutes les variables non-actives X_j la condition $|X_j'(Y - X\hat{\beta}(\lambda))| < \lambda$ reste vérifiée, on fait décroitre λ en définissant
$$\hat{\beta}(\lambda) = \frac{\lambda}{\lambda_k} \hat{\beta}(\lambda_k) + (1 - \frac{\lambda}{\lambda_k}) \hat{\beta}_k^{MCO}, \tag{4.11}$$

où $\hat{\beta}_k^{MCO}$ est l'estimateur des MCO de Y sur les k variables actives de l'étape courante, complété avec des 0 pour les variables non-actives.

2. On nomme λ_{k+1} la dernière valeur de λ issue de l'étape précédente. La variable non-active X_j ayant réalisée la condition de sortie $|X_j'(Y - X\hat{\beta}(\lambda_{k+1}))| = \lambda_{k+1}$ devient la $(k+1)$-ème variable active.

3. Retour à l'étape 1 avec $k \leftarrow k+1$.

Proposition 4.3.3. *L'algorithme LARS fournit une solution $\hat{\beta}(\lambda)$ vérifiant, pour tout $\lambda \geq 0$,* (4.9) *et* (4.8).

Proof. Lorsque $\lambda \geq \lambda_1$, $\hat{\beta}(\lambda) = 0$ et dans ce cas la condition (4.8) s'écrit $|X_j'Y| \leq \lambda$ pour tout j. Cette condition est bien vérifiée par définition de λ_1. De plus, tous les coefficients $\hat{\beta}_j(\lambda)$ étant nuls, (4.9) n'a pas besoin d'être vérifiée. On remarque par ailleurs que si $\lambda > \lambda_1$, $|X_j'Y| < \lambda$.

On montre que les propriétés (4.9) et (4.8) sont vraies lorsque $\lambda \leq \lambda_1$ en les montrant à chaque étape de l'algorithme par récurrence sur k. On vérifiera également à chaque étape que l'inégalité dans (4.8) est stricte pour les variables non-actives.

(Dans la suite de la preuve, nous supposerons pour simplifier qu'une seule variable à la fois peut devenir active. La preuve s'adapte sans problème majeur au cas où plusieurs variables peuvent être actives en même temps.)

A l'étape $k = 1$, il n'y a qu'une seule variable active. Supposons sans perte de généralité qu'il s'agit de X_p. Par définition $\lambda_1 = |X_p'Y|$ et pour toutes les variables non-actives (c'est à dire autres que X_p), $|X_j'Y| < \lambda_1$. En $\lambda = \lambda_1$, on a donc bien l'inégalité stricte dans (4.8) qui est vérifée pour les variables non actives. Elle le restera tant que l'on reste dans l'étape courante, par construction de l'algorithme LARS (dès que l'inégalité devient une égalité, on passe à l'étape suivante). On a par ailleurs $\hat{\beta}(\lambda_1) = 0$ et $X\hat{\beta}_1^{MCO} = X_p(X_p'X_p)^{-1}X_p'Y$. Ainsi

$$X_p'(Y - X\hat{\beta}(\lambda)) = X_p'\left(Y - (1 - \frac{\lambda}{\lambda_1})X_p(X_p'X_p)^{-1}X_p'Y\right) = \frac{\lambda}{\lambda_1}X_p'Y$$

dont la valeur absolue vaut λ, ce qui montre (4.9).

Supposons à présent que (4.9) et (4.8) sont vraies jusqu'à l'étape $k-1$ et montrons-les pour l'étape k. Toutes les variables inactives satisfont (4.8) avec une inégalité stricte tout au long de cette étape par construction de

l'algorithme LARS. Concernant les k variables actives à l'étape k, notons $\tilde{X}_k$ la sous matrice de X qui les contient. Par définition $\hat{\beta}_k^{MCO}$ est obtenu par projection de Y sur les k variables actives ce qui implique $X\hat{\beta}_k^{MCO} = P_{[\tilde{X}_k]}Y$. Par définition de cette projection $\tilde{X}_k' X\hat{\beta}_k^{MCO} = \tilde{X}_k' Y$ et donc

$$\begin{aligned}
\tilde{X}_k'(Y - X\hat{\beta}(\lambda)) &= \tilde{X}_k' \left(Y - X \left(\frac{\lambda}{\lambda_k}\hat{\beta}(\lambda_k) + (1 - \frac{\lambda}{\lambda_k})\hat{\beta}_k^{MCO} \right) \right) \\
&= \tilde{X}_k' Y - \frac{\lambda}{\lambda_k}\tilde{X}_k' X\hat{\beta}(\lambda_k) - (1 - \frac{\lambda}{\lambda_k})\tilde{X}_k' X\hat{\beta}_k^{MCO} \\
&= \tilde{X}_k' Y - \frac{\lambda}{\lambda_k}\tilde{X}_k' X\hat{\beta}(\lambda_k) - (1 - \frac{\lambda}{\lambda_k})\tilde{X}_k' Y \\
&= \frac{\lambda}{\lambda_k}\tilde{X}_k'(Y - X\hat{\beta}(\lambda_k)).
\end{aligned}$$

Cela signifie que pour toute les variables actives X_j de l'étape k,

$$X_j'(Y - X\hat{\beta}(\lambda)) = \frac{\lambda}{\lambda_k} X_j'(Y - X\hat{\beta}(\lambda_k)).$$

Par hypothèse de récurrence, $|X_j'(Y - X\hat{\beta}(\lambda_k))| = \lambda_k$ pour les $(k-1)$ variables actives de l'étape précédente. Cette égalité est également vraie pour la nouvelle variable active de l'étape k puisque il s'agit précisément de la condition de son activation. Ainsi (4.9) est vérifiée.

□

L'algorithme LARS ne répond que partiellement au problème LASSO car il garantit (4.9) et (4.8) mais pas (4.7). La différence a lieu lorsqu'un des coefficients actifs $\hat{\beta}_j$ change de signe, autrement dit lorsque $\hat{\beta}_j(\lambda) = 0$ pour un certain λ, alors que la variable associée est bien active. Pour prendre en compte ce cas et répondre au problème Lasso, il suffit d'ajouter le test suivant à la première étape de l'algorithme LARS

Algorithme LARS-Lasso:

1'. Si pour une variable active X_j de l'étape k, son coefficient $\hat{\beta}_j(\lambda)$ devient nul pour un certain $\lambda_0 < \lambda_k$, lorsque $\hat{\beta}(\lambda)$ évolue comme décrit en (4.11), alors on met à jour la dynamique de la façon suivante :

- cette variable X_j devient inactive,

– on fait décroitre λ en définissant $\hat{\beta}(\lambda)$ comme en (4.11), mais en s'appuyant uniquement sur les variables actives restantes :

$$\hat{\beta}(\lambda) = \frac{\lambda}{\lambda_0}\hat{\beta}(\lambda_0) + (1 - \frac{\lambda}{\lambda_0})\hat{\beta}^{MCO},$$

où $\hat{\beta}^{MCO}$ est l'estimateur des MCO de Y sur les variables actives restantes de l'étape courante, complété avec des 0 pour les variables non-actives.

On admet que cette modification permet effectivement de garantir (4.7).

La différence entre LARS et Lasso est illustrée dans la figure 4.4. Les solutions sont identiques jusqu'à ce qu'un des coefficients actifs revienne à 0, la modification étant effectuée par l'étape 1' décrite ci-dessus.

4.3.3 Aparté `R` : comparaison entre `glmnet` et `lars`

Les fonctions `glmnet` et `lars` calculent le chemin Lasso complet pour tout λ. Plus précisément :

- La fonction `lars` avec l'option par défaut `type='lasso'` calcule le chemin Lasso en utilisant l'algorithme LARS-Lasso. La solution est exacte pour tout λ.

- La fonction `glmnet` avec l'option par défaut $\alpha = 1$ approche le chemin Lasso pour une grille de valeurs de λ (choisie automatiquement par défaut), en lesquelles elle approche la solution Lasso par l'algorithme de descente par coordonnées. La précision de l'approximation en ces λ est gérée par l'option `thresh` qui correspond à la précision ϵ du critère d'arrêt de l'algorithme. Par défaut `thresh` vaut $1e-7$. L'approche de `glmnet` n'est donc pas aussi exacte qu'avec la fonction `lars` mais elle est dans certains cas plus rapide, et surtout `glmnet` autorise d'autres pénalités que le Lasso en jouant sur le paramètre α ("elastic net", voir la partie suivante) et à d'autres modèles que la régression linéaire, dont en particulier la régression logistique, ce que ne permet pas `lars`.

Dans une utilisation classique des ces fonctions, on détermine le meilleur λ par validation croisée (fonction `cv.lars` pour `lars` et fonction `cv.glmnet` pour `glmnet`). On accède alors aux coefficients Lasso et aux prévisions associés à

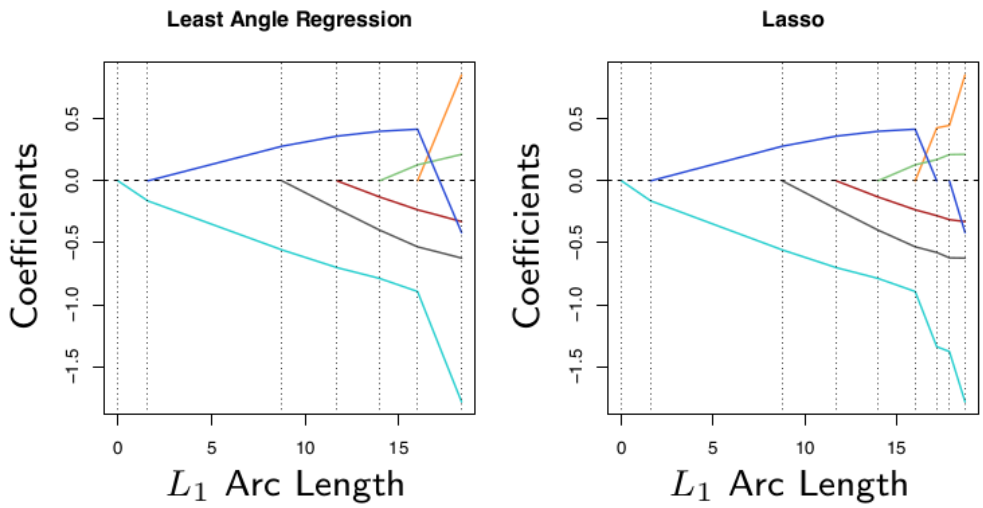

Figure 4.4: Figure extraite de l'ouvrage ESL illustrant l'estimation par LARS (à gauche) et Lasso (à droite) dans un exemple faisant intervenir $p = 6$ variables explicatives. Chaque figure montre l'évolution des 6 coefficients $\hat{\beta}_j(\lambda)$ lorsque λ varie (de gauche à droite) de $\lambda = \lambda_1$, le cas où $\hat{\beta} = 0$, à $\lambda = 0$, le cas où $\hat{\beta} = \hat{\beta}^{MCO}$. La légende indique "$L_1$ Arc Length" à la place de λ : il s'agit d'une quantité valant 0 lorsque $\lambda = \lambda_1$ pour croître linéairement (jusqu'à environ 20 dans cet exemple) lorsque λ décroit jusque 0. La différence entre LARS et Lasso survient lorsqu'un des coefficients actifs (celui en bleu foncé) revient en 0.

ce λ grâce aux fonctions `coef` et `predict` en spécifiant l'option `s=`λ dans ces fonctions. Une normalisation des données est effectuée par défaut dans `lars` et `glmnet`, mais la transformation inverse est automatiquement effectuée au moment de retourner les coefficients et les prévisions, qui sont donc à la bonne échelle des données initiales.

Mais `glmnet` et `lars` n'utilisent pas la même définition par défaut du paramètre de régularisation λ et ne normalisent pas de la même manière les données. Dans l'utilisation classique qu'on a de ces fonctions, cela n'a pas d'importance car le choix de λ renvoyé par `cv.lars` (respectivement par `cv.glmnet`) est interprété correctement par `lars` (respectivement par `glmnet`), et les normalisations sont bien sûr gérées de façon cohérente par chaque fonction.

Cependant il est difficile d'estimer exactement le même modèle (i.e. avec la même pénalité) avec ces deux fonctions, ce qui pourrait être utile afin de comparer leurs approches. En fait, la fonction `lars` minimise

$$\frac{1}{2}\|Y - X\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{4.12}$$

où Y et X sont les versions normalisées de `lars` tandis que la fonction `glmnet` minimise

$$\frac{1}{2n}\|Y - X\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

où Y et X sont les versions normalisées de `glmnet` (différentes de `lars`).

Si on souhaite estimer exactement le même modèle, pour la même pénalité, avec ces deux fonctions, il faut :

- Travailler sans standardisation (option `normalize=F` pour `lars` et `standardize=F` pour `glmnet`). Il est toujours possible de normaliser comme on le souhaite les données au préalable.
- Préciser `mode="lambda"` pour `lars` lorsqu'on spécifie la valeur λ souhaitée dans `coef` ou `predict`, pour que la pénalisation ait le même sens qu'avec `glmnet`.
- Si on a choisi la pénalité λ pour `lars`, il faut choisir la pénalité λ/n pour `glmnet` (d'après la forme des fonctions à minimiser ci-dessus).

Exemple : pour estimer un modèle associé à la pénalité $\lambda = 4$ dans (4.12)

```
fit.lars=lars(x, y, normalize = F)
beta.lars=coef(fit.lars, s=4, mode="lambda")

fit.glmnet=glmnet(x, y, standardize = F)
beta.glmnet=coef(fit.glmnet, s = 4/n)
```

Pour affiner l'approximation de `glmnet` (si cette dernière est trop différente de la solution `lars`), on peut imposer un critère d'arrêt plus strict dans `glmnet` avec l'option `thresh`, par exemple `thresh = 1e-20`. Cela nécessitera peut-être d'augmenter le nombre maximal d'itérations autorisées pour atteindre la convergence avec l'option `maxit`. De même l'estimation des coefficients pour le λ choisi (ici $4/n$) correspond à une interpolation linéaire des coefficients estimés aux λ les plus proches dans la grille choisie dans `glmnet`. En ajoutant l'option `exact=T` à `coef`, une nouvelle grille de λ est calculée au voisinage du λ choisi pour affiner l'estimation des coefficients. Dans ce cas, il faut également rappeler en option les données `x=x,y=y`. Ce raffinement donnerait donc

```
fit.glmnet=glmnet(x, y, standardize = F, thresh = 1e-20)
beta.glmnet=coef(fit.glmnet, s = 4/n, exact = T, x=x, y=y)
```

4.3.4 Aspects théoriques

L'étude théorique de l'estimateur Lasso est délicate. Par exemple son biais et sa variance sont non explicites en général.

En supposant que le vrai modèle ne contient qu'un nombre limité de variables explicatives (hypothèse de parcimonie), deux types de questions motivent la théorie :

- la qualité d'estimation du Lasso : les coefficients estimés sont-ils de bonnes approximations des vrais coefficients ?
- le pouvoir de sélection du Lasso : à quel point l'estimateur Lasso est-il capable de retrouver les bonnes variables ?

Ces deux objectifs ne se réalisent pas nécessairement en même temps. Par exemple, dans un modèle de régression Gaussien, l'estimateur par MCO est consistant dès que $(X'X)^{-1} \to 0$ mais il ne sélectionne aucune variable car

aucune de ses coordonnées n'est nulle presque surement (sa loi est Gaussienne).

Pour illustrer les principales propriétés du Lasso, on se restreint dans la proposition suivante au cas particulier où les variables explicatives sont centrées, réduites et non-corrélées (i.e. orthogonales) entre elles. Cela revient à supposer que $X'X = nI_p$. (Cette situation est largement artificielle car en pratique il y a toujours une corrélation empirique non nulle, même minime, entre les variables explicatives. De plus cette hypothèse implique que nécessairement $p \leq n$ car il n'est pas possible d'avoir plus de n vecteurs orthogonaux 2 à 2 dans $\mathbb{R}^n$). Dans ce cas, la solution Lasso à $\lambda > 0$ fixé est explicite (voir TD) et vaut, pour $j = 1, \dots, p$,

$$\hat{\beta}(\lambda) = \begin{cases} 0 & \text{si} \quad |\hat{\beta}_j^{MCO}| \leq \lambda/n, \\ \hat{\beta}_j^{MCO} - \lambda/n & \text{si} \quad \hat{\beta}_j^{MCO} > \lambda/n, \\ \hat{\beta}_j^{MCO} + \lambda/n & \text{si} \quad \hat{\beta}_j^{MCO} < -\lambda/n, \end{cases}$$

où $\hat{\beta}_j^{MCO}$ désigne l'estimateur MCO de la régression. Les deux caractéristiques principales de l'approche Lasso apparaissent : elle **sélectionne** les variables les plus pertinentes (celles pour lesquelles $|\hat{\beta}_j^{MCO}| > \lambda/n$ dans le cas précédent) et elle **réduit** les coefficients MCO des variables sélectionnées en rapprochant leur valeur de 0, ce qui est appelé la propriété de "seuillage doux" du Lasso.

La proposition suivante montre que si le paramètre de régularisation λ tend vers l'infini avec n à la bonne vitesse (moins vite que n mais plus vite que $\sqrt{n}$), l'estimateur Lasso est consistant et sélectionne les bonnes variables asymptotiquement. Elle souligne également un défaut bien connu du Lasso : à cause de sa propriété de seuillage doux, les coefficients des variables "importantes" ont tendance à être sous-estimés (en valeur absolue).

Dans le cas général (au delà de $X'X = nI_p$), le même type de propriétés a été établi dans les années 2000-2010, mais sous une condition importante assez restrictive appelée "condition d'irrépresentabilité". Cette dernière suppose que les variables non-pertinentes ne doivent pas être trop corrélées avec les variables pertinentes (condition évidemment vérifiée si $X'X = nI_p$). Cette condition témoigne d'un autre défaut du Lasso : cette méthode est sensible au problème de multicolinéarité, comme les MCO (voir TD à ce sujet).

Proposition 4.3.4. *Dans un modèle de régression Gaussien de paramètre β pour lequel $X'X = nI_p$, l'estimateur Lasso $\hat{\beta}(\lambda_n)$ associé au paramètre de régularisation λ_n vérifie les propriétés suivantes.*

i) Si $\lambda_n/n \to 0$, il est consistant : $\hat\beta(\lambda_n) \to \beta$ en probabilité lorsque $n \to \infty$.

ii) Si $\lambda_n/n \to 0$ et $\lambda_n/\sqrt{n} \to \infty$, il sélectionne asymptotiquement les bonnes variables dans le sens où, en notant $\mathcal{J}$ l'ensemble des indices j pour lesquels $\beta_j = 0$,

$$\mathbb{P}\left(\forall j \in \mathcal{J},\, \hat\beta_j(\lambda_n) = 0 \quad et \quad \forall j \notin \mathcal{J},\, \hat\beta_j(\lambda_n) \neq 0\right) \to 1.$$

iii) L'estimation des coefficients non nuls est biaisée vers 0, dans le sens où $|\mathbb{E}(\hat\beta_j(\lambda_n))| < \beta_j$ dès que $|\beta_j| > \lambda_n/n$.

Proof. *i*) Pour la consistance on écrit, pour tout $\epsilon > 0$, pour tout $j = 1, \dots, p$,

$$\begin{aligned}\mathbb{P}\left(|\hat\beta_j(\lambda_n) - \beta_j| > \epsilon\right) &= \mathbb{P}\left(|\hat\beta_j(\lambda_n) - \beta_j| > \epsilon,\ \hat\beta_j^{MCO} > \frac{\lambda_n}{n}\right) \\ &\quad + \mathbb{P}\left(|\hat\beta_j(\lambda_n) - \beta_j| > \epsilon,\ \hat\beta_j^{MCO} < -\frac{\lambda_n}{n}\right) \\ &\quad + \mathbb{P}\left(|\hat\beta_j(\lambda_n) - \beta_j| > \epsilon,\ |\hat\beta_j^{MCO}| < \frac{\lambda_n}{n}\right). \qquad (4.13)\end{aligned}$$

On nomme $T_1 + T_2 + T_3$ la décomposition précédente et on va montrer que chaque terme de cette somme tend vers 0. Pour T_1,

$$\begin{aligned}T_1 &= \mathbb{P}\left(|\hat\beta_j(\lambda_n) - \beta_j| > \epsilon,\ \hat\beta_j^{MCO} > \frac{\lambda_n}{n}\right) \\ &= \mathbb{P}\left(|\hat\beta_j^{MCO} - \frac{\lambda_n}{n} - \beta_j| > \epsilon,\ \hat\beta_j^{MCO} > \frac{\lambda_n}{n}\right) \\ &\le \mathbb{P}\left(|\hat\beta_j^{MCO} - \frac{\lambda_n}{n} - \beta_j| > \epsilon\right) \\ &= \mathbb{P}\left(\hat\beta_j^{MCO} - \beta_j > \epsilon + \frac{\lambda_n}{n}\right) + \mathbb{P}\left(\hat\beta_j^{MCO} - \beta_j < -\epsilon + \frac{\lambda_n}{n}\right) \\ &\le \mathbb{P}\left(\hat\beta_j^{MCO} - \beta_j > \epsilon\right) + \mathbb{P}\left(\hat\beta_j^{MCO} - \beta_j < -\epsilon/2\right)\end{aligned}$$

où la dernière inégalité est vraie dès que $\lambda_n/n < \epsilon/2$ ce qui est garanti à partir d'un certain n par l'hypothèse $\lambda_n/n \to 0$. Puisque $(X'X)^{-1} = n^{-1} I_p \to 0$, $\hat\beta_j^{MCO}$ est consistant et les deux probabilités dans la majoration ci-dessus

tendent vers 0. Cela montre que $T_1 \to 0$. Un raisonnement similaire conduit de même à $T_2 \to 0$. Pour le troisième terme :

$$T_3 = \mathbb{P}\left(|\hat{\beta}_j(\lambda_n) - \beta_j| > \epsilon,\ |\hat{\beta}_j^{MCO}| < \frac{\lambda_n}{n}\right) = \mathbb{P}\left(|\beta_j| > \epsilon,\ |\hat{\beta}_j^{MCO}| < \frac{\lambda_n}{n}\right).$$

Si $\beta_j = 0$, $|\beta_j| > \epsilon$ est impossible et $T_3 = 0$. Si $\beta_j \neq 0$, en supposant que n est suffisamment grand pour garantir $\lambda_n/n < |\beta_j|/2$ (ce qui est possible car $\lambda_n/n \to 0$ par hypothèse), on a

$$T_3 \leq \mathbb{P}\left(|\hat{\beta}_j^{MCO}| < \frac{\lambda_n}{n}\right) < \mathbb{P}\left(|\hat{\beta}_j^{MCO}| < |\beta_j|/2\right).$$

Or $|\hat{\beta}_j^{MCO}| < |\beta_j|/2$ ssi $-|\beta_j|/2 - \beta_j < \hat{\beta}_j^{MCO} - \beta_j < |\beta_j|/2 - \beta_j$, ce qui implique $|\hat{\beta}_j^{MCO} - \beta_j| > |\beta_j|/2$ (considérer les cas $\beta_j > 0$ et $\beta_j < 0$). Ainsi

$$T_3 \leq \mathbb{P}\left(|\hat{\beta}_j^{MCO} - \beta_j| > |\beta_j|/2\right)$$

qui tend vers 0 par consistance de $\hat{\beta}_j^{MCO}$.

La consistance de l'estimateur Lasso est ainsi montrée.

ii) Pour la propriété de sélection de variables du Lasso,

$$\mathbb{P}\left(\forall j \in \mathcal{J},\ \hat{\beta}_j(\lambda_n) = 0 \ \text{ et } \ \forall j \notin \mathcal{J},\ \hat{\beta}_j(\lambda_n) \neq 0\right)$$
$$= \mathbb{P}\left(\forall j \in \mathcal{J},\ |\hat{\beta}_j^{MCO}| < \frac{\lambda_n}{n} \ \text{ et } \ \forall j \notin \mathcal{J},\ |\hat{\beta}_j^{MCO}| > \frac{\lambda_n}{n}\right).$$

Or le modèle étant Gaussien avec $X'X = nI_p$, $\hat{\beta}^{MCO} \sim \mathcal{N}(\beta, \sigma^2 I_p/n)$. Cela montre que les coordonnées $\hat{\beta}_j^{MCO}$ sont indépendantes entre elles. Ainsi

$$\mathbb{P}\left(\forall j \in \mathcal{J},\ \hat{\beta}_j(\lambda_n) = 0 \ \text{ et } \ \forall j \notin \mathcal{J},\ \hat{\beta}_j(\lambda_n) \neq 0\right)$$
$$= \prod_{j \in \mathcal{J}} \mathbb{P}\left(|\hat{\beta}_j^{MCO}| < \frac{\lambda_n}{n}\right) \prod_{j \notin \mathcal{J}} \mathbb{P}\left(|\hat{\beta}_j^{MCO}| > \frac{\lambda_n}{n}\right). \quad (4.14)$$

Pour tout $j \in \mathcal{J}$, $\sqrt{n}\,\hat{\beta}_j^{MCO}$ suit la loi $Z \sim \mathcal{N}(0, \sigma^2)$. Donc toutes les probabilités dans le premier produit valent $\mathbb{P}(|Z| < \lambda_n/\sqrt{n})$, qui tend vers 1

car par hypothèse $\lambda_n/\sqrt{n} \to \infty$. Par ailleurs, pour tout $j \notin \mathcal{J}$, $\sqrt{n}\,(\hat{\beta}_j^{MCO} - \beta_j) \sim Z$ et dans ce cas on obtient

$$\begin{aligned}\mathbb{P}\left(|\hat{\beta}_j^{MCO}| > \frac{\lambda_n}{n}\right) &= 1 - \mathbb{P}\left(|\hat{\beta}_j^{MCO}| \leq \frac{\lambda_n}{n}\right) \\ &= 1 - \mathbb{P}\left(-\frac{\lambda_n}{\sqrt{n}} - \sqrt{n}\beta_j \leq Z \leq \frac{\lambda_n}{\sqrt{n}} - \sqrt{n}\beta_j\right) \\ &\geq 1 - \mathbb{P}\left(|Z| \geq \sqrt{n}\,\frac{|\beta_j|}{2}\right)\end{aligned}$$

dès que $\lambda_n/n < |\beta_j|/2$, ce qui arrive nécessairement à partir d'un certain n par l'hypothèse $\lambda_n/n \to 0$. Cette dernière probabilité tend vers 0, ce qui montre que $\mathbb{P}\left(|\hat{\beta}_j^{MCO}| > \lambda_n/n\right)$ tend vers 1 pour tout $j \notin \mathcal{J}$. Ainsi (4.14) tend vers 1 lorsque $n \to \infty$ ce qui montre le résultat annoncé.

iii) Enfin pour le calcul du biais, on utilise la décomposition

$$\begin{aligned}&\mathbb{E}(\hat{\beta}_j(\lambda_n)) \\ &= \mathbb{E}\left(\hat{\beta}_j(\lambda_n)1_{|\hat{\beta}_j^{MCO}| \leq \frac{\lambda_n}{n}}\right) + \mathbb{E}\left(\hat{\beta}_j(\lambda_n)1_{\hat{\beta}_j^{MCO} > \frac{\lambda_n}{n}}\right) + \mathbb{E}\left(\hat{\beta}_j(\lambda_n)1_{\hat{\beta}_j^{MCO} < -\frac{\lambda_n}{n}}\right) \\ &= 0 + \mathbb{E}\left(\left(\hat{\beta}_j^{MCO} - \frac{\lambda_n}{n}\right)1_{\hat{\beta}_j^{MCO} > \frac{\lambda_n}{n}}\right) + \mathbb{E}\left(\left(\hat{\beta}_j^{MCO} + \frac{\lambda_n}{n}\right)1_{\hat{\beta}_j^{MCO} < -\frac{\lambda_n}{n}}\right).\end{aligned}$$

On va montrer que si $\beta_j > \lambda_n/n$, alors $0 \leq \mathbb{E}(\hat{\beta}_j(\lambda_n)) < \beta_j$. Le cas contraire $\beta_j < -\lambda_n/n$ se traite de la même manière pour montrer que $\beta_j < \mathbb{E}(\hat{\beta}_j(\lambda_n)) \leq 0$. On suppose donc $\beta_j > \lambda_n/n$ et on note f la densité de $\hat{\beta}_j^{MCO}$, c'est à dire d'une loi $\mathcal{N}(\beta_j, \sigma^2/n)$, et f^* la densité d'une loi $\mathcal{N}(0, \sigma^2/n)$. On a d'une part

$$\begin{aligned}\mathbb{E}(\hat{\beta}_j(\lambda_n)) &= \int_{\frac{\lambda_n}{n}}^{\infty}(x - \frac{\lambda_n}{n})f(x)dx + \int_{-\infty}^{-\frac{\lambda_n}{n}}(x + \frac{\lambda_n}{n})f(x)dx \\ &= \int_{\frac{\lambda_n}{n}-\beta_j}^{\infty}(x + \beta_j - \frac{\lambda_n}{n})f^*(x)dx + \int_{-\infty}^{-\frac{\lambda_n}{n}-\beta_j}(x + \beta_j + \frac{\lambda_n}{n})f^*(x)dx \\ &\geq \int_{\frac{\lambda_n}{n}-\beta_j}^{0} x f^*(x)dx + \int_0^{\infty} x f^*(x)dx + \int_{-\infty}^{-\frac{\lambda_n}{n}-\beta_j} x f^*(x)dx\end{aligned}$$

en utilisant le fait que $\beta_j - \lambda_n/n \geq 0$ et $\beta_j + \lambda_n/n \geq 0$ et en décomposant le premier intervalle d'intégration en deux. Le changement de variables $x \to -x$

pour les intégrales de gauche et de droite conduit finalement à

$$\mathbb{E}(\hat{\beta}_j(\lambda_n)) \geq \int_{\beta_j - \frac{\lambda_n}{n}}^{\beta_j + \frac{\lambda_n}{n}} x f^*(x) dx \geq 0.$$

D'autre part, en reprenant la décomposition initiale,

$$\begin{aligned}
&\mathbb{E}(\hat{\beta}_j(\lambda_n)) \\
&= \mathbb{E}\left(\hat{\beta}_j^{MCO}\left(1 - 1_{|\hat{\beta}_j^{MCO}| < \frac{\lambda_n}{n}}\right)\right) - \frac{\lambda_n}{n}\left(\mathbb{P}\left(\hat{\beta}_j^{MCO} > \frac{\lambda_n}{n}\right) - \mathbb{P}\left(\hat{\beta}_j^{MCO} < -\frac{\lambda_n}{n}\right)\right) \\
&= \beta_j - \int_{-\lambda_n/n}^{\lambda_n/n} x f(x) dx - \frac{\lambda_n}{n}\left(\mathbb{P}\left(\hat{\beta}_j^{MCO} > \frac{\lambda_n}{n}\right) - \mathbb{P}\left(\hat{\beta}_j^{MCO} < -\frac{\lambda_n}{n}\right)\right).
\end{aligned} \tag{4.15}$$

Puisque $\beta_j > \lambda_n/n$,

$$\mathbb{P}\left(\hat{\beta}_j^{MCO} < -\frac{\lambda_n}{n}\right) < \mathbb{P}\left(\hat{\beta}_j^{MCO} < \beta_j\right) = \mathbb{P}\left(\hat{\beta}_j^{MCO} > \beta_j\right) < \mathbb{P}\left(\hat{\beta}_j^{MCO} > \frac{\lambda_n}{n}\right)$$

ce qui montre que le dernier terme dans (4.15) est négatif. Concernant le terme intégrale, en utilisant le fait que f est croissante sur $]-\infty; \lambda_n/n]$ lorsque $\beta_j > \lambda_n/n$, on en déduit que pour tout $x < 0$, $xf(x) \geq xf(0)$ et pour $x \in [0; \lambda_n/n]$, $xf(x) \geq xf(0)$, d'où

$$\begin{aligned}
\int_{-\lambda_n/n}^{\lambda_n/n} x f(x) dx &= \int_{-\lambda_n/n}^{0} x f(x) dx + \int_0^{\lambda_n/n} x f(x) dx \\
&\geq f(0) \int_{-\lambda_n/n}^{0} x dx + f(0) \int_0^{\lambda_n/n} x dx = 0.
\end{aligned}$$

On connait ainsi le signe de chaque terme dans (4.15), d'où l'on déduit que $\mathbb{E}(\hat{\beta}_j(\lambda_n)) < \beta_j$.

□

4.4 Quelques généralisations des régressions Ridge et Lasso

Les qualités et défauts principaux de la régression Ridge sont :

- Qualités : robuste à la multicolinéarité, robuste à la grande dimension
- Défauts : ne sélectionne aucune variable mais rétrécit les coefficients

Et pour la régression Lasso :

- Qualités : robuste à la grande dimension, sélectionne les variables
- Défauts : sensible à la multicolinéarité, biais vers 0 des coefficients importants.

4.4.1 Elastic net

Cette méthode est un compromis entre Ridge et Lasso. L'idée est de tirer partie des qualités de sélection de l'estimateur Lasso, tout en garantissant une meilleure robustesse en cas de multicolinéarité, propriété propre à l'estimateur Ridge. L'estimateur elastic-net est solution du problème :

$$\hat{\beta}_{\text{en}} = \text{argmin}_{\beta\in\mathbb{R}^p}\|Y-X\beta\|^2 \quad \text{sous la contrainte } \sum_{j=1}^{p}\left((1-\alpha)\beta_j^2+\alpha|\beta_j|\right)\leq\kappa,$$

ou en l'écrivant à l'aide du Lagrangien

$$\hat{\beta}_{\text{en}} = \text{argmin}_{\beta\in\mathbb{R}^p}\|Y-X\beta\|^2+\lambda\sum_{j=1}^{p}\left((1-\alpha)\beta_j^2+\alpha|\beta_j|\right),$$

où λ est le paramètre de régularisation (lié à κ) qu'il convient de choisir en pratique.

On remarque que la contrainte est un compromis entre la contrainte Lasso (le cas $\alpha = 1$) et la contrainte Ridge (le cas $\alpha = 0$). Une représentation du domaine des contraintes en dimension 2 pour différentes valeurs de α est donnée dans la figure 4.5.

D'un point de vue pratique, pour α donné, on choisit le paramètre λ par validation croisée puis $\hat{\beta}_{\text{en}}$ peut être obtenu par un algorithme de descente par coordonnée. Le paramètre α peut également être choisi par validation croisée.

Mise en oeuvre sous R :

Fonctions `glmnet` et `cv.glmnet` de la librairie `glmnet`, en choisissant en option la valeur de α.

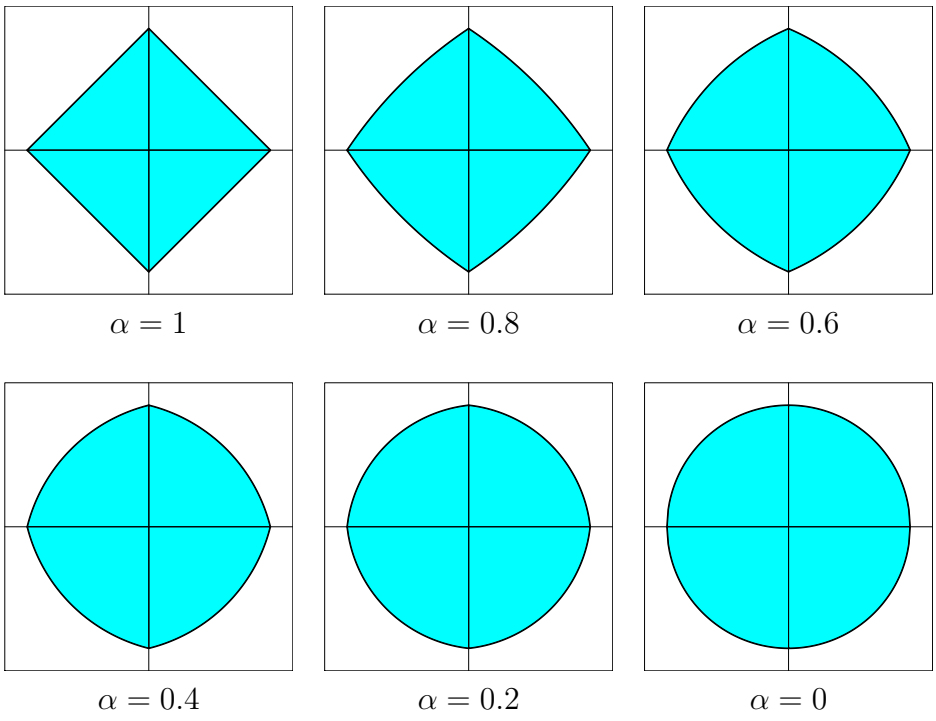

Figure 4.5: Domaine des contraintes elastic-net en dimension 2, i.e. $(1-\alpha)(\beta_1^2+\beta_2^2)+\alpha(|\beta_1|+|\beta_2|) \leq \kappa$, pour différentes valeurs de α. Le cas $\alpha = 1$ correspond au Lasso et $\alpha = 0$ au Ridge.

4.4.2 Gauss-Lasso

Pour réduire le biais dû au seuillage doux du Lasso, l'idée de la procédure Gauss-Lasso est d'effectuer l'estimation en deux étapes :

1. On effectue une régression Lasso pour connaître le support, c'est à dire les variables associées à des coefficients non nuls.

2. On effectue une régression standard par MCO sur les variables sélectionnées à la première étape, sous réserve que leur nombre soit petit devant n.

On fait ainsi confiance à la propriété de sélection du Lasso, mais on s'appuie sur les MCO pour l'estimation des coefficients importants. Cette procédure en deux étapes est assez répandue en pratique.

Attention, si un grand nombre de variables a été sélectionné par Lasso, cette démarche n'est pas raisonnable car on retombe alors dans les écueils de l'estimation par MCO en présence d'un trop grand nombre de variables : la variance d'estimation sera élevée. De plus, bien que la démarche puisse sembler naturelle, il n'y a (à ma connaissance) aucune garantie théorique permettant d'appuyer cette procédure et les études par simulation (voir TP) conduisent à des résultats peu convaincants.

Mise en oeuvre sous R :

Une procédure Lasso avec `glmnet` ou `lars`, suivie d'une estimation par MCO standard avec `lm`.

4.4.3 Adaptive Lasso

Cet estimateur est construit comme le Lasso, mais en permettant une pénalité différente pour chaque coefficient, cette dernière étant d'autant plus importante que le coefficient semble proche de 0. Cet a priori sur la valeur des coefficients est donné par un estimateur préliminaire $\hat{\beta}$. L'estimateur Lasso adaptatif est précisément défini comme solution de

$$\hat{\beta}_{\mathrm{AL}} = \mathrm{argmin}_{\beta\in\mathbb{R}^p} \|Y - X\beta\|^2 + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_j|}.$$

La pénalité propre au coefficient β_j est donc $\lambda/|\hat{\beta}_j|$. Elle est d'autant plus forte que $|\hat{\beta}_j|$ est faible, ce qui conduira la procédure Lasso à annuler plus facilement ce coefficient.

Pour l'estimateur préliminaire $\hat{\beta}$, les choix usuels sont l'estimateur par MCO s'il est calculable, l'estimateur Ridge ou l'estimateur Lasso classique.

L'intérêt de cette méthode est sa capacité à mieux sélectionner les variables pertinentes que les méthodes précédentes.

La solution peut s'obtenir à l'aide d'un algorithme de type LARS ou d'une descente par coordonnées.

Mise en oeuvre sous R :

Une fois le vecteur des estimations préliminaires $\hat{\beta}$ obtenu, il suffit de préciser `penalty.factor=1/abs(`$\hat{\beta}$`)` en option des fonctions `glmnet` et `cv.glmnet`.

4.4.4 Logistic Lasso

Pour rappel, la régression logistique consiste à modéliser une variable binaire Y en fonction des variables explicatives X de la façon suivante. En supposant que Y prend les valeurs 0 et 1, le modèle s'écrit pour l'individu i

$$\mathbb{P}(y_i = 1 |\, X_{\bullet i} = x_i') = \frac{e^{x_i'\beta}}{1 + e^{x_i'\beta}}.$$

En notant $\pi(x_i)$ la probabilité précédente, on en déduit que les Y_i sont indépendants et suivent une loi de Bernoulli $\mathcal{B}(\pi(x_i))$, d'où la vraisemblance

$$V(\beta; Y) = \prod_{i=1}^{n} \pi(x_i)^{y_i}(1 - \pi(x_i))^{1-y_i}.$$

En utilisant la forme de $\pi(x_i)$, on obtient la log-vraisemblance

$$L(\beta; Y) = \sum_{i=1}^{n} y_i x_i'\beta - \ln(1 + e^{x_i'\beta}). \tag{4.16}$$

L'estimateur du maximum de vraisemblance $\hat{\beta}$ maximise la log-vraisemblance, ou de façon équivalente minimise son opposé

$$\hat{\beta} = \text{argmin}_{\beta \in \mathbb{R}^p} - L(\beta; Y).$$

La solution n'est pas explicite et s'obtient par un algorithme d'optimisation (dans la fonction `glm` sous R, il s'agit par défaut de l'algorithme IRLS, pour "Iteratively Reweighted Least Squares").

La régression logistique Lasso suit le même principe que la régression Lasso : dans le problème d'optimisation définissant $\hat{\beta}$, on ajoute une contrainte sur la norme ℓ_1 des coefficients.

$$\hat{\beta}_{Logit-Lasso} = \text{argmin}_{\beta\in\mathbb{R}^p} - L(\beta; Y) \quad \text{sous la contrainte } \sum_{j=1}^{p} |\beta_j| \leq \kappa,$$

où L est donnée par (4.16). De façon équivalente

$$\hat{\beta}_{Logit-Lasso} = \text{argmin}_{\beta\in\mathbb{R}^p} - L(\beta; Y) + \lambda \sum_{j=1}^{p} |\beta_j|,$$

où λ est le paramètre de régularisation (lié à κ) à choisir par validation croisée.

La résolution du problème d'optimisation précédent peut se faire par un algorithme de descente par coordonnées.

Mise en oeuvre sous R :
Fonction `glmnet` avec l'option `family="binomial"`.

4.4.5 Group-Lasso

Dans certains modèles, il peut être préférable de sélectionner les variables par groupe. Par exemple, une variable qualitative à k modalités est encodée dans un modèle de régression linéaire (ou logistique) par $k-1$ variables indicatrices. Si l'on souhaite supprimer la variable qualitative du modèle, cela revient à supprimer le groupe des $k-1$ variables indicatrices correspondantes.

En supposant qu'on a formé G groupes de variables, chacun contenant k_g variables, la régression Group-Lasso pénalise les groupes de la manière suivante

$$\hat{\beta}_{Group-Lasso} = \text{argmin}_{\beta\in\mathbb{R}^p} \|Y - X\beta\|^2 + \lambda \sum_{g=1}^{G} k_g \|\beta_g\|$$

où β_g désigne le sous-vecteur de β associé aux k_g coefficients du groupe g et $\|\beta_g\|$ est sa norme euclidienne. La forme de la pénalité conduit ainsi à une solution qui annule certaines normes $\|\beta_g\|$ et donc tous les coefficients du groupe g.

Dans le cas particulier où tous les groupes sont de taille $k_g = 1$, on retrouve le Lasso classique car alors $\|\beta_g\| = \sqrt{\beta_g^2} = |\beta_g|$.

La même idée se généralise à la régression logistique Lasso.

Mise en oeuvre sous R :

Fonction `gglasso` de la librairie du même nom. Elle permet une régression linéaire Group-Lasso par défaut, mais également une régression logistique Group-Lasso avec l'option `loss = "logit"`. Les groupes de variables sont spécifiés par l'option `group` sous la forme d'un vecteur d'indices. Par exemple `group=c(1,1,2,3,2)` signifie que les deux premières variables forment un groupe, que la seconde et la dernière variable forment un autre groupe, et que la quatrième variable est seule.

Chapter 5

Tests multiples

Lors d'une analyse statistique, il n'est pas rare d'effectuer de nombreux tests statistiques.

Exemple 1: En génomique. On souhaite tester si l'expression d'un gêne est différent entre deux conditions expérimentales (en faisant typiquement un test d'égalité de moyennes). Cette procédure est effectuée pour tous les gênes mesurés, ce qui peut représenter des dizaines de milliers de tests. Historiquement la génomique est le domaine dans lequel s'est le plus développé la théorie des tests multiples.

Exemple 2: En régression linéaire lorsque p est grand. On est amené dans ce contexte à tester la significativité de chaque variable, ce qui représente p tests.

Exemple 3: Corrélation fortuite. En présence de p variables, on peut tester la significativité des corrélations linéaires entre chaque variable. Cela représente $p(p-1)/2$ corrélations à tester. Certaines risquent à tort d'être considérées comme étant significatives, ce sont des corrélations fortuites.

Lorsqu'on effectue de nombreux tests statistiques, chacun étant associé à un risque de première espèce α, on peut s'attendre à détecter de nombreux faux positifs. Ce chapitre expose en détail ce problème et les solutions courantes pour contrôler ce risque.

5.1 Présentation du problème et notations

On suppose qu'on a m tests à effectuer, chacun étant associé à une hypothèse nulle $H_{0,i}$ et une hypothèse alternative $H_{1,i}$ pour $i = 1, \dots, m$. On note I_0 l'ensemble des indices i pour lesquels $H_{0,i}$ est vrai et on note m_0 son cardinal. Evidemment I_0 et m_0 ne sont pas connus en pratique.

Dans la plupart des cas, les hypothèses sont de la forme

$$\begin{aligned} & H_{0,i} : \mu_i = 0 \quad \text{contre} \quad H_{1,i} : \mu_i > 0, \\ \text{ou} \quad & H_{0,i} : \mu_i = 0 \quad \text{contre} \quad H_{1,i} : \mu_i \neq 0, \end{aligned}$$

pour certains paramètres inconnus $\mu_1, \dots, \mu_m$. Par exemple μ_i représente la différence d'expression d'un gêne entre deux conditions expérimentales, ou μ_i représente un paramètre dans une régression linéaire, ou μ_i représente une corrélation linéaire entre deux variables. Cela motive les terminologies suivantes :

- Les positifs P sont les indices i pour lesquels on a rejeté l'hypothèse nulle $H_{0,i}$
- Les faux-positifs FP sont les indices i positifs à tort (autrement dit $H_{0,i}$ est rejeté alors que $i \in I_0$).
- Les vrais-positifs VP sont les indices i positifs à raison (autrement dit $H_{0,i}$ est rejeté et $i \notin I_0$).

On a évidemment $P = FP + VP$.

Lors de la mise en oeuvre des m tests, on obtient m p-values $\hat{p}_i$. On fera l'hypothèse non restrictive suivante :

Hypothèse : Pour tout $i \in I_0$, $\hat{p}_i \sim \mathcal{U}([0,1])$.

Exemples : si la statistique de test admet une loi continue de fonction de répartition F et la région critique est de la forme

- $RC_\alpha = \{T > F^{-1}(1-\alpha)\}$, alors $\hat{p} = 1 - F(T) \sim \mathcal{U}([0,1])$.
- $RC_\alpha = \{T < F^{-1}(\alpha/2)\} \cup \{T > F^{-1}(1-\alpha/2)\}$, alors $\hat{p} = 2\min(F(T), 1 - F(T)) \sim \mathcal{U}([0,1])$.

Preuve des exemples. Pour le premier cas, en notant F^{-1} l'inverse de F :

$$\mathbb{P}(\hat{p} \leq p) = \mathbb{P}(1 - F(T) \leq p) = \mathbb{P}(T \geq F^{-1}(1-p)) = 1 - F(F^{-1}(1-p)) = p.$$

Pour le second cas,

$$\begin{aligned}\mathbb{P}(\hat{p} \leq p) &= \mathbb{P}(\hat{p} \leq p, F(T) \leq 1 - F(T)) + \mathbb{P}(\hat{p} \leq p, F(T) > 1 - F(T)) \\ &= \mathbb{P}\left(T \leq F^{-1}(\frac{p}{2}), T \leq F^{-1}(\frac{1}{2})\right) + \mathbb{P}\left(T \geq F^{-1}(1-\frac{p}{2}), T > F^{-1}(\frac{1}{2})\right).\end{aligned}$$

Comme $p \leq 1$, on sait que $p/2 \leq 1/2$ et $1 - p/2 \geq 1/2$. Puisque F^{-1} est croissante, cela implique $F^{-1}(\frac{p}{2}) \leq F^{-1}(\frac{1}{2})$ et $F^{-1}(1-\frac{p}{2}) \geq F^{-1}(\frac{1}{2})$. D'où

$$\mathbb{P}(\hat{p} \leq p) = \mathbb{P}\left(T \leq F^{-1}(\frac{p}{2})\right) + \mathbb{P}\left(T \geq F^{-1}(1-\frac{p}{2})\right) = \frac{p}{2} + (1 - (1-\frac{p}{2})) = p.$$

□

On souhaite se donner une règle de décision sur les m p-values $\hat{p}_i$. On peut la construire de deux manières

- A partir de quel seuil τ (pour les $\hat{p}_i$) rejette-t-on les hypothèses nulles? Cela aboutit à la règle : si $\hat{p}_i < \tau$, on rejette $H_{0,i}$.
- De façon alternative, si on ordonne les $\hat{p}_i$ de $\hat{p}_{(1)}$ à $\hat{p}_{(m)}$: à partir de quel rang $\hat{k}$ considère-t-on que les p-values $\hat{p}_{(1)}$ à $\hat{p}_{(\hat{k})}$ conduisent à rejeter les hypothèses nulles correspondantes, tandis que pour $\hat{p}_{(\hat{k}+1)}$ à $\hat{p}_{(m)}$ on ne les rejette pas?

Ces deux points de vue diffèrent dans la mesure où τ n'est pas aléatoire alors que $\hat{k}$ oui. Le lien entre ces deux points de vue est illustré dans la figure 5.1.

L'objectif d'une bonne procédure de tests multiples est de trouver une règle de décision comme ci-dessus qui maximise le nombre de VP tout en minimisant les FP.

Si on se fixe un seuil τ indépendant de m, par exemple $\tau = 5\%$, on aura en moyenne

$$\mathbb{E}(FP) = \mathbb{E}\left(\sum_{i=1}^{m} 1_{\{i \in I_0,\ \hat{p}_i < \tau\}}\right) = \sum_{i \in I_0} \mathbb{P}(\hat{p}_i < \tau) = m_0 \tau. \qquad (5.1)$$

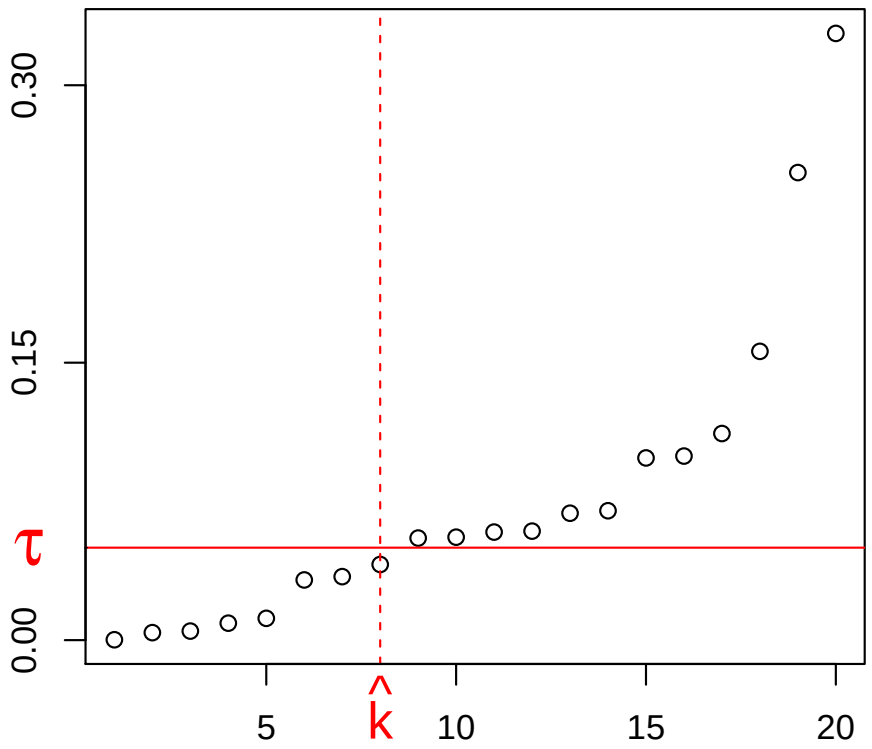

Figure 5.1: Représentation de 20 p-values ordonnées de la plus petite $\hat{p}_{(1)}$ à la plus grande $\hat{p}_{(20)}$. Si on se fixe un seuil $\tau > 0$, on rejette tous les tests ayant une p-value inférieure à τ. De façon équivalente, on rejette les tests associés aux $\hat{k}$ plus petites p-values. Ici $\hat{k} = 8$.

Par exemple, si on fait $m = 1000$ tests qui sont en théorie tous négatifs ($m = m_0$) en utilisant le seuil $\tau = 5\%$, on obtient en moyenne 50 FP. Pour contrôler l'apparition des FP, il faut donc affiner la règle de décision en tenant compte de m.

5.2 Principe des tests multiples : contrôler le $FWER$ ou le FDR

Rappel : lorsqu'on effectue un seul test (H_0 contre H_1), on adopte généralement le principe de Neyman-Pearson. On se donne la probabilité de décider H_1 à tort (faux positif), c'est à dire le risque de première espèce α, et on privilégie, si on a le choix, le test qui, à α donné, maximise la puissance, c'est à dire la probabilité de décider H_1 à raison (vrai positif).

On trouve deux façons principales d'étendre ce principe au cas des tests multiples :

1. par le $FWER$ (Family-Wise Error Rate, taux d'erreur par famille) : on décide de fixer
$$FWER = \mathbb{P}(FP > 0)$$
à α, ou autrement dit $\mathbb{P}(FP = 0)$ à $1-\alpha$, et on privilégie les procédures qui, à α fixé, maximisent le nombre de VP.

2. par le FDR (False Discovery Rate, taux de faux positifs) : on note FDP la proportion de FP parmi P, avec la convention $FDP = 0$ si $P = 0$, c'est à dire $FDP = (FP/P)\,1_{P\geq 1}$. On décide alors de fixer
$$FDR = \mathbb{E}(FDP) = \mathbb{E}((FP/P)\,1_{P\geq 1})$$
à α et on privilégie les procédures qui, à α fixé, maximisent le nombre de VP.

La proposition suivante montre que lorsque $m = 1$, les deux critères coïncident et correspondent au risque de première espèce usuel, tandis que dans le cas général $FDR \leq FWER$. Donc FDR est un critère moins exigeant que $FWER$ dans le sens où une procédure assurant $FWER \leq \alpha$ assure automatiquement $FDR \leq \alpha$, mais le contraire est faux. En particulier on fournit généralement plus de VP avec une procédure assurant $FDR \leq \alpha$ qu'avec la même procédure assurant $FWER \leq \alpha$.

Proposition 5.2.1. *On a les propriétés suivantes :*

(i) En présence d'un seul test ($m = 1$), $FWER = FDR$ correspond au risque usuel de première espèce.

(ii) Dans le cas général ($m \geq 1$), on a $FDR \leq FWER$.

Proof. (i) Si $m = 1$, $FWER = \mathbb{P}(FP = 1)$ correspond exactement à la probabilité de rejeter H_0 à tort, c'est à dire au risque de première espèce usuel. Par ailleurs, toujours si $m = 1$, FDP vaut 1 lorsqu'il y a un faux positif (situation où $FP = P = 1$) et 0 sinon, donc son espérance FDR vaut également $\mathbb{P}(FP = 1)$.

(ii) Dans le cas général $FP\,1_{P\geq 1} \leq P\,1_{FP\geq 1}$ (ce qui se vérifie en considérant les cas possibles $P = 0$ ou $P \geq 1$, et $FP = 0$ ou $FP \geq 1$), d'où $(FP/P)1_{P\geq 1} \leq 1_{FP\geq 1}$ ce qui implique le résultat en passant à l'espérance. □

5.2.1 Procédure de Bonferroni

La procédure de Bonferroni est très simple : pour assurer $FWER \leq \alpha$, il suffit de fixer le seuil $\tau = \alpha/m$.

La force de cette procédure est qu'elle garantit un contrôle du $FWER$ quelle que soit la dépendance entre les p-values des m tests. Son gros défaut est qu'elle est beaucoup trop conservative si m est grand, dans le sens où le seuil τ devient tellement petit qu'il y a très peu de positifs détectés, donc forcément très peu de FP (ce qui est un bon point) mais aussi très peu de VP.

Proposition 5.2.2. *La procédure de Bonferroni assure $FWER \leq \alpha$.*

Proof. On observe un FP si pour un test i, $\hat{p}_i \leq \alpha/m$ alors que $H_{0,i}$ est vrai, ce qui signifie $i \in I_0$. Donc

$$FWER = \mathbb{P}(FP > 0) = \mathbb{P}(\exists i \in I_0,\ \hat{p}_i \leq \alpha/m) \leq \sum_{i\in I_0} \mathbb{P}(\hat{p}_i \leq \alpha/m)$$

car la probabilité d'une union est inférieure à la somme des probabilités. Puisque pour $i \in I_0$, $\hat{p}_i \sim \mathcal{U}([0,1])$, ces dernières probabilités valent α/m et on aboutit à $FWER \leq m_0\alpha/m \leq \alpha$. □

5.2.2 Procédure de Benjamini-Hochberg

La procédure de BH contrôle le FDR. Il s'agit de la procédure de tests multiples la plus utilisée en pratique.

On rappelle que le choix d'un seuil τ pour les p-values conduit à $P = \hat{k}$ positifs, qui sont associés aux $\hat{k}$ plus petites p-values avec $\hat{p}_{(\hat{k})} \leq \tau < \hat{p}_{(\hat{k}+1)}$, voir la figure 5.1.

L'heuristique de la procédure de BH est la suivante : on sait que pour un seuil τ donné $\mathbb{E}(FP) = m_0\tau \leq m\tau$, voir (5.1). On peut donc s'attendre à ce qu'en moyenne $FP \leq m\tau$. Tant que τ reste dans l'intervalle $[\hat{p}_{(\hat{k})}, \hat{p}_{(\hat{k}+1)}[$, les conclusions des tests restent identiques et conduisent à $\hat{k}$ positifs. Donc pour $\tau = \hat{p}_{(\hat{k})}$, on peut s'attendre à ce qu'en moyenne $FP \leq m\hat{p}_{(\hat{k})}$, soit $FP/P \leq m\hat{p}_{(\hat{k})}/\hat{k}$, autrement dit $FDR \leq m\hat{p}_{(\hat{k})}/\hat{k}$. Evidemment cette inégalité n'est pas juste car l'aléa et l'espérance n'ont pas été manipulés convenablement. Néanmoins, si on se base sur cette inégalité, on aura $FDR \leq \alpha$ dès que $m\hat{p}_{(\hat{k})}/\hat{k} \leq \alpha$. Il y a plusieurs choix possibles de $\hat{k}$ conduisant à cette condition. Parmi ces possibilités, on souhaite maximiser le nombre de positifs $\hat{k}$, donc on choisit pour $\hat{k}$ le plus grand k vérifiant $\hat{p}_{(k)} \leq \alpha k/m$.

Suivant cette heuristique, la région de rejet de BH correspond donc à toutes les p-values $\hat{p}_i$ inférieures à $\hat{p}_{(\hat{k})}$ où $\hat{k}$ vérifie

$$\hat{k} = \max\{k, \hat{p}_{(k)} \leq \alpha k/m\}. \tag{5.2}$$

Comme l'illustre la figure 5.2, l'interprétation graphique est simple : $\hat{p}_{(\hat{k})}$ est la dernière p-value sous la droite de pente α/m, dans le nuage de points des p-values ordonnées. En comparaison, la procédure de Bonferroni (pour le même seuil α) rejette nécessairement moins d'hypothèses.

La proposition suivante montre que le FDR associé à la procédure BH est bien inférieure à α lorsque les p-values sont indépendantes entre elles. Si les p-values ne sont pas indépendantes entre elles, le résultat n'est pas vrai en général mais il le reste sous l'hypothèse que l'inégalité (5.4) présentée dans le lemme ci-dessous reste vraie. Cela est en particulier vérifié en présence de certaines dépendances positives entre les p-values.

Proposition 5.2.3. *Si les p-values sont indépendantes entre elles, la procédure de Benjamini-Hochberg assure $FDR \leq \alpha$.*

Proof. Par définition il y a $\hat{k}$ positifs et l'hypothèse $H_{0,i}$ est rejetée si $\hat{p}_i$ est

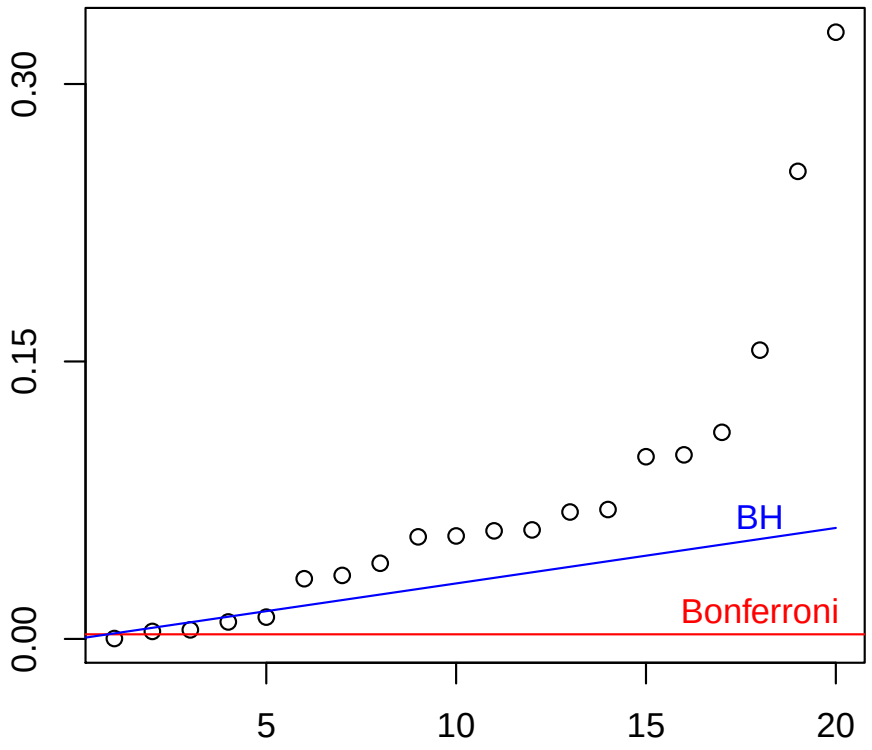

Figure 5.2: Représentation de 20 p-values ordonnées de la plus petite $\hat{p}_{(1)}$ à la plus grande $\hat{p}_{(20)}$. La région de rejet de Bonferroni correspond aux p-values inférieures à α/m (ligne rouge), tandis que la région de rejet de BH correspond aux $\hat{k}$ plus petites p-values, où $\hat{p}_{(\hat{k})}$ est la dernière p-value sous la droite de pente α/m (ligne bleu). Ici pour $m = 20$ et $\alpha = 0.05$, on a $\hat{k} = 5$ tandis que Bonferroni ne rejette qu'un seul test.

inférieure à $\hat{p}_{(\hat{k})}$. Il s'agit d'un faux positif si de plus $i \in I_0$. Ainsi

$$FDR = \mathbb{E}\left(\frac{FP}{\hat{k}} 1_{\hat{k}\geq 1}\right) = \mathbb{E}\left(\frac{\sum_{i\in I_0} 1_{\hat{p}_i \leq \hat{p}_{(\hat{k})}}}{\hat{k}} 1_{\hat{k}\geq 1}\right) \leq \sum_{i\in I_0} \mathbb{E}\left(\frac{1_{\hat{p}_i \leq \alpha\hat{k}/m}}{\hat{k}} 1_{\hat{k}\geq 1}\right)$$

où l'on a utilisé pour la dernière inégalité le fait que $\hat{p}_{(\hat{k})} \leq \alpha\hat{k}/m$ pour la procédure de BH. En décomposant sur toutes les valeurs possibles de $\hat{k}$, on obtient

$$\begin{aligned}
FDR &\leq \sum_{i\in I_0} \mathbb{E}\left(\frac{1_{\hat{p}_i \leq \alpha\hat{k}/m}}{\hat{k}} 1_{\hat{k}\geq 1} \sum_{k=0}^{m} 1_{\hat{k}=k}\right) \\
&= \sum_{i\in I_0} \sum_{k=1}^{m} \mathbb{E}\left(\frac{1_{\hat{p}_i \leq \alpha k/m}}{k} 1_{\hat{k}=k}\right) \\
&= \sum_{i\in I_0} \sum_{k=1}^{m} \frac{1}{k} \mathbb{P}\left(\hat{p}_i \leq \alpha k/m,\ \hat{k}=k\right) \\
&= \sum_{i\in I_0} \sum_{k=1}^{m} \frac{1}{k} \mathbb{P}\left(\hat{k}=k | \hat{p}_i \leq \alpha k/m\right) \mathbb{P}\left(\hat{p}_i \leq \alpha k/m\right) \\
&= \frac{\alpha}{m} \sum_{i\in I_0} \sum_{k=1}^{m} \mathbb{P}\left(\hat{k}=k | \hat{p}_i \leq \alpha k/m\right) \qquad (5.3)
\end{aligned}$$

en utilisant le fait que $\hat{p}_i \sim \mathcal{U}([0,1])$ lorsque $i \in I_0$. Le lemme suivant, démontré plus bas, est la clé de la preuve.

Lemme 5.2.4. *Si les p-values $\hat{p}_1, \ldots, \hat{p}_m$ sont indépendantes alors pour tout $i = 1, \ldots, m$,*

$$\mathbb{P}\left(\hat{k} \leq k | \hat{p}_i \leq \alpha k/m\right) \leq \mathbb{P}\left(\hat{k} \leq k | \hat{p}_i \leq \alpha(k+1)/m\right). \qquad (5.4)$$

En notant $a_k = \mathbb{P}\left(\hat{k} \leq k | \hat{p}_i \leq \alpha(k+1)/m\right)$, on a pour tout $k \geq 1$,

$$\begin{aligned}
\mathbb{P}\left(\hat{k}=k | \hat{p}_i \leq \alpha k/m\right) &= \mathbb{P}\left(\hat{k} \leq k | \hat{p}_i \leq \alpha k/m\right) - \mathbb{P}\left(\hat{k}=k-1 | \hat{p}_i \leq \alpha k/m\right) \\
&= \mathbb{P}\left(\hat{k} \leq k | \hat{p}_i \leq \alpha k/m\right) - a_{k-1} \\
&\leq a_k - a_{k-1}
\end{aligned}$$

en utilisant (5.4). En injectant cette inégalité dans (5.3), on obtient

$$FDR \leq \frac{\alpha}{m} \sum_{i \in I_0} \sum_{k=1}^{m} (a_k - a_{k-1}) = \frac{\alpha}{m} \sum_{i \in I_0} (a_m - a_0).$$

On remarque que $a_0 = 0$ car si $\hat{p}_i \leq \alpha/m$, cela implique que $\hat{p}_{(1)} \leq \alpha/m$ et donc d'après (5.2) que $\hat{k} \geq 1$, autrement dit $a_0 = \mathbb{P}\left(\hat{k} \leq 0 | \hat{p}_i \leq \alpha/m\right) = 0$. Ainsi, en utilisant de plus le fait que $a_m \leq 1$ et que $m_0 \leq m$, on conclut que

$$FDR \leq \frac{\alpha}{m} \sum_{i \in I_0} a_m \leq \frac{\alpha}{m} m_0 \leq \alpha.$$

□

Preuve du Lemme 5.2.4. Sans perte de généralité on suppose que $i = 1$ pour la preuve. On rappelle que par définition $\hat{k}$ dépend de toutes les p-values et vaut $\hat{k} = k(\hat{p}_1, \ldots, \hat{p}_m) = \max\{j, \hat{p}_{(j)} \leq \alpha j/m\}$. On a

$$\mathbb{P}\Big(k(\hat{p}_1, \ldots, \hat{p}_m) \leq k \,|\, \hat{p}_1 \leq \alpha k/m\Big) = \frac{\mathbb{P}\Big(k(\hat{p}_1, \ldots, \hat{p}_m) \leq k, \hat{p}_1 \leq \alpha k/m\Big)}{\mathbb{P}(\hat{p}_1 \leq \alpha k/m)}.$$

En notant f_i la densité de $\hat{p}_i$, le numérateur vaut par indépendance des $\hat{p}_i$

$$\begin{aligned}
&\int 1_{\{k(x_1,\ldots,x_m) \leq k, x_1 \leq \alpha k/m\}} f_1(x_1) \ldots f_m(x_m) dx_1 \ldots dx_m \\
&= \int \left(\int 1_{\{k(x_1,\ldots,x_m) \leq k, x_1 \leq \alpha k/m\}} f_1(x_1) dx_1 \right) f_2(x_2) \ldots f_m(x_m) dx_2 \ldots dx_m \\
&= \int \mathbb{P}\Big(k(\hat{p}_1, x_2, \ldots, x_m) \leq k, \hat{p}_1 \leq \alpha k/m\Big) f_2(x_2) \ldots f_m(x_m) dx_2 \ldots dx_m
\end{aligned}$$

Ainsi en divisant par $\mathbb{P}(\hat{p}_1 \leq \alpha k/m)$ on obtient

$$\begin{aligned}
&\mathbb{P}\Big(k(\hat{p}_1, \ldots, \hat{p}_m) \leq k \,|\, \hat{p}_1 \leq \alpha k/m\Big) \\
&= \int \mathbb{P}\Big(k(\hat{p}_1, x_2, \ldots, x_m) \leq k \,|\, \hat{p}_1 \leq \alpha k/m\Big) f_2(x_2) \ldots f_m(x_m) dx_2 \ldots dx_m.
\end{aligned}$$

Pour montrer l'inégalité du lemme, il suffit donc de prouver que pour tout $x_2, \ldots, x_m$,

$$\begin{aligned}
&\mathbb{P}\Big(k(\hat{p}_1, x_2, \ldots, x_m) \leq k \,|\, \hat{p}_1 \leq \alpha k/m\Big) \\
&\qquad \leq \mathbb{P}\Big(k(\hat{p}_1, x_2, \ldots, x_m) \leq k \,|\, \hat{p}_1 \leq \alpha(k+1)/m\Big).
\end{aligned}$$

En notant g la fonction $p \mapsto k(p, x_2, \dots, x_m)$ définie sur $[0,1]$, on doit donc prouver que

$$\mathbb{P}\Big(g(\hat{p}_1) \le k \,|\, \hat{p}_1 \le \alpha k/m\Big) \le \mathbb{P}\Big(g(\hat{p}_1) \le k \,|\, \hat{p}_1 \le \alpha(k+1)/m\Big).$$

D'après la définition de $\hat{k}$, la fonction g ne peut valoir que deux valeurs, soit $g(0)$ soit $g(1)$, avec un éventuel saut en une valeur p_0 de $[0,1]$ (p_0 dépendant de $x_2, \dots, x_m$) avec la convention que $p_0 = 0$ si $g(0) = g(1)$. En effet, augmenter une p-value ne peut que diminuer l'indice $\hat{k}$ retenu pour le rejet des hypothèses. Ce changement s'opère au plus un fois auquel cas $g(1) = g(0)-1$, sinon $g(0) = g(1)$. La fonction est par ailleurs continue à gauche.

On note $g^{-1}(k) = \sup\{p,\, g(p) \ge k\}$ avec la convention que $g^{-1}(k) = -\infty$ si $k > g(0)$. Il s'agit de l'inverse à droite de g dans le sens où $g(g^{-1}(k)) = k$ pour tout $k \in Im(g)$. Il est facile de vérifier que $\{p, g(p) < k\} = \{p > g^{-1}(k)\}$. En effet

- Si $k > g(0)$ alors $g^{-1}(k) = -\infty$ et $\{g(p) < k\} = \{p \ge 0\} = \{p > g^{-1}(k)\}$.
- Si $k = g(0)$ alors $g^{-1}(k) = p_0$ et $\{g(p) < k\} = \{p > p_0\} = \{p > g^{-1}(k)\}$.
- Si $k \le g(1)$, alors $g^{-1}(k) = 1$ et $\{g(p) < k\} = \emptyset = \{p > g^{-1}(k)\}$.

Ainsi, quel que soit k,

$$\begin{aligned}
\mathbb{P}\Big(g(\hat{p}_1) \le k \,|\, \hat{p}_1 \le \alpha k/m\Big) &= \mathbb{P}\Big(g(\hat{p}_1) < k+1 \,|\, \hat{p}_1 \le \alpha k/m\Big) \\
&= \mathbb{P}\Big(\hat{p}_1 > g^{-1}(k+1) \,|\, \hat{p}_1 \le \alpha k/m\Big) \\
&= 1 - \mathbb{P}\Big(\hat{p}_1 \le g^{-1}(k+1) \,|\, \hat{p}_1 \le \alpha k/m\Big) \\
&= \begin{cases} 1 - \frac{\mathbb{P}(\hat{p}_1 \le g^{-1}(k+1))}{\mathbb{P}(\hat{p}_1 \le \alpha k/m)} & \text{si } g^{-1}(k+1) \le \alpha k/m \\ 0 \quad \text{sinon.} \end{cases}
\end{aligned}$$

Puisque $\mathbb{P}(\hat{p}_1 \leq \alpha k/m) \leq \mathbb{P}(\hat{p}_1 \leq \alpha(k+1)/m)$

$$\begin{aligned}
\mathbb{P}\Big(g(\hat{p}_1) \leq k \,|\, \hat{p}_1 \leq \alpha k/m\Big) &\leq \begin{cases} 1 - \frac{\mathbb{P}(\hat{p}_1 \leq g^{-1}(k+1))}{\mathbb{P}(\hat{p}_1 \leq \alpha(k+1)/m)} & \text{si } g^{-1}(k+1) \leq \alpha k/m \\ 0 \quad \text{sinon,} \end{cases} \\
&= \begin{cases} 1 - \frac{\mathbb{P}(\hat{p}_1 \leq g^{-1}(k+1))}{\mathbb{P}(\hat{p}_1 \leq \alpha(k+1)/m)} & \text{si } g^{-1}(k+1) \leq \alpha k/m \\ 0 \quad \text{si } \alpha k/m \ < g^{-1}(k+1) \leq \alpha(k+1)/m \\ 0 \quad \text{sinon,} \end{cases} \\
&\leq \begin{cases} 1 - \frac{\mathbb{P}(\hat{p}_1 \leq g^{-1}(k+1))}{\mathbb{P}(\hat{p}_1 \leq \alpha(k+1)/m)} & \text{si } g^{-1}(k+1) \leq \alpha(k+1)/m \\ 0 \quad \text{sinon,} \end{cases} \\
&= \mathbb{P}\Big(g(\hat{p}_1) \leq k \,|\, \hat{p}_1 \leq \alpha(k+1)/m\Big).
\end{aligned}$$

□

5.2.3 p-values ajustées

Lors de la réalisation d'un test statistique ($m = 1$), on calcule généralement la p-value $\hat{p}$ car cette démarche nous évite de choisir le niveau du test α au préalable, contrairement à la construction d'une région critique. Le choix de α peut se faire a posteriori grâce à la règle : si $\hat{p} < \alpha$, on rejette l'hypothèse nulle au niveau α. On peut ainsi observer directement pour quels choix de α la décision du test se trouve modifiée.

Cet avantage est perdu avec les procédures de Bonferroni et de BH présentées ci-dessus, car même si elles s'appuient sur les m p-values initiales $\hat{p}_1, \ldots, \hat{p}_m$, elles nécessitent de se fixer un seuil α au préalable (correspondant respectivement au contrôle du $FWER$ ou du FDR). On peut néanmoins modifier les p-values initiales en $\tilde{p}_1, \ldots, \tilde{p}_m$ en s'appuyant sur la correction de Bonferroni (respectivement de BH) pour aboutir à la simple règle de décision : si $\tilde{p}_i \leq \alpha$, alors on rejette le test i pour la procédure de Bonferroni associée à $FWER \leq \alpha$ (respectivement on rejette le test i pour la procédure de BH associée à $FDR \leq \alpha$).

Proposition 5.2.5.

- *Les p-values ajustées au sens de Bonferroni valent, pour $i = 1, \ldots, m$,*

$$\tilde{p}_i = m\hat{p}_i.$$

Le test i est rejeté par la procédure de Bonferroni associée à $FWER \leq \alpha$ ssi $\tilde{p}_i \leq \alpha$.

- *Les p-values ajustées au sens de BH valent, pour $i = 1, \dots, m$,*

$$\tilde{p}_{(i)} = \min_{i \leq j \leq m} \left\{ \frac{m}{j} \hat{p}_{(j)} \right\}.$$

La présence des parenthèses en indice indique qu'il s'agit de l'ajustement de la i-ème plus petite p-value $\hat{p}_{(i)}$. Pour obtenir $\tilde{p}_i$, il suffit de remettre les p-values dans l'ordre initial.

Le test i est rejeté par la procédure de BH associée à $FDR \leq \alpha$ ssi $\tilde{p}_i \leq \alpha$.

Proof. La procédure de Bonferroni associée à $FWER \leq \alpha$ rejette le test i ssi $\hat{p}_i \leq \alpha/m \Leftrightarrow m\hat{p}_i \leq \alpha \Leftrightarrow \tilde{p}_i \leq \alpha$.

Pour BH, on montre l'équivalence (sans perte de généralité) sur les p-values ordonnées $\hat{p}_{(1)}, \dots, \hat{p}_{(m)}$. La procédure de BH associée à $FDR \leq \alpha$ rejette le test associé à $\hat{p}_{(i)}$ ssi $i \leq \hat{k}$ où

$$\hat{k} = \operatorname*{argmax}_{1 \leq j \leq m} \left\{ \hat{p}_{(j)} \leq \alpha \frac{j}{m} \right\} = \operatorname*{argmax}_{1 \leq j \leq m} \left\{ \frac{m}{j} \hat{p}_{(j)} \leq \alpha \right\}.$$

Par définition $\frac{m}{\hat{k}} \hat{p}_{(\hat{k})} \leq \alpha$ donc $\tilde{p}_{(\hat{k})} \leq \frac{m}{\hat{k}} \hat{p}_{(\hat{k})} \leq \alpha$. Ainsi, si la procédure de BH rejette le test associé à $\hat{p}_{(i)}$, alors $\tilde{p}_{(i)} \leq \tilde{p}_{(\hat{k})} \leq \alpha$.

Réciproquement, si $\tilde{p}_{(i)} \leq \alpha$, alors il existe $j \geq i$ tel que $\frac{m}{j} \hat{p}_{(j)} \leq \alpha$, ce qui montre que $\hat{k} \geq i$ et donc que le test associé à $\hat{p}_{(i)}$ est rejeté par la procédure de BH. □

Mise en oeuvre sous R : Etant donné le vecteur des p-values initiales $\hat{p}_1, \dots, \hat{p}_m$, la fonction `p.adjust` retourne les p-values ajustées $\tilde{p}_1, \dots, \tilde{p}_m$ au sens de BH avec l'option `method="BH"`.

D'autres corrections de p-values sont proposées dans cette fonction. Soit elles correspondent à des méthodes contrôlant la FEWR comme la méthode de Bonferroni (`method="bonferroni"`, `method="holm"`, `method="hochberg"`, `method="hommel"`), soit elles correspondent à des méthodes contrôlant la FDR comme la méthode de BH (`method="BH"`, `method="BY"`).