ORSAY
N^o d'ordre :

UNIVERSITÉ PARIS-SUD
FACULTÉ DES SCIENCES D'ORSAY

THÈSE

présentée pour obtenir

LE GRADE DE DOCTEUR EN SCIENCES
DE l'UNIVERSITÉ PARIS XI
Spécialité : Mathématiques

par

Tristan MARY-HUARD

RÉDUCTION DE LA DIMENSION ET SÉLECTION DE MODÈLES EN CLASSIFICATION SUPERVISÉE

Soutenue le 6 juillet 2006 devant la Commission d'examen composée de :

M. Gérard BIAU	Rapporteur
M. Olivier BOUSQUET	Examinateur
M. Gabor LUGOSI	Rapporteur
M. Pascal MASSART	Examinateur
M. Jean-Michel POGGI	Examinateur
M. Stéphane ROBIN	Directeur de thèse
M. Jean-Daniel ZUCKER	Examinateur

Remerciements

Je remercie Gérard Biau et Gabor Lugosi, les rapporteurs de cette thèse, ainsi qu'Olivier Bousquet, Pascal Massart, Jean-Michel Poggi et Jean-Daniel Zucker, membres du jury, de m'avoir fait l'honneur d'être les examinateurs de cette thèse.

Je tiens à exprimer ma profonde reconnaissance envers Stéphane Robin, mon directeur de thèse, et Jean-Jacques Daudin, directeur du département de mathématiques de l'INA-PG, pour leur accueil. J'ai eu la chance de faire auprès d'eux mes premiers pas dans la recherche et l'enseignement, je ne pouvais avoir de meilleurs précepteurs. Je souhaite à tout futur doctorant de trouver dans son laboratoire d'accueil les conditions de travail, mais aussi les conditions de vie que j'ai trouvées ici grâce à eux.

Je remercie Avner et la Bar-Hen Airlines Company, pour nos multiples collaborations nationales, européennes et bretonnes, ainsi que Colette Vuillet, qui m'a tant aidé pour les enseignements. Je remercie aussi mes deux marraines de l'INA-PG Marie-Laure Martin et Emilie Lebarbier pour leur coaching psychologique impeccable tout au long de ces années, Franck Picard et Julie Aubert pour leur soutien (et pour leur patiente envers l'informaticien aux pieds plats que je suis !), ainsi que l'ensemble des membres du département OMIP de l'INA-PG.

Puis vient le bizarre. Car il me faut ici saluer toute une kyrielle de nains teigneux, toute une galerie de freaks, une association déplorable de trompettes, de pinpins, de fissipèdes aquatiques non réglementaires, de godelureaux cuits dans leur jus, de tubes digestifs insondables et de nihilistes. Un vrai biotope en devenir, d'ailleurs : nouveaux pères ou nouvelles mères (pas plus de deux par famille), nouveaux maris ou épouses, nouveaux entrepreneurs ou cadres supérieurs embourgeoisés, nouvelles stars et néo-ruraux de Nancy, d'Angoulême ou d'ailleurs, mais tous vieux amis. J'leur dis merci, et pLus encore.

Enfin, merci à ma famille.

Table des matières

Introduction

La statistique appliquée a pour objet de développer des méthodes rigoureuses pour la description, le traitement et l'analyse des données. Bien que ces objectifs soient restés les mêmes au cours du temps, la pratique de la statistique a beaucoup évolué ces dernières années pour s'adapter aux nouvelles demandes des expérimentateurs. La nature des données à traiter a profondément changé, ainsi que la quantité de données disponible. En particulier, depuis la fin des années 90 sont apparues différentes technologies susceptibles de produire une grande quantité de descripteurs pour une même observation. On peut considérer l'exemple des puces à ADN. Cette technologie permet de mesurer l'expression de plusieurs dizaines de milliers de gènes simultanément chez un individu. On récupère alors une quantité d'information conséquente, mais cette information ne peut être collectée que sur un petit nombre d'individus, car chaque expérience de puce à ADN coûte cher. Le statisticien doit alors travailler avec un échantillon de taille n très petite comparée à la quantité d'information disponible, c'est-à-dire au nombre de variables p (ici les gènes). Cette particularité sera désignée dans la suite comme celle des données de grande dimension, pour lesquelles $n << p$. Notons que l'exemple des biopuces n'est pas unique : les images haute-définition ou les analyses de courbes sont d'autres exemples de données de grande dimension.

Nous nous intéressons dans ce manuscrit au problème de la classification supervisée des données de grande dimension. La classification supervisée a pour objectif de prédire l'appartenance d'un individu à un groupe, en fonction de l'information dont on dispose sur cet individu. L'appartenance au groupe est résumée par le label, qui prend un nombre K fini de valeurs différentes, correspondant aux K groupes. Dans la suite, nous nous intéresserons au cas où $K = 2$. L'information disponible est généralement résumée par p variables. Dans le cas des expériences de puces à ADN, la classification supervisée peut être utilisée pour formaliser le problème du diagnostic. On cherche à déterminer si un individu est sain (groupe 1) ou malade (groupe 2) à partir de l'expression de l'ensemble de ses p gènes. La méthode consiste à construire un classificateur ou règle de classification , c'est-à-dire une fonction qui prenne en argument l'information pour fournir une prédiction du label. On souhaite naturellement construire un classificateur performant, dont le taux d'erreur de prédiction soit le plus faible possible. La construction du classificateur se fait à partir d'un échantillon d'entraînement, constitué de n observations pour lesquelles l'information ainsi que le label sont disponibles. L'élaboration du classificateur consiste alors à "apprendre", à partir de l'échantillon d'entraînement, les relations liant l'information au label.

Les difficultés que posent la construction d'un classificateur pour traiter des données de grande dimension nécessitent non seulement l'adaptation des outils classiques de classification supervisée, mais aussi le développement de méthodes nouvelles et efficaces. Depuis quelques années, la communauté de l'apprentissage (*machine learning* en anglais) regroupe l'ensemble des chercheurs intéressés par ces nouvelles méthodes d'analyse. Le travail que nous présentons ici s'inscrit dans cette discipline, aujourd'hui en plein essor.

Comment construit-on un classificateur ? Cette question n'est pas nouvelle : la première approche de ce problème fut proposée par Fisher (1936). Fisher propose une approche paramétrique du problème, qui permet d'élaborer un classificateur performant sous l'hypothèse que dans chacun des deux groupes la distribution des variables soit gaussienne. La méthode d'estimation, appelée analyse discriminante, fait aujourd'hui partie des outils statistiques classiques d'analyse de données. Toutefois, cette méthode n'est valable que lorsque la loi conditionnelle des variables sachant le label est gaussienne, or la distribution des données est généralement inconnue. A la fin des années 70, Vapnik propose une stratégie générale pour la construction d'un classificateur. Cette stratégie, fondée sur la minimisation du taux d'erreur empirique, est appelée stratégie ERM (pour Empirical Risk Minimization). On considère une classe de classificateurs $\mathcal{C}$ dans laquelle nous devons sélectionner un bon classificateur. Par définition, la règle optimale de classification disponible dans la classe est celle qui minimise le taux d'erreur de classement. Cette règle n'est bien sûr pas identifiable, puisque la détermination du taux d'erreur d'une règle de classement nécessite de connaître la loi jointe des données. A défaut, il semble raisonnable de choisir le classificateur minimisant le taux d'erreur empirique, calculé sur les seules données d'entraînement. Vapnik montre que sous certaines conditions, cette stratégie de minimisation du taux d'erreur empirique est pertinente. Contrairement aux travaux de Fisher, ces conditions ne portent plus sur la distribution des données, mais sur la complexité de la classe de classificateurs $\mathcal{C}$ dans laquelle on choisit le classificateur. En particulier, Vapnik montre que quelle que soit la distribution des données, le taux d'erreur du classificateur choisi converge vers le taux d'erreur de la règle optimale de la classe lorsque la complexité de la classe est finie.

La théorie de Vapnik est présentée au chapitre 1. Nous introduisons la définition de la complexité d'une classe de classificateurs, ainsi que les mesures de quantification de cette complexité (le coefficient de hachage et la dimension de Vapnik). L'application de la stratégie ERM nécessite le choix de la classe de classificateurs. Plutôt que de choisir *a priori* une classe $\mathcal{C}$, il est pertinent de se donner une collection de classes $\mathcal{C}_1$,..., $\mathcal{C}_K$ de complexités différentes, et de baser la sélection de l'une de ces classes sur l'information dont on dispose, c'est-à-dire sur les données. Le choix d'un classificateur parmi toutes ces classes ne se fait plus en minimisant le taux d'erreur empirique, mais en minimisant le taux d'erreur empirique pénalisé, où la pénalité prend en compte la complexité de la classe dont l'estimateur fait partie. On se trouve alors dans le cadre de la sélection de modèles par critère pénalisé. La problématique de la sélection de modèles est présentée en détail au chapitre 1 pour la classification supervisée. L'annexe A traite aussi de la sélection de modèles, et présente une approche bayésienne de ce problème.

Lorsque l'on travaille sur des données de grande dimension, il n'est pas toujours pertinent de prédire le label à partir de l'ensemble des variables disponibles. En effet, lorsque le nombre de variables p est grand, il devient difficile de distinguer entre d'une part les relations entre variables et label générales à la population et d'autre part les relations spécifiques à l'échantillon d'entraînement. Un classificateur construit à partir de l'ensemble des variables sera alors surajusté : il sera capable de prédire le label pour les individus de l'échantillon d'entraînement qu'il aura appris "par coeur", mais pas pour une nouvelle observation. Par ailleurs, il se peut que seule une faible proportion des variables disponibles apporte une information réelle sur le label. Seules ses variables sont pertinentes pour la construction d'un classificateur. Il faut alors être capable d'identifier ces variables pertinentes.

La sélection de variables fait l'objet du chapitre 2. Dans ce chapitre, nous présentons tout d'abord les différentes catégories de méthodes de sélection de variables existantes. Nous

étudions ensuite en détail la famille des méthodes basées sur la théorie des couvertures de Markov, du point de vue théorique dans un premier temps, puis du point de vue pratique, en présentant deux algorithmes types de sélection de variables par couverture de Markov. Nous pourrons alors voir les écarts existants entre la théorie des méthodes à couverture de Markov et leur mise en pratique, écarts qui ne sont que peu soulignés dans la littérature sur le sujet.

Ces deux premiers chapitres introductifs montreront clairement la grande diversité des approches proposées dans la communauté de l'apprentissage. Cette diversité vient du fait que beaucoup de chercheurs, venant de champs disciplinaires très différents, ont contribué au développement des méthodes d'apprentissage. De ce fait les résultats présentés dans les articles peuvent être de nature très différentes, du plus empirique où la méthode de classification proposée n'est validée que par les faibles taux d'erreur obtenus sur plusieurs jeux de données, au plus théorique, où la méthode est justifiée par un ensemble de propriétés statistiques démontrées formellement. Ces deux chapitres nous permettront aussi d'exposer l'ensemble des difficultés liées à la mise en oeuvre des méthodes de classification supervisée pour l'analyse des données de grande dimension.

Les trois chapitres suivants regroupent l'ensemble des travaux originaux que nous proposons pour l'adaptation des méthodes de classification classiques aux données de grande dimension.

Bien que la sélection de modèles ait fait l'objet d'un développement intense ces 30 dernières années, il n'existe que peu d'approches alternatives à la stratégie de minimisation d'une borne supérieure du taux d'erreur réel initialement proposée par Vapnik pour la sélection de modèles en classification. Par ailleurs, la plupart de ces critères pénalisés sont comparativement moins performants que les méthodes de rééchantillonnage de type validation-croisée ou validation par un échantillon test, couramment employées par la communauté de l'apprentissage. Nous présentons au chapitre 3 une méthode alternative de sélection de modèles par critère pénalisé. Cette méthode est alternative au sens où elle ne repose pas sur la minimisation d'une borne sur le taux d'erreur réel, mais plutôt sur l'estimation du taux d'erreur conditionnel. Le taux d'erreur conditionnel est le taux d'erreur mesuré sur l'ensemble des individus de la population ayant une information identique à celle de l'un des individus de l'échantillon d'entraînement. Ce nouveau critère, appelé Swapping, est appliqué au problème du choix du nombre de voisins dans l'algorithme des kNN (*k Nearest Neighbors*). Plusieurs exemples d'applications à des jeux de données simulées et réelles montrent que les performances obtenues par le critère Swapping sont comparables aux performances obtenues par validation croisée, et que l'estimation du taux d'erreur est généralement meilleure avec le critère Swapping.

Les chapitres 1 et 2 présentent les problèmes de construction d'un classificateur et de sélection de variables de manière indépendante. Il semble toutefois pertinent de considérer simultanément ces deux problèmes, tant du point de vue pratique que théorique. Du point de vue pratique, beaucoup de méthodes de sélection de variables sont basées sur la minimisation du taux d'erreur empirique. C'est le cas des familles des méthodes *wrapper* et des méthodes intégrées (*embedded*) de sélection de variables. Cette sélection peut alors être considérée comme une étape à part entière de la construction du classificateur. Du point de vue théorique, il existe peu de résultats garantissant les performances des méthodes de sélection de variables. A l'inverse, la théorie de Vapnik fournit un cadre rigoureux pour étudier les performances du classificateur construit par application de la méthode ERM. De ce fait, intégrer l'étape de sélection de variables à la construction du classificateur peut permettre de garantir les performances de la méthode de sélection de variables. Nous mettons en oeuvre cette

stratégie au chapitre 4. En particulier, nous montrons que les performances des méthodes de sélection de variables intégrées et *wrapper* peuvent être garanties par une inégalité oracle. Nous utilisons ensuite les résultats théoriques obtenus comme point de départ pour l'analyse de l'algorithme de classification CART (*Classification And Regression Tree*). Nous montrons en particulier que CART peut être considéré comme une méthode de sélection de variables intégrée, et que la méthode de sélection de l'arbre optimal usuellement employée prend implicitement en compte cette étape de sélection de variables.

Enfin, nous nous intéressons aux limites de la sélection de variables appliquée aux données de grande dimension. Lorsque le nombre de variables est élevé se pose le problème de la redondance : une même information sur le label peut être portée par plusieurs variables. La sélection de variables réduit cette redondance de manière peu satisfaisante : lorsque plusieurs variables sont identiques du point de vue de l'information apportée sur le label, seule l'une de ces variables est retenue pour la construction du classificateur. La variable retenue acquiert alors un rôle privilégiée : sa présence dans le classificateur final sera interprétée comme la preuve de son implication dans le processus étudié. Ceci pose clairement le problème de l'interprétation des résultats d'une sélection de variables. Au chapitre 5, nous proposons un autre traitement de la redondance. Nous présentons une stratégie d'agrégation de variables, préalable et complémentaire à l'étape de sélection de variables, dont l'objectif est de rendre le classificateur final propre à l'interprétation. Nous verrons que l'objectif d'interprétation peut servir l'objectif de classement, et que l'agrégation de variables permet d'améliorer les performances du classificateur construit.

Une analyse efficace ne peut être entreprise sans une bonne connaissance de la nature des données à traiter. Cette connaissance n'est pas toujours considérée comme un préalable indispensable pour le développement de nouvelles méthodes, elle s'avère pourtant nécessaire dans bien des cas : sans cette connaissance, il est difficile de proposer une bonne modélisation du problème, voir de répondre aux bonnes questions. Par ailleurs, la collaboration entre le statisticien et le spécialiste du domaine qui réalise l'expérience ne commence pas lors de l'analyse des données. L'expertise du statisticien est utile à l'élaboration du plan d'expérience, ou lors des différentes étapes de pré-traitement des données, et peut sensiblement améliorer la qualité des données à analyser.

Deux études sur les données de puce à ADN sont présentées en annexe. La première aborde le problème de la normalisation. Les données de puces à ADN sont en effet caractérisées par leur grande variabilité. Une part de cette variabilité est d'origine biologique, et intéresse l'expérimentateur. Mais une part non négligeable de cette variabilité est d'origine technique. Le processus expérimental allant de l'extraction des cellules d'un organisme à la mesure de l'expression des gènes est complexe, et chaque étape de ce processus est une source potentielle de variabilité technique. L'étape de dépot des séquences d'ADN sur les puces est étudiée en détail en annexe B. La deuxième étude porte sur le mélange de matériel biologique provenant de différents individus, pratique usuelle chez les biologistes lorsque les quantités de matériel prélevées sont trop faibles. Les conséquences de cette pratique sont étudiées en annexe C.

Chapitre 1

Théorie de Vapnik

Cette partie a pour objet l'introduction des notions et des notations qui seront utilisées par la suite. Le lecteur ne trouvera pas ici de résultat nouveau, mais une présentation des principaux concepts de la théorie de Vapnik. Cette présentation servira de base à la partie suivante, dédiée à la sélection de modèles par Swapping, mais aussi à la partie 4, où un résultat théorique de sélection de variables directement inspiré de la théorie de Vapnik sera démontré.

L'objectif de la classification supervisée est de retrouver ou de prédire l'appartenance d'une observation à une classe donnée de la population à partir d'un certain nombre d'informations recueillies sur l'individu. Une observation peut ainsi être décrite comme une collection de mesures numériques $x : x^1,...,x^p$ (par exemple le salaire, l'âge ou la taille d'un individu), et par son appartenance $y = 0$ ou 1 à une classe (sain ou malade, chômeur ou actif...). L'objectif de la classification est de créer une fonction $\Phi(x)$ représentant la prédiction de y sachant x. Cette fonction est appelée classificateur. On dit que le classificateur se trompe en x lorsque $\Phi(x) \neq y$.

Bien sûr, nous cherchons le "meilleur" classificateur, c'est-à-dire celui se trompant le moins possible dans ses prédictions. La difficulté de ce problème vient du fait que y n'est pas une fonction déterministe de x: deux individus ayant la même valeur de x peuvent appartenir à des classes différentes. Cette constatation montre la nécessité d'introduire un cadre probabiliste pour décrire le problème de la classification supervisée. Un individu sera maintenant décrit comme la réalisation du vecteur aléatoire (X,Y) appartenant à $\mathbb{R}^p \times \{0,1\}$. Le classificateur fait donc une erreur lorsque $\Phi(X) \neq Y$, et la probabilité d'erreur (appelée aussi erreur, risque ou taux d'erreur réel) d'un classificateur est donnée par :

$$L(\Phi) = \mathbf{P}\{\Phi(X) \neq Y\}$$

Il existe un classificateur Φ^* optimal, au sens où ce classificateur vérifie $\Phi^* = \underset{\Phi}{\text{Argmin}}\mathbf{P}\{\Phi(X) \neq Y\}$. Il est clair que Φ^* dépend de la distribution de (X,Y). Si cette dernière est connue, alors Φ^* peut être calculé. Cette fonction particulière est appelée classificateur de Bayes, et l'erreur $L^* = L(\Phi^*)$ qui lui est associée est appelé risque de Bayes.

Généralement, la distribution de (X,Y) est inconnue. Pour construire un classificateur, nous ne disposons que d'un certain nombre d'exemples (X_i,Y_i), constituant l'échantillon d'entraînement. On suppose ces données représentatives de la population et indépendantes (i.i.d.), et tirées de la distribution (X,Y). Le choix d'un classificateur en fonction des données est ap-

pelé apprentissage, apprentissage statistique ou apprentissage supervisé.

La partie 1.1 présente une méthode simple et classique de classification supervisée, l'analyse discriminante de Fisher. Les limites de l'analyse discriminante et des méthodes paramétriques en général forment le point de départ de la théorie de Vapnik, dont les concepts et la principale démonstration sont présentés en parties 1.2 et 1.3.

1.1 L'Analyse Discriminante de Fisher

1.1.1 La règle de Bayes

Nous avons vu précédemment qu'il existe un classificateur Φ^* optimal au sens de l'erreur, appelé classificateur de Bayes. Il n'est pas difficile de montrer (cf. Devroye *et al.* (1996)) que le classificateur de Bayes n'est autre que la fonction :

$$\Phi^*(x) = \begin{cases} 1 & \text{si } \mathbf{P}\left(Y=1|X=x\right) > 1/2 \\ 0 & \text{sinon.} \end{cases}$$

La probabilité conditionnelle $\mathbf{P}\left(Y=1|X=x\right)$ peut se décomposer à l'aide de la formule de Bayes de la manière suivante :

$$\mathbf{P}\left(Y=1|X=x\right) = \frac{f_1(x)\pi_1}{f_0(x)\pi_0 + f_1(x)\pi_1} \tag{1.1}$$

où $f_k(x)$ est la fonction de vraisemblance conditionnelle de X dans le groupe $k=0$ ou 1, et π_k est la probabilité *a priori* d'appartenir au groupe k. Les vraisemblances conditionnelles comme les probabilités *a priori* sont inconnues. Le principe de l'analyse discriminante est de partir de la décomposition 1.1 pour "imiter" le classificateur bayésien, et ainsi obtenir une erreur proche de l'erreur minimum.

1.1.2 Les hypothèses de l'analyse discriminante

L'approche de l'analyse discriminante consiste à déterminer $f_k(x)$ et π_k par estimation. A partir d'un échantillon représentatif, il est facile d'estimer les probabilités *a priori* π_k. En revanche, le problème de l'estimation des densités est plus compliqué. Deux solutions s'offrent alors :

- Estimer les densités à l'aide d'estimateurs à noyaux. On fait alors de l'estimation non paramétrique, et cette méthode peut se révéler extrêmement coûteuse (en terme de nombre d'observations) lorsque X est de grande dimension.
- Simplifier le problème en imposant une forme déterminée pour les densités. On peut par exemple supposer que les densités conditionnelles $f_k(x)$ sont des densités de loi normale de paramètres inconnus. C'est le choix qui est fait en analyse discriminante.

L'intérêt de la deuxième solution est de se ramener à un problème d'estimation paramétrique en modèle gaussien, pour lequel on dispose de toute une méthodologie connue aux propriétés satisfaisantes (estimateurs efficaces, convergence).

On suppose donc que $f_0(x)$ et $f_1(x)$ sont des densités d'espérance et de matrices de variance inconnues, que l'on estime par maximum de vraisemblance. Le classificateur ainsi construit à partir de l'échantillon est alors :

$$\Phi_n^*(x) = \begin{cases} 1 & \text{si } \widehat{\mathbf{P}}(Y=1|X=x) > 1/2 \\ 0 & \text{sinon} \end{cases} \quad \text{où } \widehat{\mathbf{P}}(Y=1|X=x) = \frac{\widehat{f_1}(x)\widehat{\pi_1}}{\widehat{f_0}(x)\widehat{\pi_0}+\widehat{f_1}(x)\widehat{\pi_1}}$$

1.1.3 Limites de l'analyse discriminante

Les limites de l'analyse discriminante sont nombreuses. Elles tiennent principalement à l'hypothèse de normalité posée pour les vraisemblances conditionnelles. Plus l'écart à cette hypothèse sera important, moins il sera possible de garantir les performances du classificateur choisi. Par ailleurs, même si le modèle paramétrique permet de réduire la complexité du problème, il est clair que le nombre de paramètres à estimer (espérances et matrices de variance) ne peut dépasser le nombre d'observations de l'échantillon.

Toutefois, la critique formulée dans les travaux de Vapnik vis-à-vis des méthodes paramétriques (mais aussi des méthodes à noyaux classiques) est d'un tout autre ordre. De l'objectif premier qui était de construire un classificateur ayant les mêmes performances que le classificateur de Bayes, les méthodes paramétriques ont dérivé vers l'estimation de la distribution conditionnelle. Cette digression s'avère coûteuse : connaître complètement la distribution des données est suffisant mais peut ne pas être nécessaire pour construire un bon classificateur. Pour s'en convaincre, on peut observer que si l'on ne s'intéresse qu'au classement d'un individu il suffit de savoir si $\widehat{\mathbf{P}}\{Y = 1|X = x\} > \widehat{\mathbf{P}}\{Y = 0|X = x\}$, ce qui ne revient pas à connaître les valeurs exactes de chacune de ces probabilités. Remarquons toutefois qu'il existe des cas où la connaissances des probabilités *a posteriori* est intéressante, car plus informative que la seule connaissance de leur rapport.

1.2 Minimisation du risque empirique

1.2.1 Fondement de la méthode

La première idée fondamentale de la théorie de Vapnik est de repartir de l'objectif initial et de traduire le problème de la classification en un problème de minimisation du risque empirique (Empirical Risk Minimization). On dispose maintenant d'un échantillon, ainsi que d'un certain nombre de classificateurs réunis dans une classe $\mathcal{C}$. On veut choisir parmi ces classificateurs le meilleur au sens du taux d'erreur réel, c'est-à-dire le classificateur : $\Phi^*_{\mathcal{C}} = \underset{\Phi\in\mathcal{C}}{\text{Argmin}} L(\Phi)$. Toutefois, ne disposant pas de la loi de (X,Y), on ne peut pas calculer ce taux d'erreur. L'idée est donc d'estimer le taux d'erreur réel d'un classificateur Φ par son taux d'erreur empirique :

$$\hat{L}_n(\Phi) = \frac{1}{n}\sum_{i=1}^{n} I_{\{\Phi(X_i)\neq Y_i\}}$$

On choisira ensuite le classificateur Φ^*_n minimisant l'erreur empirique parmi les classificateurs de la classe $\mathcal{C}$. Pour que cette stratégie s'avère pertinente, il faut pouvoir justifier chacune des étapes :

- Il faut démontrer que l'estimateur empirique de l'erreur est un estimateur satisfaisant (i.e. consistant) de l'erreur véritable d'un classificateur. Concrètement, $\hat{L}_n(\Phi) - L(\Phi)$ doit converger vers 0 en probabilité pour tout Φ.
- Il faut aussi montrer que l'erreur véritable du classificateur Φ^*_n choisi tend vers l'erreur du meilleur estimateur de la classe $\Phi^*_{\mathcal{C}}$, c'est-à-dire montrer que $L(\Phi^*_n) - L(\Phi^*_{\mathcal{C}})$ tend aussi vers 0 en probabilité.

Il est facile de constater que ces deux convergences sont assurées si l'on garantit la convergence uniforme du risque empirique vers le risque réel sur la classe $\mathcal{C}$. En effet, supposons la convergence uniforme de l'erreur empirique. On a :

$$\mathbf{P}\left(|\hat{L}_n(\Phi) - L(\Phi)| > \varepsilon\right) \leq \mathbf{P}\left(\sup_{\Phi\in\mathcal{C}}|\hat{L}_n(\Phi) - L(\Phi)| > \varepsilon\right) \quad ,$$

ce qui garantit la consistance de l'erreur empirique. Passons maintenant à la convergence de $L(\Phi_n^*)$ vers $L(\Phi_{\mathcal{C}}^*)$. On a :

$$\begin{aligned} L(\Phi_n^*) - L(\Phi_{\mathcal{C}}^*) &= L(\Phi_n^*) - \hat{L}_n(\Phi_{\mathcal{C}}^*) + \hat{L}_n(\Phi_{\mathcal{C}}^*) - L(\Phi_{\mathcal{C}}^*) \\ &\leq L(\Phi_n^*) - \hat{L}_n(\Phi_n^*) + \hat{L}_n(\Phi_{\mathcal{C}}^*) - L(\Phi_{\mathcal{C}}^*) \\ \Rightarrow |L(\Phi_n^*) - L(\Phi_{\mathcal{C}}^*)| &\leq |L(\Phi_n^*) - \hat{L}_n(\Phi_n^*)| + |\hat{L}_n(\Phi_{\mathcal{C}}^*) - L(\Phi_{\mathcal{C}}^*)| \\ &\leq 2 \sup_{\Phi \in \mathcal{C}} |\hat{L}_n(\Phi) - L(\Phi)| \ . \end{aligned}$$

Il suffit de passer aux probabilités pour avoir :

$$\mathbf{P}\left(|L(\Phi_n^*) - L(\Phi_{\mathcal{C}}^*)| > \varepsilon\right) \leq \mathbf{P}\left(2 \sup_{\Phi \in \mathcal{C}} |\hat{L}_n(\Phi) - L(\Phi)| > \varepsilon\right) \ ,$$

ce qui achève la démonstration. Il ne reste plus qu'à déterminer les conditions nécessaires pour que l'erreur empirique converge uniformément sur la classe $\mathcal{C}$. Notons que selon les travaux initiaux de Vapnik, les conditions de convergence ne devaient porter que sur la classe elle-même, et non sur la distribution des données (X,Y), puisque l'on suppose cette dernière inconnue. Les travaux de Tsybakov (2004) ont depuis montré qu'il était profitable de faire des hypothèses sur la distribution des données (condition de marge), pour atteindre des vitesses de convergence plus rapides. Nous ne présentons ici que la démonstration classique du résultat de Vapnik, qui fait l'objet de la partie suivante.

1.2.2 Consistence de la méthode

Vitesse de convergence recherchée

Commençons par préciser le type de résultat que l'on recherche. On désire non seulement montrer que

$$\mathbf{P}\left(\sup_{\Phi \in \mathcal{C}} |\hat{L}_n(\Phi) - L(\Phi)| > \varepsilon\right)$$

converge vers 0, mais aussi montrer que cette convergence est "rapide". Toute vitesse de convergence en probabilité peut être comparée à celle obtenue par le théorème central limite (TCL). Rappelons l'énoncé de ce théorème :

Théorème 1.2.1. *Soit X_1,...,X_n une suite de variables aléatoires i.i.d. de moyenne μ et de variance σ^2. Lorsque n tend vers l'infini, on a :*

$$\frac{\sqrt{n}}{\sigma}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right) \hookrightarrow \mathcal{N}(0,1) \tag{1.2}$$

Asymptotiquement, la vitesse de convergence en probabilité de $\dfrac{1}{n}\sum_{i=1}^{n} X_i$ vers μ se calcule comme suit :

$$\mathbf{P}\left(\frac{\sqrt{n}}{\sigma}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right) > \varepsilon\right) \approx 1 - \mathtt{F}(\varepsilon) \quad \Rightarrow \quad \mathbf{P}\left(\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right) > \varepsilon\right) \approx e^{-n\varepsilon^2/2\sigma^2}$$

(voir Györfi (2002)) On estimera avoir une convergence en probabilité "rapide" lorsque la vitesse de convergence sera du même ordre que celle obtenue avec le TCL. L'objectif est donc de démontrer un résultat du type :

$$\mathbf{P}\left(\sup_{\Phi\in\mathcal{C}}|\hat{L}_n(\Phi)-L(\Phi)|>\varepsilon\right)\leq Ae^{-Bn\varepsilon^2} \text{ où } A \text{ et } B \text{ sont des constantes positives}$$

Nous verrons en partie 1.2.3 que les décroissances exponentielles garantissent des performances satisfaisantes pour la stratégie ERM.

Interprétation du problème en termes de fonction de répartition

Notons ν la mesure de probabilité de (X,Y). L'objectif est ici de montrer qu'étudier $\sup_{\Phi\in\mathcal{C}}|\hat{L}_n(\Phi)-L(\Phi)|$ revient à étudier $\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|$. Nous nous ramènerons ainsi au problème de la convergence en probabilité de la mesure de probabilité empirique, pour lequel nous disposons des résultats fournis par Glivenko et Cantelli. Il sera alors possible d'adapter leur démonstration au problème considéré. Soit ν la loi de probabilité de (X,Y). On a :

$$\begin{aligned} L(\Phi) &= \nu\left(\{(x,y) \ : \ \phi(x)\neq y\}\right) \\ \hat{L}_n(\Phi) &= \nu_n\left(\{(x,y) \ : \ \phi(x)\neq y\}\right) \ , \end{aligned}$$

où $\nu_n(A)=\frac{1}{n}\sum_{i=1}^{n} I_{\{(x_i,y_i)\in A\}}$

D'où

$$\sup_{\Phi\in\mathcal{C}}|\hat{L}_n(\Phi)-L(\Phi)|=\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|$$

où $\mathcal{A}=\bigcup_{\Phi\in\mathcal{C}}\{(x,y)/\Phi(x)\neq y\}$

Que peut-on dire de la convergence de ν_n vers ν ? Dans le cas où les parties A de $\mathbb{R}$ ont la forme particulière $]-\infty,z]$, on dispose du résultat suivant proposé par Glivenko-Cantelli :

Théorème 1.2.2. *Soit $Z_1,...,Z_n$ une suite de variables aléatoires réelles de fonction de répartition $F(x)=\boldsymbol{P}\{Z_1<z\}$. Alors :*

$$\boldsymbol{P}\left(\sup_{z\in\mathbb{R}}|F(z)-F_n(z)|>\varepsilon\right)\leq 8(n+1)e^{-n\varepsilon^2/32}$$

Le théorème de Glivenko Cantelli démontre donc la convergence uniforme de $\nu_n(A)$ vers $\nu(A)$ dans un cas particulier. Nous allons maintenant détailler la démonstration de ce théorème, et dans la partie suivante nous présenterons les modifications à apporter à cette démonstration pour obtenir l'inégalité de Vapnik-Chervonenkis démontrant la convergence uniforme de l'erreur empirique sur une classe.

Démonstration du théorème de Glivenko Cantelli

La démonstration présentée ici est directement inspirée de celle présentée dans Devroye *et al.* (1996). Elle est décomposée en quatre étapes de majoration successives.

Etape 1 : Symétrisation à l'aide d'un échantillon fantôme.
Soit $Z'_1,...,Z'_n$ une suite de variables aléatoires réelles telles que $Z'_1,...,Z'_n,Z_1,...,Z_n$ est une

suite i.i.d. On note ν'_n la mesure empirique calculée avec l'échantillon Z'_1,...,Z'_n. On définit aussi A^* une partie de $\mathcal{A}$ quelconque. On a :

$$\begin{aligned}\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu'_n(A)|>\varepsilon/2\right) &\geq \mathbf{P}\left(|\nu_n(A^*)-\nu'_n(A^*)|>\varepsilon/2\right)\\ &\geq \mathbf{P}\left\{|\nu_n(A^*)-\nu(A^*)|>\varepsilon \bigcap |\nu'_n(A^*)-\nu(A^*)|<\varepsilon/2\right\}\end{aligned}$$

En effet, l'intersection des événements $|\nu_n(A^*)-\nu(A^*)|>\varepsilon$ et $|\nu'_n(A^*)-\nu(A^*)|<\varepsilon/2$ implique l'événement $|\nu_n(A^*)-\nu'_n(A^*)|>\varepsilon/2$. Par ailleurs, ces deux événements sont indépendants (puisque les échantillons $(Z_1,..,Z_n)$ et Z'_1,...,Z'_n sont indépendants. Ainsi :

$$\mathbf{P}\left(|\nu_n(A)-\nu'_n(A)|>\varepsilon/2\right) \geq \mathbf{P}\left(|\nu_n(A^*)-\nu(A^*)|>\varepsilon\right)\times\mathbf{P}\left\{|\nu'_n(A^*)-\nu(A^*)|<\varepsilon/2\right\}$$

Le deuxième terme de ce produit peut être borné en utilisant d'une part l'inégalité de Tchebychev :

$$\mathbf{P}\left(|Z-E(Z)|>t\right)\leq V(Z)/t^2$$

et d'autre part le fait que $|\nu'_n(A^*)-\nu(A^*)|$ peut être décomposé en somme de variables de Bernoulli centrées :

$$\nu'_n(A^*)-\nu(A^*)=\frac{1}{n}\sum_{i=1}^{n}[\underbrace{I_{\{Z'_i\in A^*\}}-E\left(I_{\{Z'_i\in A^*\}}\right)}_{\text{Bernoulli centrées}}]$$

Chaque Bernoulli est de paramètre $\mathbf{P}\{A^*\}$. Leur variance s'écrit $\mathbf{P}\{A^*\}(1-\mathbf{P}\{A^*\})<1/4$. Si l'on suppose en plus que $n\varepsilon^2>2$, on obtient :

$$\begin{aligned}\mathbf{P}\left\{|\nu'_n(A^*)-\nu(A^*)|<\varepsilon/2\right\} &\geq 1-\frac{\mathbf{P}\{A^*\}(1-\mathbf{P}\{A^*\})}{n\varepsilon^2/4}\\ &\geq 1/2\\ \text{D'où } \mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu'_n(A)|>\varepsilon/2\right) &\geq \frac{1}{2}\mathbf{P}\left(|\nu_n(A^*)-\nu(A^*)|>\varepsilon\right)\end{aligned}$$

La dernière inégalité est vraie pour toute partie de $\mathcal{A}$. En particulier, elle est vraie pour une partie telle que $|\nu_n(A^*)-\nu(A^*)|>\varepsilon$. On a alors :

- Si une telle partie existe : $\mathbf{P}\left(|\nu_n(A^*)-\nu(A^*)|>\varepsilon\right)=1\geq\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|>\varepsilon\right)$
- Sinon, $\mathbf{P}\left(|\nu_n(A^*)-\nu(A^*)|>\varepsilon\right)=0$ et $\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|>\varepsilon\right)=0$ d'où

  $\mathbf{P}\left(|\nu_n(A^*)-\nu(A^*)|>\varepsilon\right)\geq\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|>\varepsilon\right)$

L'étape 1 permet donc de conclure que

$$\frac{1}{2}\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|>\varepsilon\right) \leq \mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu'_n(A)|>\varepsilon/2\right)$$

Etape 2 : Symétrisation par variables de Rademacher
Soient σ_1,...,σ_n n variables i.i.d, indépendantes de Z_1,..,Z_n et Z'_1,...,Z'_n, et telles que $\mathbf{P}\{\sigma_i=1\}=\mathbf{P}\{\sigma_i=-1\}=1/2$. La distribution de la variable :

$$\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\left(I_A(Z_i)-I_A(Z'_i)\right)|$$

est identique à celle de la variable :

$$\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i\left(I_A(Z_i)-I_A(Z_i')\right)|$$

Nous pouvons donc reprendre le résultat de l'étape 1, et y faire apparaître les variables σ_i :

$$\begin{aligned}\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|>\varepsilon\right) &\leq 2\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu_n'(A)|>\varepsilon/2\right)\\ &\leq 2\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}(\sigma_i\left(I_A(Z_i)-I_A(Z_i')\right))|>\varepsilon/2\right)\\ &\leq 2\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4 \text{ ou } \sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i')|>\varepsilon/4\right)\end{aligned}$$

En effet, si ni $|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|$ ni $|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i')|$ n'est supérieure à $\varepsilon/4$, alors leur somme est strictement inférieure à $\varepsilon/2$, or cette somme est un majorant de $|\frac{1}{n}\sum_{i=1}^{n}(\sigma_i\left(I_A(Z_i)-I_A(Z_i')\right))|$.
La probabilité d'une union d'événements peut être bornée par la somme des probabilités des événements :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|>\varepsilon\right) \leq 2\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\right)+2\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i')|>\varepsilon/4\right)$$

Enfin, les deux probabilités sont égales puisque les variables Z_1,..,Z_n et Z_1',...,Z_n' ont la même distribution. A la fin de l'étape 2, on a :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|>\varepsilon\right) \leq 4\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\right)$$

Etape 3 : Conditionnement par les Z_i
C'est ici que vont servir les variables σ_i introduites à l'étape précédente. Grâce à elles, il va être possible de conditionner la somme des $\sigma_i Z_i$ par les Z_i tout en gardant un aléa. On a :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\right)=\int_{Z_1,...,Z_n}\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \Big|Z_1,...,Z_n\right)$$

Un majorant indépendant des Z_i pour la probabilité conditionnelle sera donc aussi un majorant du terme de gauche de l'inégalité (car l'intégrale du majorant sur les Z_i sera égale au majorant lui-même). On s'intéresse donc à cette probabilité conditionnelle. Pour $A=]-\infty,z]$ on peut écrire :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \Big|Z_1,...,Z_n\right)=\mathbf{P}\left\{\sup_{z\in\mathbb{R}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_{\{Z_i<z\}}|>\varepsilon/4\ \Big|Z_1,...,Z_n\right\}$$

Une fois les Z_i fixés par le conditionnement, on constate que le vecteur $(I_{\{Z_1<z\}},...,I_{\{Z_n<z\}})$ ne peut prendre en fonction de z que n+1 valeurs différentes, que l'on notera v_1,...v_{n+1}. D'où :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\tfrac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \big|Z_1,...,Z_n\right)=$$

$$\mathbf{P}\left\{\sup_{A/v_1}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4 \text{ ou ... ou } \sup_{A/v_{n+1}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \big|Z_1,...,Z_n\right\}$$

Pour chaque v_i fixé, la valeur absolue considérée ne dépend plus de A, on peut donc retirer le sup. Par ailleurs, on a ici aussi la probabilité d'une réunion, on peut donc la majorer par la somme des $n+1$ probabilités. Pour finir, chacune de ces probabilité est inférieure ou égale au sup de la probabilité sur la classe. On obtient à l'étape 3:

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \big|Z_1,...,Z_n\right)\leq(n+1)\sup_{A\in\mathcal{A}}\mathbf{P}\left\{|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \big|Z_1,...,Z_n\right\}$$

Etape 4 : Borne exponentielle pour la probabilité conditionnelle
Maintenant que le sup se trouve à l'extérieur de la probabilité conditionnelle, il suffit de borner cette dernière. C'est ici qu'interviennent les inégalités de concentration,qui permettent de "contrôler" la valeur d'une variable aléatoire autour de son espérance [1]. Nous utilisons l'inégalité de Hoeffding :

Théorème 1.2.3. *Soient $X_1,...,X_n$ des variables aléatoires telles que X_i prend ses valeurs dans $[a_i,b_i]$. Soit S_n leur somme. Pour tout $\varepsilon>0$, on a :*

$$\boldsymbol{P}\{S_n-E(S_n)\geq\varepsilon\}\leq e^{-2\varepsilon^2/\sum_{i=1}^{n}(b_i-a_i)^2}$$

$$\text{et } \boldsymbol{P}\{S_n-E(S_n)\leq-\varepsilon\}\leq e^{-2\varepsilon^2/\sum_{i=1}^{n}(b_i-a_i)^2}$$

Dans notre cas, chaque variable $\sigma_i I_A(Z_i)$ est une variable prenant ses valeurs dans [1,-1], et dont l'espérance est nulle. L'application du théorème 1.2.3 donne :

$$\mathbf{P}\left\{|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \big|Z_1,...,Z_n\right\}\leq 2e^{-n\varepsilon^2/32}$$

On peut donc reprendre la probabilité conditionnelle de l'étape 3 et la borner de la manière suivante :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \big|Z_1,...,Z_n\right)\leq 2(n+1)e^{-n\varepsilon^2/32}$$

Il ne reste plus qu'à intégrer sur $Z_1,...,Z_n$ les deux cotés de l'inégalité :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)|>\varepsilon/4\ \right)\leq 2(n+1)e^{-n\varepsilon^2/32}$$

Pour conclure, en reprenant l'inégalité obtenue à l'étape 2, on retrouve bien :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)|>\varepsilon\right)\leq 8(n+1)e^{-n\varepsilon^2/32}$$

1. Remarque : l'obtention d'inégalités de concentration performantes, comme celles démontrées par McDiarmid ou Ledoux et Talagrand, a fortement contribué au développement théorique de l'apprentissage statistique.

Démonstration de l'inégalité de Vapnik-Chervonenkis

Nous revenons maintenant à notre problème initial de la convergence de l'erreur empirique. Nous venons de voir que lorsque les parties A de $\mathcal{A}$ sont de la forme $]-\infty,z]$, la convergence est établie. Quelles sont les modifications à apporter lorsque les parties A sont d'une forme plus générale ? Dans la démonstration précédente, la forme des parties A intervient à l'étape 3, pour compter le nombre de vecteurs v_i différents que l'ensemble des parties A de $\mathcal{A}$ engendre (i.e. dans le terme "$(n+1)$"). Ce nombre est aussi le nombre de sous-ensembles différents $\{z_1,...z_n\}\bigcap]-\infty,z]$, $z \in \mathbb{R}$ que l'on peut engendrer. Il faut donc reprendre la démonstration à cette étape, et déterminer le cardinal de l'ensemble $\{\{z_1,...z_n\}\bigcap A,\ A \in \mathcal{A}\}$, où $\mathcal{A}$ n'a plus de forme particulière. Pour cela, on introduit deux notions, le "shatter-coefficient" (ou coefficient de hachage) et la VC-dimension (ou dimension de Vapnik) :

Définition 1.2.1. *Le shatter-coefficient d'ordre n associé à la classe $\mathcal{A}$ est l'entier* $\mathcal{S}(\mathcal{A},n) = \max_{(z_1,...,z_n)} Card\,\{\{z_1,...,z_n\}\bigcap A, A \in \mathcal{A}\}$

On remarque que le shatter-coefficient ne dépend que de la classe $\mathcal{A}$. Il en mesure sa richesse, en terme de capacité de séparation de n points.

Définition 1.2.2. *Soit $\mathcal{A}$ un ensemble de parties. On appelle dimension de Vapnik-Chervonenkis (ou VC-dimension) le plus grand entier $k \geq 1$ pour lequel $\mathcal{S}(\mathcal{A},k) = 2^k$. Cet entier est noté $V_{\mathcal{A}}$. Si $\mathcal{S}(\mathcal{A},n) = 2^n$ pour tout n, alors par définition $V_{\mathcal{A}} = \infty$*

La VC-dimension d'une classe de parties est donc le nombre maximum de points que l'on peut toujours séparer en deux groupes. Enfin, nous pouvons énoncer pour le shatter-coefficient la propriété suivante :

Propriété 1.2.1. *Pour tout $n > 2$, on a : $\mathcal{S}(\mathcal{A},n) \leq n^{V_{\mathcal{A}}}$*

Si l'on reprend la démonstration à l'étape 3, on voit que le nombre de vecteurs v_i que l'on peut constituer à partir de l'ensemble de parties $\mathcal{A}$ est majoré par le shatter-coefficient. On obtient alors :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)| > \varepsilon/4\ \Big|Z_1,...,Z_n\right) \leq \mathcal{S}(\mathcal{A},n)\sup_{A\in\mathcal{A}}\mathbf{P}\left\{|\frac{1}{n}\sum_{i=1}^{n}\sigma_i I_A(Z_i)| > \varepsilon/4\ \Big|Z_1,...,Z_n\right\}$$

D'où :

$$\mathbf{P}\left(\sup_{A\in\mathcal{A}}|\nu_n(A)-\nu(A)| > \varepsilon\right) \leq 8\mathcal{S}(\mathcal{A},n)e^{-n\varepsilon^2/32}$$

Pour finir, il faut repasser du supremum pris sur un ensemble de parties au supremum pris sur une classe de classificateurs, et pour cela définir le shatter coefficient et la VC-dimension pour une telle classe :

Définition 1.2.3. *Soient $\mathcal{C}$ une classe de classificateurs de la forme $\Phi : \mathbb{R}^d \to \{0,1\}$, et $\mathcal{A}$ la collections des parties de la forme :*

$$\{\{x : \Phi(x) = 1\} \times \{0\}\}\bigcup\{\{x : \Phi(x) = 0\} \times \{1\}\},\ \Phi \in \mathcal{C}$$

Le shatter-coefficient et la VC-dimension sont définis pour la classe $\mathcal{S}(\mathcal{A},n)$ par :

$$\mathcal{S}(\mathcal{C},n) = \mathcal{S}(\mathcal{A},n) \text{ et } V_{\mathcal{C}} = V_{\mathcal{A}}$$

On démontre ainsi l'inégalité de Vapnik-Chervonenkis :

Théorème 1.2.4. *Pour toute classe $\mathcal{C}$ de classificateurs, on a :*

$$\boldsymbol{P}\left(\sup_{\Phi\in\mathcal{C}}|\hat{L}_n(\Phi) - L(\Phi)| > \varepsilon\right) \leq 8\mathcal{S}(\mathcal{C},n)e^{-n\varepsilon^2/32}$$

Commentaires

La démonstration de l'inégalité de Vapnik présentée donne une vue d'ensemble des méthodes de majoration couramment utilisées en théorie de l'apprentissage (inégalités de concentration, symétrisation). Différentes améliorations de l'inégalité de concentration obtenue ont été proposées. Nous nous servirons dans la suite de l'inégalité suivante, proposée dans Lugosi et Zeger (1995) :

$$\mathbf{P}\left(\sup_{\Phi\in\mathcal{C}}|\hat{L}_n(\Phi) - L(\Phi)| > \varepsilon\right) \leq 4e^8\mathcal{S}(\mathcal{C},n^2)e^{-2n\varepsilon^2} \tag{1.3}$$

Par ailleurs, il a été préalablement démontré que la convergence uniforme de l'erreur empirique entraîne la convergence de $L(\Phi_n^*)$ vers $L(\Phi_{\mathcal{C}}^*)$. En reprenant la dernière inégalité, on obtient :

$$\mathbf{P}\left\{|L(\Phi_n^*) - L(\Phi_{\mathcal{C}}^*)| > \varepsilon\right\} \leq 4e^8\mathcal{S}(\mathcal{C},n^2)e^{-n\varepsilon^2/2} \tag{1.4}$$

Ce résultat appelle plusieurs commentaires. Tout d'abord, remarquons une fois de plus que l'inégalité (1.4) ne dépend pas de la distribution des données. Elle justifie la stratégie de minimisation du risque empirique lorsque la classe de classificateurs considérée n'est pas trop riche, c'est-à-dire de VC-dimension finie. Pour comprendre la portée de ce résultat, considérons deux exemples.

On suppose que les données (X_i,Y_i) sont des éléments de $\mathbb{R}\times\{0,1\}$. On se donne aussi une classe $\mathcal{F}$ de fonctions paramétriques f coupant $\mathbb{R}$, la droite de représentation des données, en 2 types de parties : les parties de $\mathbb{R}$ où $f(x)$ est positive, et celles où $f(x)$ est négative. On associe alors chacun des deux types de parties à une classe (0 ou 1). On définit bien ainsi la classe $\mathcal{C}$ des classificateurs de la forme : $\{x \ : \ I_{\{f(x)\geq 0\}}\}$. On peut considérer deux classes de fonctions paramétriques :

- l'ensemble des polynômes de degré inférieur ou égal à p fixé.
- l'ensemble des polynômes de degré quelconque.

On peut déterminer la VC-dimension associée à chacune de ces classes en appliquant le lemme suivant :

Lemme 1. *Si $\mathcal{C}$ est la classe de tous les demi-espaces vectoriels, c'est-à-dire de toutes les parties de $\mathbb{R}^d$ de la forme :$\{x \ : \ a^Tx \geq b\}$, où $a\in\mathbb{R}^d$ et $b\in\mathbb{R}$, alors $V_{\mathcal{C}} = d+1$.*

Si l'on considère la première classe, on remarque que tous ses éléments peuvent se mettre sous la forme : $\{x =: \sum_{i=0}^{p}\alpha_i x^i \geq \alpha_0\}$. Cette classe est donc assimilable à la classe de tous les demi-espaces vectoriels de $\mathbb{R}^p$, et le lemme précédent permet de conclure que sa VC-dimension est

$p+1$. Par ailleurs, la deuxième classe contient tous les demi-espaces de dimensions k, $k \in \mathbb{N}$, donc la deuxième classe est de VC-dimension infinie.

Ces deux exemples montrent que le théorème 1.4 permet de prendre en compte des classes dont le nombre d'éléments est infini, à condition que ces éléments n'aient pas un pouvoir séparateur trop fort (i.e. dans notre exemple, à condition que les fonctions ne puissent pas découper la droite en n tronçons pour tout n). En conclusion, la stratégie de minimisation du risque empirique n'est assurée que lorsque le pouvoir séparateur de la classe $\mathcal{C}$ est limité. Plus la classe est riche, plus la convergence sera lente.

Il est important de noter que la démonstration présentée ici démontre que la VC-dimension finie de la classe $\mathcal{C}$ est une condition suffisante pour que la stratégie soit efficace. Vapnik et Chervonenkis (1981) ont démontré que cette condition est aussi nécessaire.

Il faut enfin remarquer que la minimisation du risque empirique ne garantit pas la convergence de $L(\Phi_n^*)$ vers L^*, le risque de Bayes, mais vers $L(\Phi_{\mathcal{C}}^*)$. Dans le cas où la règle de Bayes fait partie de la classe de classificateurs considérée, L^* et $L(\Phi_{\mathcal{C}}^*)$ sont bien sûr identiques. Mais les classes de VC-dimension finie sont en général trop petites pour contenir à coup sûr la règle de Bayes, et l'erreur du meilleur estimateur de cette classe peut être beaucoup plus grand que L^*. Par ailleurs, les conclusions précédentes montrent qu'il est exclu de réaliser la minimisation du risque empirique sur des classes de VC-dimension infinie. Le choix de la classe de classificateurs fait l'objet de la minimisation du risque structurel (SRM), développée dans la partie 1.3.

1.2.3 Performances de la méthode ERM

Nous nous intéressons ici au risque de la stratégie ERM, défini de la manière suivante :

$$R(\Phi_{\mathcal{C}}^*,\Phi_n^*) = \mathbb{E}L(\Phi_n^*) - L(\Phi_{\mathcal{C}}^*)$$

On fera attention à bien distinguer l'*erreur* d'un classificateur (parfois appelée risque) $L(\Phi)$ du *risque* défini entre un estimateur et sa "cible" : $R(\Phi_{\mathcal{C}}^*,\Phi_n^*)$ par exemple. Le risque d'un estimateur mesure donc la "distance" que l'on peut attendre en moyenne entre le minimiseur de l'erreur empirique et le meilleur classificateur de la classe. Nous supposerons dans toute cette partie que le classificateur de Bayes se trouve dans la classe considérée (i.e. $\inf_{\Phi\in\mathcal{C}} L(\Phi) = L^*$). Pour déterminer le risque de la stratégie ERM, nous revenons à l'inégalité de Vapnik-Chervonenkis (1.4). Dans une classe $\mathcal{C}$ donnée, il est possible de déduire directement de cette inégalité une borne supérieure pour le risque de l'estimateur ERM :

Théorème 1.2.5. *Pour toute classe $\mathcal{C}$ de VC-dimension finie, on a :*

$$R(\Phi^*,\Phi_n^*) \leq c\sqrt{\frac{V_{\mathcal{C}}\log(n)}{n}} \tag{1.5}$$

où c est une constante universelle.

Pour démontrer le théorème précédent, nous avons besoin du lemme qui suit :

Lemme 2. *Si Z est une variable aléatoire positive telle que $\boldsymbol{P}(Z > t) \leq ce^{-2nt^2}$ pour tout t, avec $c > 0$, alors :*

$$\mathbb{E}(Z) \leq \sqrt{\frac{\log(ce)}{2n}}$$

Démonstration. On commence par majorer $\mathbb{E}(Z^2)$

$$\mathbb{E}(Z^2) \leq \int_0^\infty \mathbf{P}\left(Z^2 > t\right) dt \leq \int_0^u 1dt + \leq \int_u^\infty ce^{-2nt}dt \leq u + \frac{ce^{-2nu}}{2n}$$

Cette inégalité est vraie pour tout u, on peut donc choisir celui qui minimise le majorant, c'est-à-dire $u = \frac{\log(c)}{2n}$. On obtient alors :

$$\mathbb{E}(Z^2) \leq \frac{\log(c)}{2n} + \frac{ce^{-\log(c)}}{2n} \leq \frac{\log(ec)}{2n}$$

Il suffit ensuite d'appliquer l'inégalité de Jensen à $\mathbb{E}(Z^2)$ pour obtenir :

$$\mathbb{E}(Z) \leq \sqrt{\mathbb{E}(Z^2)} \leq \sqrt{\frac{\log(ec)}{2n}}$$

□

Nous pouvons maintenant démontrer le théorème 1.2.5.

Démonstration. En appliquant directement le lemme 2 à l'inégalité de concentration (1.4) pour $L(\Phi_n^*) - \inf_{\Phi \in \mathcal{C}} L(\Phi)$ puis en utilisant la propriété 1.2.1, on obtient la majoration :

$$R(\Phi^*,\Phi_n^*) \leq 2\sqrt{\frac{\log(4e^9\mathcal{S}(\mathcal{C},n^2))}{2n}} \leq c\sqrt{\frac{V_\mathcal{C}\log(n)}{n}} \ ,$$

ce qui achève la démonstration. □

Nous pouvons donc maintenant affirmer que le risque $R(\Phi^*,\Phi_n^*)$ est au plus de l'ordre de grandeur de $O\left(\sqrt{V_\mathcal{C}\log(n)/n}\right)$. Il faut alors déterminer si cet ordre de grandeur est satisfaisant, c'est-à-dire déterminer s'il existe un meilleur ordre de grandeur pour la minimisation du risque empirique, et s'il existe une méthode de sélection de classificateur ayant de meilleures performances. On peut répondre à ces deux questions en utilisant des résultats de type minimax. Nous présentons ici un résultat issu de Devroye *et al.* (1996). La démonstration n'est pas donnée ici, le lecteur intéressé pourra se reporter à la référence précédente.

Théorème 1.2.6. *(Simon (1993)) Soit $\mathcal{C}$ une classe de classificateurs de VC-dimension supérieure à 2. Soit $\mathcal{X}$ l'ensemble de toutes les variables aléatoires (X,Y) pour lesquelles on a, pour un $L \in]0,\frac{1}{2}[$ fixé :*

$$L = \inf_{g \in \mathcal{C}} \boldsymbol{P}\{g(X) \neq Y\}$$

Alors, pour toute stratégie de choix d'un classificateur g_n basée sur l'échantillon d'apprentissage $X_1,Y_1,...,X_n,Y_n$, il existe un N tel que pour tout $n \geq N$:

$$\sup_{(X,Y)\in\mathcal{X}} \mathbb{E}L(g_n) - L \geq \sqrt{\frac{L(V-1)}{24n}}e^{-8}$$

Si l'on compare la majoration (1.5) à la borne proposée par Simon, on constate que la vitesse de décroissance de la borne supérieure est trop lente, à cause du facteur logarithmique au numérateur. En passant par des techniques plus raffinées de "chaînage", il est possible de retirer ce facteur logarithmique et d'obtenir ainsi une borne supérieure optimale (Györfi

(2002)). Ainsi, quelle que soit la stratégie pour choisir un classificateur (minimisation du risque empirique ou autre), il existe toujours une distribution pour laquelle la convergence de la règle de classification vers le classificateur optimal est au moins aussi lente que la borne inférieure proposée en (1.5). Ce résultat minimax montre qu'il n'existe pas de règle de classement ayant une vitesse de convergence systématiquement meilleure que celle de la stratégie ERM.

Remarque : Les récents développements de la théorie de l'apprentissage montrent qu'il est en réalité possible d'obtenir des vitesses de convergence plus rapides que $\sqrt{n}$ en imposant des conditions de marge, portant sur la distribution des données. La notion de condition de marge fut initialement introduite dans Mammen et Tsybakov (1999), puis développée dans Tsybakov (2004) notamment. Cette notion peut être appliquée à la minimisation du risque empirique, le lecteur intéressé pourra consulter avec profit Massart et Nédélec (2006).

1.3 Minimisation du risque structurel (SRM)

1.3.1 Présentation de la stratégie

En l'absence d'information sur la règle de Bayes, nous ne pouvons pas considérer une classe unique $\mathcal{C}$ de classificateurs, car rien ne garantit l'existence de bons classificateurs (i.e. proches du classificateur de Bayes) dans cette classe. En effet, le risque $R(\Phi^*,\Phi^*_{n,\mathcal{C}})$ peut se décomposer comme suit :

$$\begin{aligned} R(\Phi^*,\Phi^*_{n,\mathcal{C}}) &= L(\Phi^*_{n,\mathcal{C}}) - L(\Phi^*) \\ &= \underbrace{L(\Phi^*_{n,\mathcal{C}}) - L(\Phi^*_{\mathcal{C}})}_{variance} + \underbrace{L(\Phi^*_{\mathcal{C}}) - L(\Phi^*)}_{biais} \end{aligned}$$

La première partie (variance) est appelée l'erreur d'estimation. Plus la classe $\mathcal{C}$ est complexe (de VC-dimension élevée), plus cette erreur est grande. La deuxième partie (biais) est appelée l'erreur d'approximation. Plus la classe $\mathcal{C}$ est complexe plus elle est susceptible de contenir le classificateur de Bayes, et plus cette erreur sera petite. Dans une classe $\mathcal{C}$ de VC dimension finie, les résultats précédents nous garantissent que le premier terme tend vers 0 lorsque n augmente, mais le deuxième terme est incompressible et peut être arbitrairement grand. Il faut donc considérer une série (finie ou dénombrable) de classes $\mathcal{C}_1$, $\mathcal{C}_2$,..., parmi lesquelles choisir. Afin d'exploiter au maximum l'information disponible, on utilisera l'échantillon d'entraînement pour choisir la classe, comme cela est fait pour le choix du classificateur dans une classe donnée.

Avant de proposer une quelconque stratégie, il est nécessaire d'avoir une idée du type de performances souhaité. Si nous avions de l'information *a priori* sur les classes, nous choisirions la classe $\mathcal{C}_{opt}$ vérifiant :

$$\begin{aligned} \mathcal{C}_{opt} &= \underset{\mathcal{C}_i}{\text{Argmin}}\ R(\Phi^*,\Phi^*_{n,\mathcal{C}_i}) \\ &= \underset{\mathcal{C}_i}{\text{Argmin}}\ R(\Phi^*,\Phi^*_{\mathcal{C}_i}) + R(\Phi^*_{\mathcal{C}_i},\Phi^*_{n,\mathcal{C}_i}) \end{aligned}$$

Autrement dit, nous choisirions la classe garantissant le meilleur compromis biais-variance pour la quantité d'observations à disposition. Cette classe optimale n'est évidemment pas accessible sans connaissance *a priori*, mais le risque de cette classe servira dans la suite de risque de référence : nous cherchons une stratégie garantissant la convergence en probabilité

de l'erreur du classificateur choisi vers l'erreur de Bayes, et dont le risque est comparable à celui que l'on obtiendrait en se plaçant directement dans la classe optimale. Formellement, notre objectif est donc de proposer une méthode de sélection d'un classificateur Φ_n^* parmi l'une des classes proposées, vérifiant :

$$R(\Phi^*,\Phi_n^*) \leq C \times \inf_i R(\Phi^*,\Phi_{n,\mathcal{C}_i}^*)$$

Ce type de résultat, appelé inégalité oracle, est à la base de la formulation désormais classique des résultats obtenus en sélection de modèles depuis les travaux de Donoho et Johnstone (1994) et dont la formulation générale fut proposée par Barron *et al.* (1995).

Choisir un classificateur parmi l'une des classes proposées est l'objectif de la minimisation du risque structurel, ou SRM (Structural Risk Minimization), dont le principe est le suivant. Ne connaissant pas la classe "optimale", on se donne un ensemble dénombrable de classes $\mathcal{C}_i$, de plus en plus grandes ($V_{\mathcal{C}_i} \leq V_{\mathcal{C}_{i+1}}$). Dans chaque classe, on détermine le minimiseur du risque empirique $\Phi_{n,\mathcal{C}_i}^*$. La comparaison directe des candidats $\Phi_{n,\mathcal{C}_i}^*$ à partir de leur erreur empirique $\hat{L}_n(\Phi_{n,\mathcal{C}_i}^*)$ n'est pas possible, car le biais de cette estimation est d'autant plus fort que la classe est complexe. Pour s'en convaincre, il suffit de constater que si l'échantillon d'apprentissage comporte n points, les candidats des classes de VC dimension supérieure à n auront un risque empirique de 0, alors que le risque de Bayes lui-même peut-être élevé.

L'alternative proposée par Vapnik est de trouver une borne supérieure (garantie avec une grande probabilité) de l'erreur de chacun des candidats, et de choisir ensuite le candidat ayant la borne supérieure la plus petite. Les deux chapitres qui suivent traitent de chacune des deux étapes de la stratégie :

- Trouver une borne $M_{\mathcal{C}_i}$ pour $L(\Phi_{n,\mathcal{C}_i}^*)$ tel que $P(L(\Phi_{n,\mathcal{C}_i}^*) < M_{\mathcal{C}_i}) = 1-\alpha$
- Prendre parmi tous les candidats $\Phi_{n,\mathcal{C}_i}^*$ celui ayant la plus petite borne $M_{\mathcal{C}_i}$.

La dernière partie démontre les performances de la règle de décision ainsi construite.

1.3.2 Consistence

Nous commençons par chercher une borne supérieure pour $L(\Phi_{n,\mathcal{C}_i}^*)$, erreur du minimiseur empirique de la classe $\mathcal{C}_i$. Il suffit pour cela de reprendre l'inégalité de concentration (1.3) :

$$\begin{aligned}
\mathbf{P}\left(L(\Phi_{n,\mathcal{C}_i}^*) - \hat{L}_n(\Phi_{n,\mathcal{C}_i}^*) > \varepsilon\right) &\leq \mathbf{P}\left(\sup_{\Phi\in\mathcal{C}} |\hat{L}_n(\Phi) - L(\Phi)| > \varepsilon\right) \\
&\leq 4e^8 \mathcal{S}(\mathcal{C}_i,n^2) e^{-2n\varepsilon^2} \\
&\leq \exp\left\{-2n\varepsilon^2 + \log(4e^8\mathcal{S}(\mathcal{C}_i,n^2))\right\} \\
\Rightarrow \mathbf{P}\left(L(\Phi_{n,\mathcal{C}_i}^*) > \hat{L}_n(\Phi_{n,\mathcal{C}_i}^*) + \sqrt{\frac{\delta_i + \log(4e^8\mathcal{S}(\mathcal{C}_i,n^2))}{2n}}\right) &\leq e^{-\delta_i}
\end{aligned}$$

en réalisant le changement de variable $\delta_i = 2n\varepsilon^2 - \log(4e^8\mathcal{S}(\mathcal{C},n^2))$. En prenant l'événement complémentaire, on obtient avec une probabilité de $1-e^{-\delta_i}$:

$$L(\Phi_{n,\mathcal{C}_i}^*) < \hat{L}_n(\Phi_{n,\mathcal{C}_i}^*) + \sqrt{\frac{\delta_i + \log(4e^8\mathcal{S}(\mathcal{C}_i,n^2))}{2n}} = M_{\mathcal{C}_i} \tag{1.6}$$

Cette borne supérieure est donc l'erreur empirique pénalisée par un terme dépendant de la complexité de la classe considérée. On notera que l'on retrouve ici le contexte de la sélection

de modèles par contraste pénalisé développée par Birgé et Massart (2001a) dans leur théorie de l'estimation par contraste pénalisé.

L'idée est maintenant de choisir le classificateur ayant la plus petite borne supérieure. Il faut toutefois prendre en compte le fait que l'on a garanti les niveaux de confiance des bornes supérieures indépendamment les uns des autres. On cherche maintenant à garantir toutes les bornes supérieures simultanément au niveau $e^{-\delta}$. On utilise pour cela l'inégalité de Bonferroni :

$$\begin{aligned}
\mathbf{P}\left\{\bigcup_{i\geq 1}\left\{L(\Phi^*_{n,\mathcal{C}_i}) > \hat{L}_n(\Phi^*_{n,\mathcal{C}_i}) \quad + \quad \sqrt{\frac{\delta_i + \log(4e^8\mathcal{S}(\mathcal{C},n^2))}{2n}}\right\}\right\} & \\
\leq \quad \sum \mathbf{P}\left\{L(\Phi^*_{n,\mathcal{C}_i}) > \hat{L}_n(\Phi^*_{n,\mathcal{C}_i}) + \sqrt{\frac{\delta_i + \log(4e^8\mathcal{S}(\mathcal{C},n^2))}{2n}}\right\}\right\} & \\
\leq \quad \sum e^{-\delta_i} &
\end{aligned}$$

Pour que toutes les majorations soient vérifiées simultanément avec une grande probabilité, cette dernière somme doit être finie et de limite inférieure à $e^{-\delta}$. Il faut donc diminuer les niveaux de confiance $e^{-\delta_i}$. On peut les décomposer de la manière suivante :

$$\delta_i = \delta + \delta_{\mathcal{C}_i} \text{ avec } \sum_{i\geq 1} e^{-\delta_{\mathcal{C}_i}} \leq 1$$

Dans cette décomposition, les $\delta_{\mathcal{C}_i}$ sont là pour garantir la sommabilité, et le δ garantit le niveau de confiance fixé.

La somme considérée étant infinie, il faut bien évidemment prendre des δ_i non constants, et donc différents pour chaque classe. Or il faut noter que lorsque l'on joue sur les valeurs des δ_i, on perturbe les valeurs des bornes supérieures, fonction des δ_i. Il ne faut donc pas que le contrôle des probabilités (i.e. la sommabilité des $^{-\delta_i}$) affecte trop sensiblement la borne, car on risquerait de disqualifier des classes que l'on aurait choisi autrement. On choisira donc des $\delta_{\mathcal{C}_i}$ tels que leur ordre de grandeur soit au plus comparable à celui de la borne supérieure obtenue pour la classe $\mathcal{C}_i$ (donc croissants avec la complexité de la classe). Nous décrivons ici la solution proposée par Lugosi et Zeger (1995).

Théorème 1.3.1. *Soient $\mathcal{C}_1$, $\mathcal{C}_2$,... une suite de classes de VC-dimension finies $V_{\mathcal{C}_1}<V_{\mathcal{C}_2}$... telle que pour toute distribution de (X,Y) on a :*

$$\lim_{i\to\infty} \inf_{\Phi\in\mathcal{C}_i} L(\Phi) = L^* \tag{1.7}$$

et Φ^*_n le classificateur choisi parmi toutes les classes par SRM :

$$\tilde{L}_n(\Phi^*_n) \quad = \quad \text{Argmin} \left(\hat{L}_n(\Phi^*_{n,\mathcal{C}_i}) + \sqrt{\frac{\log(4e^8\mathcal{S}(\mathcal{C}_i,n^2)) + i}{2n}}\right)$$

Alors le classificateur par SRM satisfait :

$$\lim_{i\to\infty} L(\Phi^*_n) = L^* \text{ universellement en probabilité}$$

Remarque : Dans le terme de pénalité, on reconnaît d'une part le majorant trouvé en (1.6), et d'autre part le poids $\delta_{\mathcal{C}_i} = i$ rajouté pour la convergence de la somme des probabilités. Un terme en i^k ou en $2\log(i)$ auraient aussi garanti la convergence, le choix de ces poids est donc en partie arbitraire, et il n'y a pas de pondération optimale. Notons simplement que les poids augmentent avec la complexité de la classe, de manière à perturber le moins possible la valeur de la borne supérieure. Une interprétation possible des $e^{-i}/\sum e^{-i}$ est de les comprendre comme une probabilité *a priori* de choisir la classe $\mathcal{C}_i$.

Démonstration. On part de la décomposition suivante :

$$L(\Phi_n^*) - L(\Phi^*) = L(\Phi_n^*) - \tilde{L}_n(\Phi_n^*) + \tilde{L}_n(\Phi_n^*) - L(\Phi^*)$$

Pour le premier terme, on a :

$$\begin{aligned}
\mathbf{P}\left(L(\Phi_n^*) - \tilde{L}_n(\Phi_n^*) > \varepsilon\right) &\leq \mathbf{P}\left(\sup_{i\geq 1}(L(\Phi_{n,\mathcal{C}_i}^*) - \tilde{L}_n(\Phi_{n,\mathcal{C}_i}^*)) > \varepsilon\right) \\
&\leq \mathbf{P}\left(\sup_{i\geq 1}(L(\Phi_{n,\mathcal{C}_i}^*) - \hat{L}_n(\Phi_{n,\mathcal{C}_i}^*) - \sqrt{\frac{\log(4e^8\mathcal{S}(\mathcal{C}_i,n^2)) + i}{2n}}) > \varepsilon\right) \\
&\leq \sum_{i=1}^{\infty}\mathbf{P}\left(L(\Phi_{n,\mathcal{C}_i}^* > \hat{L}_n(\Phi_{n,\mathcal{C}_i}^*) + \sqrt{\frac{\log(4e^8\mathcal{S}(\mathcal{C}_i,n^2)) + i}{2n}} + \varepsilon\right) \\
&\leq \sum_{i=1}^{\infty} 4e^8\mathcal{S}(\mathcal{C}_i,n^2)\exp\left\{-2n\left(\sqrt{\frac{\log(4e^8\mathcal{S}(\mathcal{C}_i,n^2)) + i}{2n}} + \varepsilon\right)^2\right\}
\end{aligned}$$

en utilisant $(a+b)^2 \geq a^2 + b^2$ pour a, b positifs, et en simplifiant par $2n$:

$$\begin{aligned}
\mathbf{P}\left(L(\Phi_n^*) - \tilde{L}_n(\Phi_n^*) > \varepsilon\right) &\leq \sum_{i=1}^{\infty} 4e^8\mathcal{S}(\mathcal{C}_i,n^2)\exp\left\{-\left(\log(4e^8\mathcal{S}(\mathcal{C}_i,n^2)) + i + 2n\varepsilon^2\right)\right\} \\
&\leq e^{-2n\varepsilon^2}\sum_{i=1}^{\infty} e^{-i} \leq e^{-2n\varepsilon^2}
\end{aligned}$$

Ce qui établit la convergence du premier terme. Nous passons maintenant à l'étude du deuxième terme. Notons toute d'abord que d'après (1.7), il existe une classe $\mathcal{C}_k$ telle que $L(\Phi_{\mathcal{C}_k}^*) - L(\Phi^*) < \frac{\varepsilon}{2}$. Il suffit donc de montrer que $\tilde{L}_n(\Phi_n^*) - L(\Phi_{\mathcal{C}_k}^*) < \frac{\varepsilon}{2}$ pour pouvoir conclure. On a :

$$\begin{aligned}
&\mathbf{P}\left(\tilde{L}_n(\Phi_n^*) - L(\Phi_{\mathcal{C}_k}^*) > \frac{\varepsilon}{2}\right) \leq \mathbf{P}\left(\tilde{L}_n(\Phi_{\mathcal{C}_k,n}^*) - L(\Phi_{\mathcal{C}_k}^*) > \frac{\varepsilon}{2}\right) \\
&\leq \mathbf{P}\left(\hat{L}_n(\Phi_{\mathcal{C}_k,n}^*) + \sqrt{\frac{\log(4e^8\mathcal{S}(\mathcal{C}_i,n^2)) + i}{2n}} - L(\Phi_{\mathcal{C}_k}^*) > \frac{\varepsilon}{2}\right)
\end{aligned}$$

Pour n assez grand, la racine est inférieure à $\frac{\varepsilon}{4}$, et on obtient alors :

$$\begin{aligned}
\mathbf{P}\left(\tilde{L}_n(\Phi_n^*) - L(\Phi_{\mathcal{C}_k}^*) > \frac{\varepsilon}{2}\right) &\leq \mathbf{P}\left(\hat{L}_n(\Phi_{\mathcal{C}_k,n}^*) - L(\Phi_{\mathcal{C}_k}^*) > \frac{\varepsilon}{4}\right) \\
&\leq \mathbf{P}\left(\sup_{\Phi\in\mathcal{C}_k}|\hat{L}_n(\Phi) - L(\Phi)| > \frac{\varepsilon}{4}\right) \\
&\leq 4e^8\mathcal{S}(\mathcal{C}_i,n^2)\exp\left(-n\frac{\varepsilon^2}{8}\right)
\end{aligned}$$

Ce qui achève la démonstration. □

Nous avons donc démontré, sous des conditions raisonnables, que la SRM est une règle de décision garantissant la convergence uniforme en probabilité de l'erreur du minimiseur SRM vers l'erreur de Bayes. Nous allons maintenant nous intéresser à la performance de la SRM en terme de risque.

1.3.3 Performances de la SRM

Nous avons vu au chapitre 1.3.1 que si nous avions connaissance de la classe optimale au sens du risque, le risque de l'estimateur ERM serait exactement :

$$R(\Phi^*,\Phi^*_{n,\mathcal{C}_{opt}}) \;=\; \inf_{i\geq 1}\left(R(\Phi^*,\Phi^*_{\mathcal{C}_i}) + R(\Phi^*_{\mathcal{C}_i},\Phi^*_{n,\mathcal{C}_i})\right)$$

A-t-on beaucoup perdu, i.e. le risque est-il beaucoup plus grand lorsque l'on passe de la connaissance au choix de la classe ? Le théorème 1.3.2 montre que le risque de la règle SRM est comparable au risque optimal :

Théorème 1.3.2. *La règle SRM vérifie pour tout n:*

$$R(\Phi^*,\Phi^*_n) \leq \inf_{k\geq 1}\left(R(\Phi^*,\Phi^*_{\mathcal{C}_k}) + \sqrt{\frac{16V_k\log(n)+8(k+11)}{n}}\right) \tag{1.8}$$

Démonstration. On peut remarquer que :

$$R(\Phi^*,\Phi^*_n) = \inf_{k\geq 1}\left(R(\Phi^*,\Phi^*_{\mathcal{C}_k}) + R(\Phi^*_{\mathcal{C}_k},\Phi^*_n)\right)$$

En fixant k, on a :

$$\begin{aligned}
R(\Phi^*_n,\Phi^*_{\mathcal{C}_k})^2 &\leq \mathbb{E}\left\{\left(L(\Phi^*_n)-L(\Phi^*_{\mathcal{C}_k})\right)^2\right\} \quad \text{(Inégalité de Jensen)}\\
&\leq \int_0^\infty \mathbf{P}\left(\left(L(\Phi^*_n)-L(\Phi^*_{\mathcal{C}_k})\right)^2 > t\right)dt\\
&\leq u + \int_u^\infty \mathbf{P}\left(\left(L(\Phi^*_n)-L(\Phi^*_{\mathcal{C}_k})\right)^2 > t\right)dt\\
&\leq u + \int_u^\infty 4e^8\mathcal{S}(\mathcal{C}_i,n^2)e^{-nt/8} + e^{-nt/2}dt
\end{aligned}$$

en redécomposant la probabilité ci-dessus et en reprenant les majorants de la démonstration du théorème 1.3.1. En choisissant $u = 16\dfrac{\log(4e^8\mathcal{S}(\mathcal{C}_i,n^2))+j}{2n}$, on obtient finalement :

$$R(\Phi^*_n,\Phi^*_{\mathcal{C}_k})^2 \;\leq\; \frac{16V_k\log(n)+8(k+11)}{n}\;.$$

□

L'intérêt de ce résultat est double. D'une part nous avons obtenu une garantie sur les performances de la SRM. Un tel résultat n'est pas systématiquement disponible pour toutes les règles. D'autre part, il est possible d'interpréter la borne trouvée : le premier terme de la borne est exactement l'erreur d'approximation de la classe $\mathcal{C}_k$, et le deuxième terme (la

racine) est comparable à la borne obtenue pour l'erreur d'estimation dans la partie précédente ($O(\sqrt{\frac{V\mathcal{C}\log(n)}{n}})$). La règle SRM choisit donc le classificateur faisant le meilleur compromis entre l'erreur d'approximation d'une classe et un terme légèrement plus grand que l'erreur d'estimation. Nous pouvons donc conclure que la borne obtenue pour le risque est comparable à celle que nous aurions obtenue si nous avions eu connaissance de la classe $\mathcal{C}_{opt}$ à l'avance.

1.3.4 Alternatives et extensions de la SRM

L'objectif de la théorie de Vapnik était de proposer une méthode universelle pour trouver un bon classificateur, à partir de la seule donnée d'un échantillon d'entraînement. Cet objectif est atteint, dans le sens où :

- La règle SRM est universellement consistante, et sa vitesse de convergence vers le meilleur classificateur est exponentielle
- Le risque de la règle SRM est comparable au risque que l'on aurait obtenu en sachant par avance "où chercher", c'est-à-dire dans quelle classe se placer pour trouver un bon classificateur.

Le choix du classificateur est basé sur la minimisation d'un critère pénalisé qui met confronte les performances et la complexité de la règle de classification considérée.

La stratégie SRM proposée par Vapnik est un exemple de méthode de sélection de modèles. La sélection de modèles est un domaine de recherche important en statistique depuis la fin des années 70. Akaike (1973) fut le premier à proposer un critère pénalisé de sélection de modèles dans le cadre de la régression. Ce travail est à l'origine de l'ensemble des méthodes de sélection adaptatives. Les méthodes adaptatives sont les méthodes pour lesquelles le modèle optimal est défini en fonction de la quantité d'information disponible sur le processus. Dans le cadre de la stratégie SRM, nous avons vu que la classe optimale est la classe qui réalise le meilleur compromis biais-variance. La stratégie SRM est donc une méthode adaptative. Il existe toutefois d'autres approches pour la sélection de modèles. L'une de ces approches alternatives est présentée en détail en annexe A, où nous introduisons le critère BIC (Schwarz (1978)), ainsi que les méthodes consistantes de sélection de modèles.

Dans le cadre de la classification supervisée, l'approche SRM est à l'origine de la grande majorité des critères pénalisés de sélection de modèles (Boucheron *et al.* (2005)). Toutes ces méthodes sont basées sur l'obtention d'une borne supérieure fine du taux d'erreur réel du classificateur construit. Au chapitre 3, nous présentons une méthode alternative de sélection de modèles appelée Swapping, basée sur l'estimation du risque conditionnel d'un classificateur. Nous montrons en particulier que la méthode Swapping peut être considérée une méthode de pénalisation par covariance, dont la théorie fut très récemment développée par Efron (2004).

Enfin, la théorie de Vapnik propose un cadre très général pour l'obtention de résultats théoriques sur les performances d'une méthode d'apprentissage donnée. Au chapitre 4, nous présentons une adaptation de la théorie de Vapnik au problème de la sélection de variables en classification supervisée. Nous montrons en particulier comment l'ensemble des concepts et des techniques présentés dans ce chapitre peuvent être employés pour garantir les performances des méthodes de sélection de variables.

Chapitre 2

Sélection de variables

2.1 Introduction

La visée de ce chapitre sur la sélection de variables est multiple. Tout d'abord, il a pour vocation de présenter les critères usuellement employés dans la littérature pour classer les méthodes de sélection de variables. Nous présentons ici les différences entre méthodes filter et wrapper d'une part, et entre sélections indépendante et best subset d'autre part. Nous introduisons ensuite en détail une famille de méthodes basées sur les techniques de couvertures de Markov. L'analyse de ces méthodes nous permettra de montrer les limites des méthodes filter, et de motiver notre travail sur les méthodes wrapper présenté au chapitre 4. Enfin, cette première présentation de méthodes de sélection de variables sera l'occasion d'étudier le traitement de la redondance. En effet, les méthodes filter explicitent une hypothèse souvent consensuelle : la redondance est une difficulté supplémentaire pour la sélection de variables et doit être éliminée. Le chapitre 5 de cette thèse est consacrée au traitement de la redondance, et nous prendrons pour point de départ les considérations sur la redondance exposées ici.

Visées de la sélection de variables

Au premier chapitre, nous avons présenté la théorie de l'apprentissage statistique de Vapnik. Nous avons en particulier introduit la stratégie ERM, qui permet de construire un classificateur ϕ_n^* pour prédire le label d'un individu à partir de l'information X disponible pour cet individu. La stratégie ERM est à l'origine de nouvelles méthodes de régularisation, qui ont permis le développement d'algorithmes de classification capables d'éviter le surajustement, même lorsque l'espace de représentation des données est de dimension infinie. Les Support Vector Machines (SVM, Boser *et al.* (1992), Cristianini et Shawe-Taylor (1999), Vapnik (1998)) sont une bonne illustration de ces nouveaux algorithmes, et leur application à des problèmes aussi divers que la classification de caractères manuscrits (Schölkopf et Smola (2002)) ou la classification fonctionnelle de gènes (Brown *et al.* (2000)) a donné des résultats très encourageants.

Bien que les méthodes de classification basées sur la régularisation furent développées pour traiter le problème du surajustement, plusieurs auteurs (Fukumizu *et al.* (2004), Krishnapuram *et al.* (2004a), Xiong *et al.* (2001)) ont fait remarquer qu'une étape de réduction de dimension par sélection de variables permet bien souvent d'améliorer considérablement les performances de classification. La dimension à réduire est ici celle de l'information X. Dans la plupart des applications, X peut être décrit comme une collection de p mesures numériques X^1,...,X^p appelées variables. Dans la stratégie ERM, l'ensemble de ces p variables est utilisé pour la prédiction. Mais il est possible de baser la prédiction sur les seules variables appor-

tant une véritable information sur le label. L'étape de réduction de dimension par sélection de variables consiste alors à identifier un sous-ensemble de variables $S \subset X$ optimal pour la prédiction. Cette sélection de variables semble particulièrement pertinente lorsque l'on est confronté à des problèmes de grande dimension, où la plupart des variables n'apportent pas d'information sur le statut (irrelevant features).

L'objectif de la sélection de variables en classification supervisée est donc de choisir, parmi l'ensemble des variables dont on dispose, un sous-ensemble si possible petit de variables à garder pour construire un classificateur très performant (Dougherty (2001)). La pratique expérimentale a largement démontré que les performances obtenues avec un nombre de variables restreint sont souvent meilleures que celles obtenues avec l'ensemble des variables. La réduction de dimension peut dans certains cas être drastique : Geman *et al.* (2004) proposent une méthode de classification basée sur une unique paire de gènes (tirée à partir de plusieurs milliers). Les performances du classificateur ainsi construit dépassent sur plusieurs exemples les performances des classificateurs construits à partir de l'ensemble des gènes. D'autres auteurs proposent des règles de classification basées sur un nombre de variables très réduit par rapport au nombre de variables initial (Ben-Dor *et al.* (2000), Golub *et al.* (1999),Krishnapuram *et al.* (2004b)).

L'amélioration des performances du classificateur obtenu n'est pas la seule motivation pour sélectionner les variables. La complexité de certains algorithmes de classification dépendant directement du nombre de variables, une réduction de ce nombre permet de réduire considérablement les temps de calcul. Enfin, il est souvent avancé que restreindre le sous-ensemble de variables à un petit nombre devrait permettre de déterminer les variables d'intérêt et donc d'améliorer l'interprétation du problème traité (Blum et Langley (1997), Hall (2000)). Ce dernier objectif est souvent contesté, et certains auteurs ont suggéré que la sélection de variables n'est pas susceptible d'apporter ce type d'information (Michiels *et al.* (2005)). Nous reviendrons sur le problème de l'interprétation des résultats d'une sélection de variables au chapitre 5.

Méthodes Filter - Méthodes Wrapper

Il est aujourd'hui impossible de recenser le nombre de méthodes de sélection de variables proposées dans la littérature. Devant une telle profusion de méthodes, plusieurs auteurs se sont intéressés à la caractérisation de grandes familles de méthodes. En particulier, Kohavi et John (1997) ont introduit la distinction filter / wrapper, qui est aujourd'hui couramment utilisée pour définir les méthodes existantes. D'autres catégories se sont ensuite ajoutées, comme l'ensemble des méthodes intégrées (embedded) qui connaissent depuis quelques années de grands développements. Nous nous intéressons ici à la distinction filter / wrapper. Les méthodes wrapper nécessitent le choix préalable d'une méthode de classification pour être appliquées : la qualité d'un sous-ensemble de variables S est quantifiée par le taux d'erreur obtenu en combinant S avec l'algorithme de classification choisi. On dit alors que la méthode de sélection est dédiée (à une méthode de classification donnée). L'algorithme RFE (Recursive Feature Elimination) développé par Guyon *et al.* (2002) est une bonne illustration des méthodes de sélection wrapper. A l'inverse, les méthodes filter sont celles pour lesquelles le problème de la sélection de variables est traitée indépendamment du problème de la mise au point du classificateur. La méthode filter traditionnellement présentée est la méthode des T-tests (Krishnapuram *et al.* (2004b), Dudoit *et al.* (2002)), qui consiste à faire pour chaque variable un test d'égalité des moyennes entre les deux classes, et à sélectionner les variables pour lesquelles le test est significatif.

Malgré ces deux définitions, la distinction entre méthodes filter et wrapper n'est pas toujours claire pour les auteurs, comme le montrent les deux exemples suivants. Bi *et al.* (2003) considèrent le problème de la sélection de variable pour l'algorithme SVM polynomial. Pour des considérations de temps de calcul, on ne réalise pas la sélection de variables en minimisant le taux d'erreur de l'algorithme SVM polynomial, mais celui de l'algorithme SVM linéaire. On choisit les variables par minimisation du taux d'erreur, mais la règle de classification utilisée pour la sélection de variables n'est pas celle qui sera employée par la suite pour la prédiction. C'est pourquoi Tsamardinos et Aliferis (2003) et d'autres auteurs classent cette stratégie dans les méthodes filter. De la même manière, la méthode des T-tests mentionnée précédemment consiste à sélectionner les variables qui maximisent (une à une) la vraisemblance des données dans le modèle d'analyse discriminante linéaire de Fisher (ADL, Fisher (1936)). En ce sens, on peut considérer cette méthode comme dédiée à l'ADL, bien que l'on ne cherche pas à minimiser le taux d'erreur de l'algorithme. Nous précisons donc ici les définitions de ces deux notions, qui seront ensuite utilisées dans la suite de ce document :

Définition 2.1.1. *On appelle méthode wrapper une méthode de sélection de variables pour laquelle la liste des variables sélectionnées est établie en minimisant le taux d'erreur d'une règle de classification donnée, et ce quel que soit l'usage qui est ensuite fait de cette liste.*

Définition 2.1.2. *On appelle méthode filter toute méthode de sélection de variables pour laquelle la liste des variables n'est pas basée sur la minimisation du taux d'erreur d'une règle de classification.*

Suivant ces définitions, et contrairement à Tsamardinos et Aliferis (2003), nous classerons la méthode de Bi *et al.* (2003) parmi les méthodes wrapper. Remarquons enfin que pour les méthodes wrapper, le pouvoir prédictif d'un sous-ensemble quelconque est toujours définie par le taux d'erreur de la règle de prédiction associée. En revanche, pour les méthodes filter, le pouvoir prédictif doit être défini au cas par cas.

Sélection indépendante - Sélection Best Subset

Une autre manière de distinguer les méthodes de sélection de variables est de déterminer si les variables sont sélectionnées indépendamment les unes des autres ou non. On introduit ainsi les notions de sélection indépendante et de sélection best-subset :

Définition 2.1.3. *La sélection est dite indépendante si le sous-ensemble S est constitué de variables qui, considérées indépendamment les unes des autres, ont un pouvoir prédictif élevé.*

Définition 2.1.4. *La sélection est dite best subset si les variables sélectionnées, considérées conjointement, forment un sous-ensemble de pouvoir prédictif élevé.*

Il est important de préciser que la différenciation indépendante/best subset est distincte de la différenciation filter/wrapper établie au paragraphe précédent. Il existe donc quatre familles différentes de méthodes, correspondant aux quatres cases du tableau 2.1.

Ce point est loin d'être clair dans la littérature, comme l'illustre la présentation de Krishnapuram *et al.* (2004b). Après avoir présenté les méthodes de sélection filter, les auteurs concluent par :

	Indépendante	Best Subset
Filter	T-tests	×
Wrapper	×	RFE

TAB. 2.1 – *Les quatres familles de méthodes de sélection de variables*

The main limitation of methods that select features independently of one another...

On retrouve ici une confusion récurrente dans les articles entre méthodes filter et méthodes indépendantes. Comme le montre le tableau 2.1, les auteurs illustrent généralement les méthodes wrapper par une méthode best subset comme la RFE, et les méthodes filter par une méthode indépendante, comme la sélection par T-tests. Les méthodes filter best subset ne sont jamais présentées, bien qu'elles existent : Klecka (1980) propose une sélection pas-à-pas des variables pour l'ADL, basée sur une procédure de tests. Cette méthode est le strict équivalent de la méthode T-tests en best subset, mais on n'en trouve peu de trace dans la littérature.

La conséquence de la confusion entre méthodes filter et méthodes indépendantes est que beaucoup d'auteurs considèrent la sélection filter comme une simple méthode de préfiltrage des données (Bi *et al.* (2003)), et non comme une classe de méthodes statistiques à part entière. Ceci explique en partie pourquoi les méthodes de sélection filter sont souvent ignorées dans la bibliographie consacrée à la comparaison de méthodes : dans le numéro spécial de Journal of Machine Learning Research consacré à la sélection de variables (Guyon et Elisseeff (2003)), aucune méthode filter ne fut présentée. Pourtant, il existe des arguments en faveur des méthodes filter. Tout d'abord, les méthodes filter sont en général très peu coûteuses en temps de calcul comparées aux méthodes wrapper. Cette considération peut devenir critique lorsque le nombre de variables est grand. Par ailleurs, l'application de méthodes filter à des données réelles montre que ces méthodes améliorent sensiblement les performances des méthodes de classification (Hall (2000)). Enfin le sous-ensemble de variables sélectionnées ne dépend pas par définition d'un algorithme de classification donné. Cet argument est souvent présenté comme un point important pour l'interprétation des données : en santé humaine par exemple, les médecins souhaitent obtenir une liste de facteurs impliqués dans la maladie, et non une liste de facteurs efficaces pour la prédiction avec une méthode de classification donnée.

Les méthodes filter best subset de type "Markov Blanket"

Il demeure que beaucoup de méthodes filter proposées n'ont pour seules justifications que leur caractère intuitif et le faible temps de calcul qu'elles nécessitent. Peu de résultats théoriques sont avancés dans les articles pour établir leur pertinence. Ce n'est pas le cas des méthodes de type Markov Blanket que nous présentons ici, qui sont des méthodes filter qui reposent sur une théorie rigoureuse du point de vue mathématique. Nous présentons cette théorie, et nous montrons en particulier comment la qualité et l'optimalité d'un sous-ensemble S peuvent être définies en l'absence de la spécification d'une méthode de classification. Nous montrons ensuite qu'il existe entre la théorie et l'application de ces méthodes un écart important qui n'est que peu souligné dans les articles. Enfin, l'implémentation des méthodes Markov Blanket nécessite généralement des simplifications de calculs, qui ne sont pas toujours explicitement présentées par les auteurs.

La partie 2.2 expose la théorie sur laquelle reposent les méthodes Markov Blanket. La partie 2.3 clarifie les rapports entre théorie et pratique, et présente un cadre unifié permettant de mieux comprendre la construction de certains algorithmes proposés dans la littérature. Ce cadre est utilisé pour présenter deux algorithmes d'application caractéristiques des méthodes Markov Blanket, et proposés respectivement par Yu et Liu (2004b) et Ding et Peng (2003). Quelques conclusions sur les méthodes filter sont présentées en partie 2.4.

2.2 Cadre théorique des méthodes Markov Blanket

On rappelle que le vecteur d'information X est constitué de p variables X^1,...,X^p. On note S un sous-ensemble de X de taille quelconque. La sélection de variables peut être vue comme la recherche d'un sous-ensemble S de variables contenant toute l'information possible sur la variable Y, et qui soit minimal, au sens où retirer une variable de S dégraderait l'information sur Y. Il faut noter que non seulement un tel sous-ensemble n'est pas unique a priori, mais que sa dimension n'est pas fixée : deux sous-ensembles S et S' peuvent contenir la même information sur Y bien que cette information soit plus condensée dans l'un des deux sous-ensemble. Cette manière de concevoir la sélection de variables n'est donc pas canonique, puisque la taille de S n'est pas considérée.

Est-il possible de proposer une méthode (théorique) de sélection de variables qui nous permette de trouver un sous-ensemble S optimal ? Cette question a été largement traitée dans la littérature (Koller et Sahami (1996), Tsamardinos et Aliferis (2003)), et nous allons voir qu'il existe une méthode de sélection permettant de trouver S. Avant de présenter cette méthode, il nous faut définir plus précisément quels sont les sous-ensembles S éligibles et les caractériser.

2.2.1 Définition du sous-ensemble optimal

Le premier objectif est de quantifier l'information apportée par un sous-ensemble S de variables sur le label Y. Pour cela, on compare l'information portée par S à l'information complète, c'est-à-dire l'information portée par X. Ceci peut être accompli en comparant les distributions conditionnelles $\mathbf{P}(Y = y|X = x)$ et $\mathbf{P}(Y = y|S = s)$. On définit ainsi les sous-ensembles optimaux pour la sélection de variables :

Définition 2.2.1. *Un sous-ensemble S est optimal si $\boldsymbol{P}(Y|S) = \boldsymbol{P}(Y|X)$ et pour tout X^i de S :*

$$\boldsymbol{P}(Y|S^i) \neq \boldsymbol{P}(Y|X)$$

où S^i désigne le sous-ensemble S privé de la variable X^i. Autrement dit, S contient toute l'information sur Y et aucune variable ne peut être retirée de S sans dégrader cette information. On peut caractériser les sous-ensembles S optimaux à l'aide de la notion d'indépendance conditionnelle, définie comme suit :

Définition 2.2.2. *Soient X, Y, Z trois variables aléatoires. les variables Y et Z sont indépendantes conditionnellement à X si et seulement si :*

$$\begin{aligned} \boldsymbol{P}(Y,Z|X) &= \boldsymbol{P}(Y|X)\boldsymbol{P}(Z|X) \\ \Leftrightarrow \boldsymbol{P}(Y|X,Z) &= \boldsymbol{P}(Y|X) \end{aligned}$$

Les sous-ensembles optimaux S sont donc des sous-ensembles tel que $X \setminus S$ est indépendant de Y conditionnellement à S, et pour tout sous-ensemble S' de S, $X \setminus S'$ n'est pas indépendant de Y conditionnellement à S'. Cette caractérisation n'est pas sans intérêt, car elle va permettre de proposer une méthode théorique pour l'obtention d'un sous-ensemble optimal.

2.2.2 Algorithmes backward exacts

A partir de la notion d'indépendance conditionnelle, Koller et Sahami (1996) ont proposé une méthode de sélection backward pour l'obtention d'un sous-ensemble optimal. Nous rappelons tout d'abord la notion d'algorithme backward, puis nous présentons la méthode de Koller et Sahami (1996).

La définition 2.2.1 nous permet de déterminer si un sous-ensemble S donné est optimal ou non. De ce fait, une méthode triviale pour trouver un sous-ensemble de variables optimal consiste à considérer la totalité des 2^p sous-ensembles possibles, et de vérifier pour chacun d'entre s'il est optimal ou non. On parle alors de recherche exhaustive. Dans la plupart des cas, la recherche exhaustive est l'unique stratégie qui garantisse l'identification d'un sous-ensemble optimal. Toutefois, du point de vue algorithmique cette stratégie se révèle généralement impossible car le nombre de sous-ensembles à explorer est trop important. Différents algorithmes de recherche plus parcimonieux peuvent alors être envisagés. Ces algorithmes ne visitent qu'un nombre restreint de sous-ensembles. Leur complexité algorithmique s'en trouve alors considérablement réduite, mais leur emploi ne garantit pas l'identification d'un sous-ensemble optimal.

Les algorithmes de recherche parcimonieux les plus couramment utilisés sont les algorithmes séquentiels. Ces méthodes procèdent par étapes successives, chaque étape consistant à améliorer le pouvoir prédictif du sous-ensemble S constitué à l'étape précédente par l'ajout ou la suppression d'une variable supplémentaire. En particulier, les méthodes backward procèdent par élimination : on commence la recherche en fixant $S = X$, et à chaque étape, la variable X^i dont l'absence affecte le moins le pouvoir prédictif du sous-ensemble $S^i = S \setminus X^i$ est retirée. La complexité algorithmique de cette procédure est p^2, et est donc très inférieure à la complexité algorithmique de la recherche exhaustive.

La méthode de sélection de variables backward proposée par Koller et Sahami (1996) est la suivante :

Algorithme 1.

1. On commence avec $S = X$,
2. on cherche une variable X^i telle que X^i est indépendante de Y conditionnellement à S^i,
3. si une telle variable existe, on met à jour $S = S^i$ et on repart à l'étape 2.

Cette procédure cumule donc les avantages des méthodes exhaustive et séquentielle, puisqu'elle permet l'identification d'un sous-ensemble optimal pour un coût algorithmique d'ordre p^2. S'il existe plusieurs sous-ensemble optimaux, le sous-ensemble identifié dépendra de l'ordre dans lequel les variables sont considérées.

Pour appliquer un tel algorithme, il faut être capable de déterminer quelles sont les variables conditionnellement indépendantes de Y. On définit pour cela un indice synthétique de distance entre deux lois de probabilités. Dans les articles, plusieurs distances sont proposées. Ces distances sont souvent présentées pour des lois discrètes :

• L'information mutuelle (Ding et Peng (2003)) :

$$I(Y,S) = \sum_{y,s} \mathbf{P}(Y = y,S = s) \log \frac{\mathbf{P}(Y = y,S = s)}{\mathbf{P}(Y = y)\mathbf{P}(S = s)}$$

qui n'est autre que la distance de Kullback Leibler entre la loi jointe de Y et S et le produit de leur loi marginale.

• L'incertitude symétrique (Press *et al.* (1988), Yu et Liu (2004a)) :

$$I(Y,S) = 2\left[\frac{H(Y) - H(Y|S)}{H(Y) + H(S)}\right]$$

$$\text{où} \begin{cases} H(Y) = -\sum_y \mathbf{P}(Y = y) \log \mathbf{P}(Y = y) \\ H(Y|S) = -\sum_s \mathbf{P}(S = s) \sum_y \mathbf{P}(Y = y|S = s) \log \mathbf{P}(Y = y|S = s) \end{cases}$$

• L'entropie croisée intégrée (Integrated Cross-Entropy, Koller et Sahami (1996)) :

$$D(Y,S) = \sum_x \mathbf{P}(X = x) \sum_y \mathbf{P}(Y = y,X = x) \log \frac{\mathbf{P}(Y = y|X = x)}{\mathbf{P}(Y = y,S = s)}$$

On remarque que seule la troisième expression définit explicitement un distance entre les lois conditionnelles $\mathbf{P}(Y = y|S = s)$ et $\mathbf{P}(Y = y|X = x)$. Pour les autres définitions, il suffit de constater que plus la dépendance entre Y et S est forte, plus $I(Y,S)$ est élevé. On peut alors considérer $I(Y,X) - I(Y,S)$ comme une distance relative : minimiser cette distance revient à maximiser $I(Y,S)$. Dans la suite, on supposera de manière très générale que l'on dispose soit d'une distance $D(.,.)$ entre lois de probabilité, soit d'une fonction score $I(.,.)$ entre lois.

Il est bien sûr possible de caractériser les sous-ensembles optimaux de variables en fonction de la distance choisie :

Proposition 1. *Un sous-ensemble optimal S vérifie :*

$$D(\boldsymbol{P}(Y|S),\ \boldsymbol{P}(Y|X)) = 0$$

et pour toute variable X^i de S :

$$D(\boldsymbol{P}(Y|S^i),\ \boldsymbol{P}(Y|X)) > 0 \quad .$$

On peut alors récrire l'algorithme de sélection de variables 1 de la manière suivante :

Algorithme 2.

1. On commence avec $S = X$,
2. on cherche une variable X^i telle que

$$D(\mathbf{P}(Y|S),\ \mathbf{P}(Y|S^i)) = 0 \quad ,$$

3. si une telle variable existe, on met à jour $S = S^i$ et on repart à l'étape 2.

Bien que l'introduction d'une distance entre distributions permette de tester l'indépendance conditionnelle entre sous-ensembles de variables, l'algorithme précédent reste théorique : sa réalisation requiert l'estimation des différentes distances. Cette estimation est réalisée à partir d'un échantillon fini de données, ce qui va avoir deux conséquences. La première est que les distances estimées entre variables conditionnellement indépendantes ne seront *a priori* pas nulles. Cette première difficulté peut être contournée en supposant qu'à chaque étape on retire la variable pour laquelle la distance $D(\mathbf{P}(Y|S), \mathbf{P}(Y|S^i))$ est minimale, mais il faut alors définir un critère d'arrêt pour la règle. La deuxième conséquence est qu'il va être difficile d'avoir une bonne estimation de la distance, car la dimension des sous-espaces S et S^i considérés peut être grande. C'est cette dernière difficulté qui motive les différents auteurs à introduire la notion de couverture de Markov comme justification théorique des simplifications envisagées pour le calcul de la distance.

2.2.3 Couverture de Markov

Puisque le problème de l'estimation des distances entre distributions conditionnelles tient à la grande dimension du sous-ensemble S par lequel on conditionne, on peut se demander s'il serait possible de conditionner par un sous-espace S' inclus dans S mais de dimension beaucoup plus petite, telle que les conclusions tirées sur la variable X^i soient identiques. Si un tel sous-espace existe, on pourrait alors se contenter de tester l'indépendance entre Y et X^i conditionnellement à S' pour déterminer si X^i doit être retirée de S. Il serait alors tentant de proposer la procédure suivante :

Algorithme 3.

1. On commence avec $S = X$,
2. On cherche une variable X_i dans S et un sous-ensemble $S' \subset S$ tel que $X_i \notin S'$, et que X_i et Y sont indépendants conditionnellement à S', X_i et Y sont indépendants conditionnellement à S^i,
3. si une telle variable existe, on met à jour $S = S^i$ et on repart à l'étape 2.

Il est clair que dans le cas général l'indépendance d'une variable avec Y conditionnellement à un sous-ensemble de S ne garantit pas l'indépendance conditionnellement à S. Cette procédure n'est valide que sous l'hypothèse que S' est une couverture de Markov pour la variable considérée :

Définition 2.2.3. *Soit X^i une variable de l'ensemble S. Soit S' un sous-ensemble de S ne contenant pas X^i. On dit que S' est une couverture de Markov pour la variable X^i ssi :*

$$\boldsymbol{P}\big((Y,\, S \setminus S') \mid (S', X^i)\big) = \boldsymbol{P}\big((Y,\, S \setminus S') \mid S'\big)$$

Cette condition sur le sous-ensemble S' est plus stricte que l'indépendance conditionnelle : X^i peut être retirée de S à condition que l'on sache trouver un sous-ensemble S' qui apporte la même information que X^i sur Y mais aussi sur les variables restantes $S \setminus S'$. L'idée qu'une variable possédant une couverture de Markov peut être retirée de la liste sans perte d'information est confirmée par le théorème suivant :

Théorème 2.2.1. *Soit X^i une variable de l'ensemble S. Si X^i possède une couverture de Markov dans S^i, alors*

$$D(\boldsymbol{P}(Y|S),\, \boldsymbol{P}(Y|S^i)) = 0$$

On souhaite maintenant simplifier l'algorithme backward initial en utilisant le principe des couvertures de Markov. Malgré la précédente propriété, il n'est pas clair que la procédure backward combinée avec l'astuce des couvertures de Markov soit stable. On peut en effet imaginer qu'à une étape donnée, une variable X^i est retirée de S car elle possède une couverture de Markov S^i, mais qu'au fil des étapes ultérieures les variables composant S^i soient retirées, si bien que si l'on testait à nouveau X^i on ne lui trouverait plus de couverture de Markov. En réalité il n'en est rien, comme le prouve ce théorème :

Théorème 2.2.2. *Soit S le sous-ensemble de variables à une étape donnée de l'algorithme backward. Supposons que la variable X^i ait été précédemment retirée de S. Soit X^j la variable que l'on s'apprête à retirer. Alors X^i possède une couverture de Markov dans S^j.*

Une démonstration des théorèmes 2.2.1 et 2.2.2 peut être trouvée dans Koller et Sahami (1996). L'algorithme backward (1) de sélection de variables est donc strictement équivalente à la suivante :

Algorithme 4.

1. On commence avec $S = X$,
2. on cherche une variable X^i telle que X^i possède une couverture de Markov dans S^i,
3. si une telle variable existe, on met à jour $S = S^i$ et on repart à l'étape 2.

2.2.4 Commentaires

Il est important de noter que le dernier algorithme présenté ainsi que les différentes considérations sur les couvertures de Markov de la partie précédente forment le coeur de la théorie (et de l'argumentation) sur les méthodes Markov Blanket. Pourtant, le raisonnement sur les couvertures de Markov ne pousse pas à l'optimisme. Initialement, on espérait trouver un petit sous-ensemble par lequel conditionner de manière à éviter le problème des grandes dimensions dans l'estimation des distributions conditionnelles. Or nous avons montré que nous ne pouvions conclure sur l'intérêt d'une variable X^i qu'à condition d'établir que celle-ci n'est pas informative sur Y ni sur $S \setminus S'$ conditionnellement à S'. Ainsi, le gain obtenu en conditionnant par S' seulement est perdu puisque l'on considère maintenant la distribution de Y et $S \setminus S'$ conditionnellement à S'. La distribution conditionnellement à la couverture de Markov n'est donc pas (beaucoup) plus simple à calculer que la distribution conditionnelle initiale, le "gain" n'est ici que théorique. Si l'on examine les travaux des différents auteurs employant des méthodes Markov Blanket, on constate qu'en fait aucune des méthodes proposées ne cherche une couverture de Markov pour une variable X^i, mais que toutes cherchent un sous-ensemble S' tel que conditionnellement à S' les variables Y et X^i sont indépendantes. Autrement dit, ces méthodes implémentent l'algorithme (3) et non l'algorithme (4), bien que seule cette dernière soit rigoureuse. Ainsi, la théorie des couvertures de Markov ne joue qu'un rôle de caution scientifique de la procédure employée, mais n'est en réalité jamais implémentée.

Il a déjà été évoqué que les sous-ensembles optimaux ne sont *a priori* pas uniques. Bien qu'en théorie chacun de ces sous-ensembles apporte une information identique sur la réponse Y, la non-unicité du sous-ensemble peut poser problème. D'abord, il est clair que l'ordre dans lequel la procédure backward va considérer les variables va jouer un rôle fondamental, puisqu'en partant de deux variables différentes il est probable que l'on aboutira à deux sous-ensembles optimaux différents. Par ailleurs, il n'est pas évident que tous ces sous-ensembles

soient de même cardinal. Dans ce cas, il est clair que dans un contexte où chaque variable a un coût statistique, lié à l'estimation des paramètres qui lui sont associés dans le cadre paramétrique, nous devrions préférer les sous-ensembles optimaux de taille minimale. L'unicité du sous-ensemble a été étudiée par Tsamardinos et Aliferis (2003), qui ont utilisé la distinction pertinence forte / pertinence faible (strong and weak relevancy) introduite par Kohavi et John (1997) pour déterminer des cas particuliers d'unicité. Nous ne reprenons pas ici leur résultat, mais fondamentalement l'unicité n'est acquise que lorsque chacune des variables inclues dans S apporte une information spécifique, qu'aucune autre variable ne contient. On peut douter que cette situation soit celle des données de grande dimension comme les données de puce à ADN ou de spectrométrie par exemple, souvent décrites comme étant des données très redondantes (Krishnapuram *et al.* (2004b)).

Ces considérations, bien que jamais explicitées dans la littérature sur les méthodes Markov Blanket, ont été implicitement traitées par certains auteurs. En conséquence, il existe un écart important entre la théorie et les algorithmes présentés dans un même article sur les méthodes Markov Blanket. Ce point sur l'approche théorique de ces méthodes était donc nécessaire pour mesurer cet écart, mais aussi pour comprendre sur quels concepts se basent les auteurs pour justifier la construction de leurs algorithmes.

2.3 Le passage à la pratique

2.3.1 Critère de coût

La partie précédente nous a permis de définir et de caractériser les sous-ensembles optimaux, ainsi que de présenter des algorithmes permettant de trouver l'un de ces ensembles. Toutefois, cette seule analyse ne suffit pas à résoudre le problème de la sélection de variables, puisque plusieurs sous-ensembles S peuvent vérifier la condition d'optimalité. Il nous faut donc maintenant définir un critère permettant de choisir parmi tous ces sous-ensembles optimaux. Pour cela, on propose de décrire le problème de sélection de variables de manière plus précise : on désire sélectionner un sous-ensemble S de variables qui soit à la fois parfaitement informatif pour la variable d'intérêt Y et le moins redondant possible.

Le coût d'un sous-ensemble va donc être déterminé par sa redondance. Cette définition du coût lié à la redondance est un simple prolongement de l'algorithme backward : lorsqu'une variable n'apporte pas d'information spécifique on peut la retirer, et on obtient ainsi un nouvel ensemble qui est aussi informatif que le précédent mais moins coûteux, car moins redondant. On peut donc voir la sélection de variables comme étant une "chasse" à la redondance concernant l'information sur Y (Hall (2000), Yu et Liu (2004a), Xing *et al.* (2001)). Définir le sous-ensemble optimal comme étant celui de redondance minimum (sous contrainte d'optimalité au sens de la partie précédente) est donc justifié. On peut remarquer que théoriquement cette condition de redondance minimum ne garantit pas toujours l'unicité du sous-ensemble. Néanmoins dans la grande majorité des cas il n'y aura pas d'ex-aequo. Par ailleurs, le critère choisi ici n'est pas le seul possible, un autre choix aurait entraîné une autre définition du sous-ensemble unique recherché. Nous reposerons le problème des critères alternatifs en partie 2.4.

En reprenant les notations de la partie précédente, on définit le coût d'un sous-ensemble comme étant son degré de redondance :

Définition 2.3.1. *La redondance d'un sous-ensemble S est définie par :*

$$R(S) = \frac{1}{|S|} \sum_{X^i \in S} I(X^i,S^i) \quad ,$$

où $I(.,.)$ est l'une des fonctions score définies en partie 2.2.2. On peut maintenant définir de manière précise le sous-ensemble S optimal au sens de la sélection de variables :

Définition 2.3.2. *Le sous-ensemble S^* optimal pour la classification est le sous-ensemble vérifiant le programme suivant :*

$$S^* = \underset{S}{\text{Argmin}} R(S) \quad s.c. \quad I(Y,S^*) = \max_{S'} I(Y,S') \tag{2.1}$$

L'information apportée par S^* est maximum, et la redondance la plus petite possible. Cette définition est (parfois implicitement) adoptée par différents auteurs (Ding et Peng (2003), Yu et Liu (2004a), Yu et Liu (2004b)), pourtant elle n'est pas strictement égale à la définition proposée initialement : ici, on considère tous les sous-ensembles d'information maximale, y compris ceux qui ne sont pas optimaux au sens de la définition 2.2.1. Du point de vue théorique il y a donc un décalage par rapport à la définition souhaitée du sous-ensemble optimal, mais il n'est pas simple de comprendre quel sera l'impact de ce décalage suivant les algorithmes considérés.

2.3.2 Les algorithmes d'application

La recherche du sous-ensemble optimal au sens de la définition 2.3.2 soulève différentes difficultés techniques, essentiellement liées à la taille des données considérées. Diverses aménagements du problème d'optimisation (2.1) ont donc été envisagées par les auteurs, aboutissant à des versions du problème très simplifiées. Il est alors possible de trouver le sous-ensemble de variables recherché en un temps de calcul raisonnable. Nous présentons ici l'ensemble de ces difficultés ainsi que les choix faits par la majorité des auteurs pour les traiter.

Difficultés d'estimation

La principale difficulté rencontrée pour résoudre le problème (2.1) est le calcul des termes $I(Y,S)$ et $I(X^i, S^i)$. Dans les différentes expressions proposées pour ces distances, $I(.,.)$ est définie comme une somme sur toutes les valeurs possibles de Y et de S. En pratique, on suppose que les variables composant S peuvent être discrétisées : la somme est donc finie. Toutefois, si l'on suppose par exemple que l'on discrétise la distribution de chaque variable X^i en quartile, que S contient k variables et que Y est binaire, le tableau croisé des valeurs de S et Y contient 2×4^k cases. Le nombre de cases vides de ce tableau va donc augmenter très rapidement avec k et l'information $I(Y,S)$ sera très mal évaluée. Dans la plupart des articles (Ding et Peng (2003), Koller et Sahami (1996) et implicitement Yu et Liu (2004b)), la solution apportée à ce problème est d'approcher les quantités $I(Y,S)$ et $I(X^i, S^i)$ respectivement par :

$$\begin{aligned} I(Y,S) &\approx \frac{1}{|S|} \sum_{X^i \in S} I(Y,X^i) \\ I(X^i, S^i) &\approx \frac{1}{|S|} \sum_{X^j \in S^i} I(X^i,X^j) \end{aligned}$$

On remplace ainsi des termes multidimensionnels par des sommes de termes unidimensionnels. D'autres alternatives ont été proposées pour le calcul du terme de redondance. Par exemple, dans Koller et Sahami (1996), les auteurs proposent d'estimer $I(X^i, S^i)$ par $I(X^i,M^i)$, où M^i est composé des K variables de S^i les plus corrélées à X^i. La difficulté de calcul fait que la valeur de K est souvent fixé à 1, ce qui revient au cas précédent, ou à 2.

Il est intéressant de noter que la solution ci-dessus va aussi simplifier grandement les calculs : la qualité d'un sous-ensemble S peut maintenant être établie à partir de la qualité de chacune des variables composant S. Il suffit donc de calculer les p termes $I(Y,X^i)$ pour pouvoir calculer la qualité de n'importe quel sous-ensemble.

Difficultés d'optimisation

Le problème d'optimisation (2.1) tel qu'il est défini ne peut pas être résolu en pratique, car il demande la maximisation (ou la minimisation) de deux critères en même temps. Ce type d'optimisation multi-critères ne peut être résolu que par une solution approchée, en définissant un nouveau critère à optimiser fonction des deux critères initiaux. On peut ainsi reformuler le problème d'optimisation (2.1) de la manière suivante :

$$\max I(Y,S) - \lambda \frac{1}{|S|} \sum_{X^i \in S} I(X^i,S^i) \tag{2.2}$$

où λ est une constante à préciser, qui définit le poids accordé à chacun des deux critères initiaux. Le nouveau critère obtenu peut être interprété comme la recherche du sous-ensemble de variables réalisant le meilleur compromis entre pouvoir prédictif et redondance. Dans la littérature, il n'est pas toujours clair que le nouveau programme (2.2) n'est qu'une approximation du programme initial. De plus, la constante λ est généralement fixée à 1 de manière arbitraire (Ding et Peng (2003)).

Si l'on prend en compte les nouvelles définitions de la qualité et de la redondance, on obtient le programme suivant :

$$\begin{aligned} \max_S \quad & \tfrac{1}{|S|} \quad \sum_{X^i \in S} I(Y,X^i) - \lambda \frac{1}{|S|^2} \sum_{X^i \in S} \sum_{X^j \in S^i} I(X^i,X^j) \\ \Leftrightarrow \max_S \quad & \tfrac{1}{|S|^2} \quad \sum_{X^i \in S} \sum_{X^j \in S^i} \left[I(Y,X^i) - \lambda I(X^i,X^j) \right] \end{aligned} \tag{2.3}$$

Temps de calcul

Le nouveau problème d'optimisation (2.3) est beaucoup plus simple à implémenter que le problème initial. En effet, il suffit maintenant d'estimer les termes $I(Y,X^i)$ et $I(X^i,X^j)$ à partir des données pour avoir à disposition toutes les quantités nécessaires pour la résolution du problème. Toutefois, une recherche exhaustive de la solution nécessite un temps de calcul de l'ordre de $\mathcal{O}(2^p)$, et ne peut donc pas être envisagée la plupart du temps. On emploie alors des méthodes de recherche séquentielles (backward ou forward) pour réduire les temps de calcul. Bien qu'aboutissant à des solutions sous-optimales, ces procédures sont largement employées en sélection de variables (Dudoit *et al.* (2002), Guyon et Elisseeff (2003)) et donnent de bons résultats. Nous présentons ici deux algorithmes pour la résolution du problème d'optimisation (2.3).

Ding et Peng (2003) proposent la procédure forward suivante : on commence par sélectionner la variable $X^{(1)}$ qui maximise $I(Y,X^i)$. Puis on sélectionne les variables suivantes

une par une, en minimisant à chaque fois le critère (2.3). Il est intéressant de constater que la procédure ne nécessite pas de critère d'arrêt : lorsque l'on ne peut plus rentrer de variable sans faire diminuer le critère, la procédure s'arrête. Toutefois, dans leur article, les auteurs fixent un nombre de variables *a priori* et stoppent la procédure de sélection une fois ce nombre atteint. Si le nombre de variables à sélectionner $|S|$ est précisé *a priori*, le temps de calcul de la procédure est de l'ordre de $\mathcal{O}(|S| \times n)$.

L'algorithme de sélection de variables de Yu et Liu (2004b) peut être considéré comme une version sévère de l'algorithme forward traditionnel. La première étape est identique : on commence par sélectionner $X^{(1)}$ maximisant $I(Y,X^i)$. Puis, on retire de l'ensemble des variables $X \setminus \{X^{(1)}\}$ toutes les variables X^j vérifiant :

$$I(Y,X^i) \geq I(Y,X^j) \quad \text{et} \quad I(X^j,X^i) \geq I(Y,X^i) \quad . \tag{2.4}$$

Parmi les variables restantes, on sélectionne à nouveau la variable maximisant $I(Y,X^i)$ et on répète ainsi les procédures de sélection et de défausse des variables jusqu'à ce que toutes les variables de X soient sélectionnées ou disqualifiées. Cette procédure, appelée Approximate Markov Blanket, n'est que peu justifiée par les auteurs qui écrivent [1] :

we heuristically use $[I(Y,X^i)]$ *as a threshold to determine wether the F-correlation* $[I(X^j,X^i)]$ *is strong or not.*

Si l'on considère cette procédure comme une méthode d'approximation du problème d'optimisation (2.3), nous pouvons justifier cette heuristique de la manière suivante : à la première étape, on a sélectionné $X^{(1)}$, et l'on cherche maintenant à sélectionner une nouvelle variable. Si X^j vérifie la condition 2.4, alors le problème d'optimisation (2.3) s'écrit à l'étape 2 (en prenant $\lambda = 1$):

$$\begin{aligned}\sum_{X^i \in S} \sum_{X^j \in S^i} \left[I(Y,X^i) - I(X^i,X^j)\right] \quad &= \quad I(Y,X^{(1)}) - I(X^{(1)},X^{(1)}) + \underbrace{I(Y,X^{(1)}) - I(X^{(1)},X^j)}_{\leq 0} \\ &\quad + \underbrace{I(Y,X^j) - I(X^j,X^{(1)})}_{\leq 0} + \underbrace{I(Y,X^j) - I(X^j,X^j)}_{\leq I(Y,X^{(1)}) - I(X^{(1)},X^{(1)})}\end{aligned}$$

Autrement dit, rajouter X^j ne peut que diminuer la valeur de la somme. Il est alors raisonnable de disqualifier cette variable.

L'heuristique de Yu et Liu (2004b) est radicale et raccourcit beaucoup les temps de calcul. Dès la deuxième étape de sélection de variables, il n'y a plus que $p - k - 1$ variables à considérer, où k est le nombre de variables défaussées durant la première étape. Les temps de calcul observés sont donc généralement très inférieurs à $\mathcal{O}(p^2)$, le temps de calcul d'une procédure forward usuelle. Par ailleurs, il n'y a pas de règle d'arrêt à spécifier : l'algorithme s'arrête lorsqu'il n'y a plus de variable à sélectionner.

2.4 Commentaires

Nous avons vu que l'un des arguments avancés pour l'utilisation des méthodes filter est l'identification d'un sous-ensemble de variables optimal pour la discrimination qui ne dépende pas d'un algorithme de classification particulier. Toutefois, comme le soulignent Tsamardinos

1. Les termes entre crochet indiquent ici un changement de notation par rapport à la notation initiale des auteurs.

et Aliferis (2003), ne pas passer par la spécification d'un algorithme ne signifie en aucun cas que le sous-ensemble de variables trouvé est universel : il dépend forcément d'un certains nombre de choix arbitraires, comme celui de la distance entre distribution (partie 2.2.2) ou de la définition de la redondance d'un sous-ensemble S (partie 2.2.2). Bien qu'aucune étude n'ait été effectuée pour déterminer à quel point la définition du sous-ensemble optimal est sensible à la distance choisie, il n'y a *a priori* pas de raison pour que la liste des variables discriminantes soient plus stables entre différentes méthodes filter qu'entre différentes méthodes wrapper. L'absence de spécification d'un algorithme ne rendra donc pas la liste des variables sélectionnées plus pertinente, ou plus simple à interpréter.

Nous avons vu en partie 2.3.1 que la définition du coût d'un sous-ensemble n'est pas unique, et qu'elle n'est pas non plus dictée *a priori* par notre objectif de sélection de variables. Il est donc possible de modifier toute la partie précédente en considérant un critère de coût alternatif. Il existe au moins un autre critère qui semble raisonnable : ce coût pourrait être défini par la mesure de la redondance d'information sur Y des variables. Il semble même que ce critère soit plus pertinent que celui adopté dans la partie précédente (et dans la majorité des articles), car c'est fondamentalement cette redondance sur Y que les méthodes de type backward présentées dans la première partie de ce chapitre cherchent à éliminer. Ce critère a été récemment adopté par Yu et Liu (2004a), et d'autres auteurs se proposent d'utiliser cette définition alternative dans le futur.

Il est intéressant de constater que les méthodes filter best-subset Markov Blanket définissent le sous-ensemble optimal indépendamment du problème de l'estimation : quelle que soit la quantité d'observations n dont on dispose, il est clair que l'on cherche toujours à déterminer le sous-ensemble optimal défini en partie 2.3.1. On retrouve donc ici une distinction déjà décrite pour les méthodes de sélection de modèles entre critères efficaces et critères consistants (voir annexe A). Les critères de sélection de modèles efficaces cherchent à sélectionner un modèle optimal pour le nombre de d'observations dont on dispose : le modèle optimal change donc en fonction de la quantité d'information disponible. A l'inverse, les critères de sélection de modèles consistants cherche à sélectionner le "vrai" modèle, sans considération de la quantité d'information disponible. En cela les méthodes Markov Blanket présentées ici sont similaires aux critères consistants, au sens où l'on vise un sous-ensemble fixe. Notons toutefois que les méthodes filter ne sont pas par essence des méthodes consistantes, bien qu'il n'existe pas, à notre connaissance, de méthode filter efficace.

Nous avons déjà souligné l'écart important qui existe entre la théorie sur laquelle reposent les méthodes Markov Blanket et l'implémentation de ces méthodes. La théorie est parfaitement rigoureuse du point de vue mathématique, mais ne prend pas en compte l'étape d'estimation. De ce fait le passage à la pratique nécessite des aménagements considérables. Ces aménagements rendent l'étude théorique des algorithmes proposés difficile, ce qui explique qu'il n'existe pas de résultat (de type inégalité oracle ou consistance) garantissant les performances d'une méthode filter. A l'inverse, ce type de résultat est accessible pour les méthodes wrapper, et nous montrons au chapitre 4 un résultat théorique garantissant les performances d'une méthode de sélection wrapper.

Chapitre 3

Swapping

Nous avons vu au chapitre 1 que la théorie de Vapnik repose sur l'étude du risque empirique, noté EER. On s'intéresse alors au biais qui existe entre le risque réel (TER) et le risque empirique lorsque l'on sélectionne un classificateur par minimisation du risque empirique. Les inégalités de concentration permettent de déterminer une borne supérieure en probabilité de ce biais, borne supérieure qui sera par la suite utilisée comme pénalité lors de la sélection de modèles.

Nous proposons ici un point de vue alternatif pour l'étude du risque empirique et de la sélection de modèles. Dans cette partie, on s'intéresse non plus au risque réel, mais au risque conditionnel (CER, pour "conditional error rate" en anglais) du classificateur ϕ_n^* construit à partir des données

$$L_x(\phi_n^*) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{P}(\phi_n^*(X) \neq Y | X = x_i) \quad .$$

Ce risque, souvent appelé *in-sample error* dans la littérature statistique, représente la probabilité de se tromper lorsque l'on cherche à classer un nouvel individu dont l'information x serait identique à celle de l'un des individus de l'échantillon d'entraînement. Intuitivement ce risque devrait être moins élevé que le risque réel, puisque l'on ne s'intéresse qu'à la prédiction en des points déjà observés, i.e. pour lesquels on dispose d'une réalisation dans les données d'apprentissage. Il est alors intéressant de constater que le risque empirique de ϕ_n^*

$$L_n(\phi_n^*) = \frac{1}{n}\sum_{i=1}^{n} I_{\{\phi_n^*(X_i) \neq Y_i\}} \quad ,$$

utilisé précédemment comme estimateur du risque réel, peut aussi être considéré comme un estimateur naturel de $L_x(\phi_n^*)$. Mais ici encore cet estimateur est biaisé, puisque les données servent à la fois pour la construction du classificateur et pour l'estimation du risque conditionnel qui lui est associé, et l'on désire étudier ce biais.

Dans la partie 3.1, nous étudions de manière théorique la distribution de la variable aléatoire

$$B(\Phi_n^*) = L_x(\Phi_n^*) - L_n(\Phi_n^*) \quad .$$

Remarquons dès à présent que dans cette formule, seuls les labels Y_1,...,Y_n sont supposés aléatoires. Les informations x_1,...,x_n sont fixes, puisque l'on s'intéresse au risque conditionnel. Par ailleurs on distinguera ici le classificateur ϕ_n^*, obtenu à partir d'un échantillon donné,

du classificateur Φ_n^*, défini comme étant la règle de classification aléatoire obtenu pour des échantillons ayant pour information x_1,...,x_n, et pour label le vecteur aléatoire Y_1,...,Y_n. Le classificateur ϕ_n^* est donc une réalisation de la variable aléatoire Φ_n^*. Nous commençons par l'étude du cas simple où l'information x_i est identique pour tous les individus de l'échantillon. Nous donnons en particulier la distribution exacte de la variable aléatoire $B(\Phi_n^*)$, ainsi que la valeur exacte du biais du risque empirique :

$$E_Y\left(B(\Phi_n^*)\right) = E_Y\left(L_x(\Phi_n^*) - L_n(\Phi_n^*)\right) \quad . \tag{3.1}$$

La formule du biais est ensuite présentée dans le cas général. A partir du calcul exact du biais, un estimateur non biaisé de $E_Y\left(B(\Phi_n^*)\right)$ est proposé, appelé estimateur Swapping.

Dans la partie 3.2, nous montrons que le calcul du biais (3.1) peut être motivé par l'application de ce résultat à la régularisation en classification supervisée. L'une des clés de la sélection de modèles en classification supervisée est en effet d'estimer précisément le risque réel du classificateur sélectionné dans une classe $\mathcal{C}_k$ (cf. chapitre 1). Lorsque le taux d'erreur conditionnel est proche du taux d'erreur réel, on peut proposer pour le TER l'estimateur suivant :

$$C(\Phi_n^*) = L_n(\Phi_n^*) + S_n \tag{3.2}$$

qui peut alors être utilisé comme critère pénalisé. On appellera ce critère le critère Swapping, et la minimisation de ce critère la méthode Swapping, que l'on désignera dans la suite par (S). Une première étude empirique du comportement de (S) est présentée sur l'exemple des données de Kearns *et al.* (1997), classiquement utilisées pour la comparaison de critères pénalisés en classification supervisée. Nous comparons en particulier les performances obtenues avec (S) à celles obtenues en utilisant la procédure de validation croisée.

Le critère (S) peut être employé pour choisir les paramètres d'une méthode de classification supervisée. Nous présentons l'application de la méthode (S) à l'algorithme des k plus proches voisins, noté kNN dans la suite, dans la partie 3.3. (S) est alors utilisé pour choisir le nombre de voisins k. Nous commençons par montrer que le terme de pénalité S_n peut être calculé en un temps algorithmique réduit et comparable au temps de calcul de la validation croisée. Les kNN et (S) sont ensuite appliqués à différents jeux de données classiques de la littérature, et comparés aux performances obtenues avec la validation croisée. Nous présentons aussi un raffinement de la méthode Swapping pour les kNN, appelé S_0kNN, qui conduit à une estimation du taux d'erreur plus précise que celle obtenue par validation croisée.

Enfin, la partie 3.4 présente quelques conclusions sur la méthode (S), et présente les liens existant entre le Swapping et la théorie de la pénalisation par covariance, récemment développée par Efron (2004) et Tibshirani et Knight (1999).

Ce travail a été présenté à la conférence International Symposium on Applied Stochastic Models and Data Analysis (ASMDA) en 2005, et a reçu à cette occasion le prix scientifique IBM-ASMDA pour la meilleure contribution étudiante. Un article sur le sujet est actuellement en relecture au journal Computational Statistics and Data Analysis.

3.1 Estimation du biais en classification

Les notations de cette partie suivent essentiellement celles du chapitre 1. Nous rappelons toutefois quelques précisions importantes pour la suite. Nous désignons par ϕ_n^* un classificateur fixé, obtenu à partir d'un échantillon particulier. L'astérisque "*" signifie que ce classificateur

a été construit par optimisation d'un critère donné, critère qui peut ne pas être la minimisation du risque empirique, contrairement au chapitre 1. On désigne par Φ_n^* la règle de classification obtenue à partir d'un échantillon ayant les mêmes x_1,...,x_n, mais avec des labels Y_1,...,Y_n aléatoires. Remarquons ici qu'en pratique nous aimerions connaître les performances de ϕ_n^*, puisque c'est le classificateur que nous utiliserons *in fine* pour faire de la prédiction. Toutefois il est difficile d'étudier les performances et les propriétés du classificateur ϕ_n^*, et nous devons nous limiter à l'étude de celles de Φ_n^* à la place.

3.1.1 Calcul exact du biais dans le cas non informatif

Dans cette partie, on suppose que l'information x_i est identique pour tous les individus. Cette information ne peut donc pas servir à prédire le label : on se trouve dans le cas "non informatif". On dispose d'un n-échantillon dans lequel on a observé k labels 1 et $n-k$ labels 0. Par abus de notations, k désigne dans cette partie la variable aléatoire du nombre de 1 dans l'échantillon et sa réalisation. Comme nous ne disposons d'aucune information, la seule stratégie pertinente consiste à classer l'ensemble des points selon le label majoritaire. Le classificateur correspondant est

$$\Phi_n^* = I\left(\frac{k}{n} > \frac{1}{2}\right) \quad .$$

Si $k = n/2$, on peut tirer au sort le classement final. On peut ici donner la distribution exacte de la variable aléatoire $B(\Phi_n^*)$:

Proposition 2.

$$\boldsymbol{P}(B(\Phi_n^*) > c) = F_{n,p}\left(min\left(n(p-c),\frac{n}{2}\right)\right) + 1 - F_{n,p}\left(max\left(n(p+c),\frac{n-1}{2}\right)\right)$$

où

$$F_{n,p}(a) = \sum_{i<a} \boldsymbol{P}\left(B(n,p) = i\right)$$

désigne la fonction de répartition de la loi binomiale $B(n,p)$ et $p = \boldsymbol{P}(Y = 1)$.

Démonstration.

$$B(\Phi_n^*) = \begin{cases} p - k/n & \text{si } k < n/2 \\ k/n - p & \text{si } k \geq n/2 \end{cases} \quad . \tag{3.3}$$

Ainsi

$$\begin{aligned} \mathbf{P}(B(\Phi_n^*) > c) &= \mathbf{P}\left(\left[\left(p - \frac{k}{n} > c\right) \cap \left(k < \frac{n}{2}\right)\right] \cup \left[\left(\frac{k}{n} - p > c\right) \cap \left(k \geq \frac{n}{2}\right)\right]\right) \\ &= \mathbf{P}\left(\left[(k < n(p-c)) \cap \left(k < \frac{n}{2}\right)\right] \cup \left[(k > n(p+c)) \cap \left(k \geq \frac{n}{2}\right)\right]\right) \\ &= \mathbf{P}\left(\left[k < min\left(n(p-c),\frac{n}{2}\right)\right] \cup \left[k > max\left(n(p+c),\frac{n-1}{2}\right)\right]\right) \end{aligned}$$

□

Le biais entre le CER et l'EER est directement obtenue à partir de la proposition précédente :

Proposition 3.

$$E_Y(B(\Phi_n^*)) = 2p(1-p)\boldsymbol{P}\left(U_{n-1} = \left[\frac{n-1}{2}\right]\right)$$

où $U_{n-1} \sim B(n-1,p)$ *et* $[a]$ *désigne la partie entière de* a.

Pour démontrer cette proposition on utilise le lemme suivant :

Lemme 3. *Avec les notations précédentes, on a :*

$$\sum_{i<a} i\boldsymbol{P}(k=i) = npF_{n-1,p}(a-1)$$

Démonstration.

$$\begin{aligned}
\sum_{i<a} i\mathbf{P}(k=i) &= \sum_{i<a} iC_n^i p^i(1-p)^{n-i} \\
&= \sum_{i<a} \frac{in!}{i!(n-i)!} p^i(1-p)^{n-i} \\
&= np\sum_{i<a} \frac{(n-1)!}{(i-1)!(n-i)!} p^{i-1}(1-p)^{n-i} \\
&= npF_{n-1,p}(a-1)
\end{aligned}$$

□

On peut maintenant démontrer la proposition 2 :

Démonstration. On part de l'expression (3.3) de la variable B pour écrire le biais :

$$E(B(\Phi_n^*)) = \sum_{i<\frac{n}{2}} \left(p - \frac{i}{n}\right)\mathbf{P}(k=i) + \sum_{i\geq\frac{n}{2}} \left(\frac{i}{n} - p\right)\mathbf{P}(k=i)$$

Notons que $E(k) = np$ implique

$$\sum_i \left(p - \frac{i}{n}\right)\mathbf{P}(k=i) = 0 \ .$$

Ainsi

$$\sum_{i\geq\frac{n}{2}} \left(\frac{i}{n} - p\right)\mathbf{P}(k=i) = \sum_{i<\frac{n}{2}} \left(p - \frac{i}{n}\right)\mathbf{P}(k=i)$$

et

$$E(B(\Phi_n^*)) = 2\sum_{i<\frac{n}{2}} \left(p - \frac{i}{n}\right)\mathbf{P}(k=i)$$

$$E(B(\Phi_n^*)) = 2pF_{n,p}\left(\frac{n}{2}\right) - \frac{2}{n}\sum_{i<\frac{n}{2}} i\mathbf{P}(k=i)$$

On applique maintenant le lemme cité plus haut :

$$E(B(\Phi_n^*)) = 2p\left[F_{n,p}\left(\frac{n}{2}\right) - F_{n-1,p}\left(\frac{n}{2}-1\right)\right]$$

Soit $A = \left(U_n < \frac{n}{2}\right)$ et $C = \left(U_{n-1} < \frac{n}{2} - 1\right)$ où U_n est la somme de n variables aléatoires suivant une loi de Bernoulli $B(p)$. C est inclus dans A, on a donc

$$\mathbf{P}(A) - \mathbf{P}(C) = \mathbf{P}(A \cap C^c) = \mathbf{P}\left(\left(U_{n-1} = \left[\frac{n-1}{2}\right]\right) \cap (X = 0)\right) \quad ,$$

où $X \sim B(p)$. On obtient finalement :

$$F_{n,p}(\frac{n}{2}) - F_{n-1,p}\left(\frac{n}{2} - 1\right) = \mathbf{P}(A) - \mathbf{P}(C) = (1-p)\mathbf{P}\left(U_{n-1} = \left[\frac{n-1}{2}\right]\right)$$

□

La proposition 3 suggère plusieurs commentaires. Tout d'abord, remarquons que dans le cas non informatif, le risque conditionnel et le risque réel sont identiques. Par ailleurs, il est aisé de constater que le classificateur Φ_n^* défini n'est autre que le minimiseur du risque empirique. La distribution obtenue est donc la distribution de la différence TER-EER classiquement étudié dans la théorie de Vapnik.

Il est par ailleurs intéressant d'interpréter la formule du biais obtenue. Ce biais peut se décomposer en deux parties. Le terme $p(1-p)$ mesure le bruit du problème de classification considéré. L'évènement $U_{n-1} = (n-1)/2$ décrit le cas de la "majorité courte" : si l'on suppose par simplicité que n est impair, cet évènement décrit le cas où il y a exactement autant de 1 que de 0 parmi les $n-1$ observations, et où seule la $n^{\text{ième}}$ observation permet de définir le label majoritaire. On constate que c'est le cas le plus défavorable : la conséquence d'un changement de label de l'une des observations donnerait la règle de classification inverse.

3.1.2 Calcul exact du biais dans le cas général

On suppose maintenant que l'on dispose d'un échantillon (X_i,Y_i), où X_i est un vecteur aléatoire. On note $p_x = \mathbf{P}(Y = 1|X = x)$. Le théorème suivant généralise la formule du biais obtenue dans la partie précédente :

Théorème 3.1.1. *Soit $E_Y(B(\Phi_n^*)) = E_Y\left(L_x(\Phi_n^*) - L_n(\Phi_n^*)\right)$ le biais du risque empirique EER. On a :*

$$E_Y(B(\Phi_n^*)) = \frac{2}{n}\sum_{i=1}^{n} cov\left(I_{\{Y_i=1|X_i=x_i\}}, \Phi_n^*(x_i)\right) \quad .$$

Démonstration.

$$\begin{aligned}
E_Y(B(\Phi_n^*)) = \quad & E_Y\left\{\frac{1}{n}\sum_{i=1}^{n}[(1-p_{x_i})\Phi_n^*(x_i) + p_{x_i}(1-\Phi_n^*(x_i))] - \frac{1}{n}\sum_{i=1}^{n} I_{\{\Phi_n^*(x_i)\neq Y_i\}}\right\} \\
= \quad & E_Y\left\{\frac{1}{n}\sum_{i=1}^{n}[(1-p_{x_i})\Phi_n^*(x_i) + p_{x_i}(1-\Phi_n^*(x_i))]\right. \\
- \quad & \left.\frac{1}{n}\sum_{i=1}^{n}[(1-I_{\{Y_i=1|X_i=x_i\}})\Phi_n^*(x_i) + I_{\{Y_i=1|X_i=x_i\}}(1-\Phi_n^*(x_i))]\right\} \\
= \quad & E_Y\left\{\frac{1}{n}\sum_{i=1}^{n}(I_{\{Y_i=1|X_i=x_i\}} - p_{x_i})(2\Phi_n^*(x_i) - 1)\right\}
\end{aligned}$$

Comme $E(I_{\{Y_i=1|X_i=x_i\}}) = p_{x_i}$ le théorème est démontré. □

Efron (1986) a démontré un résultat similaire dans le cas de la régression logistique, et le cas général est traité dans Hastie *et al.* (2001). La démonstration présentée ici diffère de la preuve initiale d'Efron, et ce résultat sert de point de départ pour le théorème (3.1.2), qui donne une forme plus simple du biais :

Théorème 3.1.2. *Pour toute règle de classification Φ_n^* on a :*

$$E_Y(B(\Phi_n^*)) = \frac{2}{n}\sum_{i=1}^{n} p_{x_i}(1-p_{x_i})E_Y[\Phi_n^*(x_i|Y_i=1) - \Phi_n^*(x_i|Y_i=0)] \ , \tag{3.4}$$

où $\Phi_n^*(.\ |Y_i = y)$ désigne la règle de décision construite à partir de l'échantillon d'entraînement, en fixant le label Y_i de l'observation i à y.

Démonstration. On part du théorème 3.1.1, et on définit $U_n = \frac{2}{n}\sum_{i=1}^{n} U_n^i$, où

$$U_n^i = E_Y\left\{(I_{\{Y_i=1|X_i=x_i\}} - p_{x_i})\Phi_n^*(x_i)\right\} \ .$$

On a :

$$U_n^i \quad = \quad E_{Y(i)}\left[E_{Y_i}\left[(I_{\{Y_i=1|X_i=x_i\}} - p_{x_i})\Phi_n^*(x_i)\right]\right]$$

où $E_{Y(i)}$ désigne l'espérance prise pour les variables aléatoires Y_1,...,Y_{i-1},Y_{i+1}...,Y_n.

$$\begin{aligned} U_n^i &= E_{Y(i)}\left[(1-p_{x_i})\Phi_n^*(x_i|Y_i=1)p_{x_i} + (0-p_{x_i})\Phi_n^*(x_i|Y_i=0)(1-p_{x_i})\right] \\ &= p_{x_i}(1-p_{x_i})E_{Y(i)}\left[\Phi_n^*(x_i|Y_i=1) - \Phi_n^*(x_i|Y_i=0)\right] \ . \end{aligned}$$

On conclut en sommant cette dernière égalité sur x_1,...x_n. □

Ainsi, le biais mesure à quel point le classement en un point x_i est sensible au label observé en ce même point dans les données d'entraînement. On peut alors distinguer deux cas. Si la règle est stable (en espérance), au sens où le classement au point x_i ne dépend pas du label Y_i observé en ce même point dans l'échantillon d'entraînement, alors le changement de label n'entraîne pas de changement de classement (en espérance), et la contribution au biais du point x_i est nulle. A l'inverse, si la règle est versatile et que le changement de label change le classement au point x_i, la contribution de ce point sera de $\pm 2p_{x_i}(1-p_{x_i})$. Comme on le voit, la contribution d'un point au biais peut être négative. Toutefois, cet événement devrait être rare. Le cas où la contribution est négative correspond à la configuration suivante : lorsque x_i est de label 1, le classificateur $\phi_n^*(.\ |Y_i = 1)$ le classe en 0. Puis lorsque l'on change le label de x_i de 1 à 0, le nouveau classificateur $\phi_n^*(.\ |Y_i = 0)$ le classe en 1. Cette configuration où le point est toujours classé à l'envers devrait donc être exceptionnelle. Il est même facile de constater que pour certaines méthodes de classification, comme les kNN par exemple, ce cas est strictement impossible.

3.1.3 Estimation du biais dans le cas général

Nous pouvons déduire du théorème (3.1.2) un estimateur sans biais du biais conditionnel :

Corollaire 1. *Avec les notations du Théorème 3.1.2, un estimateur sans biais de $E_Y(B(\Phi_n^*))$ est :*

$$S_n = \frac{2}{n}\sum_{i=1}^{n} p_{x_i}(1-p_{x_i})[\phi_n^*(x_i|Y_i=1) - \phi_n^*(x_i|Y_i=0)] \ .$$

Cet estimateur est appelé estimateur Swapping.

Analogie avec le $\mathcal{C}_p$ de Mallows

La majeure partie des travaux d'estimation du biais en classification supervisée ont porté sur le biais réel, c'est-à-dire sur le biais de l'EER pour l'estimation du TER. Le biais conditionnel ne fut que rarement envisagé. Il n'en est pas de même en régression, où les premiers travaux d'estimation du biais conditionnel commencent avec Mallows (1973). Ce dernier s'intéresse à l'estimation du biais conditionnel dans le cadre de l'estimation par moindres carrés en régression linéaire gaussienne. Nous reprenons rapidement les résultats obtenus en régression sur le biais conditionnel.

On considère le modèle linéaire suivant :

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon, \quad \varepsilon \hookrightarrow \mathcal{N}(0, \sigma^2\mathbf{I}) \ ,$$

où $\mathbf{Y}$ est un vecteur de taille n, $\mathbf{X}$ est une matrice de taille $n \times p$, et β est un vecteur de paramètres. L'estimateur des moindres carrés classique de $\mu = \mathbf{X}\beta$ est

$$\hat{\mu} = \mathbf{PY}.$$

où $\mathbf{P}$ est la matrice de projection sur l'espace engendré par $\mathbf{X}$. Mallows propose l'estimateur sans biais suivant pour le biais conditionnel :

$$\hat{E}_Y \|\mu - \mathbf{PY}\|^2 = \|\mathbf{Y} - \mathbf{PY}\|^2 + 2\frac{p}{n}\sigma^2 \ ,$$

où $\|.\|$ est la norme euclidienne dans $\mathbb{R}^n$, renormalisée par n :

$$\|A\|^2 = \frac{1}{n}\sum_{i=1}^{n} a_i^2$$

Ce critère est appelé le $\mathcal{C}_p$ de Mallows. Dans le cadre de la régression linéaire en modèle gaussien, ce critère est strictement équivalent au critère d'Akaike (1973).

Plus récemment, l'heuristique de Mallows fut adaptée par différents auteurs, en particulier Ye (1998), au cas de modèles de régression non paramétrique. Ye garde l'hypothèse que le vecteur $\mathbf{Y}$ est gaussien, mais ne fait aucune hypothèse sur la forme de la fonction de régression $f : \mathbf{Y} \to \hat{\mu}$. Ye propose comme nouvel estimateur du biais conditionnel

$$\|\mathbf{Y} - \hat{\mu}\|^2 + 2\frac{\mathrm{D}(f)}{n}\sigma^2 \ ,$$

où

$$\mathrm{D}(f) = \sum_{i=1}^{n} cov\,(\hat{\mu}_i(\mathbf{Y}), y_i - \mu_i)$$

est appelé le degré de liberté généralisé du modèle.Reste alors à estimer les différentes quantités considérées : le degré de liberté généralisé d'une fonction et la variance du bruit dans les données.

L'estimateur Swapping que nous proposons est donc un strict équivalent de l'heuristique de Mallows généralisée dans le cadre de la classification supervisée. On retrouve les mêmes quantités dans le critère Swapping que dans le critère de Mallows généralisé. Le terme

$$\sum_{i=1}^{n} \phi_n^*(x_i|Y_i = 1) - \phi_n^*(x_i|Y_i = 0)$$

est un estimateur du degré de liberté généralisé de l'algorithme de classification considéré. Par ailleurs, nous sommes aussi confrontés à l'estimation du bruit de classification $p_{x_i}(1 - p_{x_i})$, propre à chaque observation. Il nous faut donc estimer ces probabilités afin de disposer d'un estimateur du biais utilisable en pratique. Nous envisageons ici différentes stratégies.

Estimation par plug-in

La plupart des méthodes de classification supervisée implémentées fournissent directement des estimations $\hat{p}_{x_i}$ des probabilités *a posteriori*. On peut alors utiliser directement ces estimations dans la formule de l'estimateur Swapping. Cette méthode a été envisagée par différents auteurs, en particulier par Efron (2004). Toutefois, cette première stratégie peut poser problème. En effet, plus la règle de classification utilisée est complexe, plus les probabilités *a priori* estimées seront proches de 0 ou de 1. Le terme $p_x(1-p_x)$ et par conséquence le biais seront alors proche de 0 pour les classificateurs les plus surajustés. On peut illustrer cet inconvénient en considérant la méthode des k plus proches voisins, avec $k = 1$. Cette valeur du paramètre correspond à l'ajustement maximum de la règle de classification aux données d'entraînement. Or pour $k = 1$, $\hat{p}_{x_i} = 1$ ou 0. Le biais estimé est donc nul pour la règle de classification 1NN.

Majoration du terme de variance

p_x étant compris entre 0 et 1, on a $p_x(1-p_x) < 1/4$. On peut donc remplacer le terme de variance $p_x(1-p_x)$ par cette borne supérieure. On obtient alors un estimateur pessimiste du biais. Le terme de variance étant majoré, la pénalité risque d'être forte, et d'entraîner la sélection d'un modèle sous-ajusté.

Estimateur plug-in robuste

L'estimateur suivant, que nous appelons estimateur plug-in robuste, est un compromis entre les deux stratégies précédentes :

$$\alpha \times \hat{p}_{x_i}(1-\hat{p}_{x_i}) + (1-\alpha) \times \frac{1}{4} \ ,$$

où $\hat{p}_x$ est l'estimateur plug-in, et α est un paramètre à fixer compris entre 0 et 1. Si $\alpha = 1$, on retrouve l'estimateur plug-in, et si $\alpha = 0$, on retrouve la borne supérieure du terme de variance.
Certaines méthodes de classification fournissent des estimations de $\hat{p}_x$ en se basant sur un sous-ensemble de n_x points de l'échantillon d'entraînement. C'est en particulier le cas des méthodes CART et kNN (pour ce dernier algorithme $n_x = k$). Pour ces méthodes, nous proposons une autre version de l'estimateur robuste :

$$\hat{p}_{x,R} = \frac{n_x \hat{p}_x + n_0 \times (1/2)}{n_x + n_0} \ , \tag{3.5}$$

où $\hat{p}_x$ désigne l'estimateur plug-in, et n_0 est un nombre entier à fixer. L'indice "R" désigne le caractère robuste de cet estimateur. On retrouve ici aussi le compromis entre les deux estimateurs proposés plus haut : si n_0 =0, on retrouve l'estimateur plug-in, et si $n = \infty$, on obtient la borne supérieure.

Enfin, notons que l'on n'est pas contraint de réaliser l'estimation des probabilités *a posteriori* avec l'algorithme utilisé pour construire la règle de classification. Ainsi, si l'on désire estimer le biais conditionnel du classificateur SVM ou d'un réseau de neurones, qui ne fournissent pas d'estimation des quantités p_x, on pourra toujours utiliser les estimations plug-in fournies par une autre méthode de classification.

Dans la suite, sauf mention du contraire, nous utiliserons la deuxième version de l'estimateur plug-in robuste. Le choix de la valeur du paramètre n_0 est important, et peut influer sur

la qualité de l'estimateur Swapping. Ce paramètre devrait être choisi en fonction des données : lorsque les données sont fortement bruitées, n_0 devrait être grand, et inversement. Pour les diverses applications présentées ici, nous avons fixé la valeur de ce paramètre à 10. Nous avons pu empiriquement constater que les résultats obtenus sur de nombreux jeux de données avec cette valeur sont satisfaisants, quelle que soit la taille des données et la complexité des modèles envisagés. Les résultats obtenus avec une valeur de n_0 allant de 5 à 15 donnent des résultats comparables (non présentés ici). Nous verrons dans la partie 3.3.4 que dans le cas de l'algorithme des kNN il est possible de se libérer du choix de n_0 en limitant le choix de k aux valeurs strictement supérieures à 1.

3.2 Sélection de modèles par Swapping

3.2.1 Le critère Swapping

Nous avons présenté dans la partie 1 la théorie de Vapnik et la stratégie de minimisation du risque structurel (SRM), qui permet de sélectionner, parmi plusieurs modèles $\mathcal{C}_k$ de complexités différentes, le modèle dans lequel se placer pour construire la règle de décision Φ_n^*. Cette sélection se base sur la minimisation d'un critère pénalisé de la forme

$$\begin{aligned} \phi_n^* &= \underset{k}{\text{Argmin}}\, Crit(\phi_{n,k}^*) \\ &= \underset{k}{\text{Argmin}} \left(L_n(\phi_{n,k}^*) + pen(k) \right) \quad , \end{aligned} \tag{3.6}$$

où $\phi_{n,k}^*$ est le minimiseur du risque empirique de la classe $\mathcal{C}_k$. La pénalité est une fonction croissante de la complexité de la classe $\mathcal{C}_k$, mesurée par la dimension de Vapnik par exemple. Le critère pénalisé est généralement une borne supérieure en probabilité du risque réel.

Nous adoptons ici un autre point de vue pour la sélection de modèles, basé sur la minimisation du risque conditionnel. On suppose maintenant qu'un candidat $\phi_{n,k}^*$ est sélectionné dans chaque classe $\mathcal{C}_k$ par optimisation d'un critère donné. La sélection parmi tous les classificateurs candidats s'opère en minimisant

$$\begin{aligned} C(\phi_{n,k}^*) &= L_n(\phi_{n,k}^*) + S_n \\ &= L_n(\phi_{n,k}^*) + \frac{2}{n} \sum_{i=1}^{n} \hat{p}_{x_i}(1 - \hat{p}_{x_i})[\phi_n^*(x_i|Y_i = 1) - \phi_n^*(x_i|Y_i = 0)] \quad . \end{aligned} \tag{3.7}$$

Bien que cette stratégie soit elle aussi basée sur la minimisation d'un critère pénalisé, la différence avec la SRM tient à l'interprétation de ce critère. Le critère (3.7) est un estimateur du risque conditionnel de la règle de classification, et non une borne supérieure du risque réel.

La justification de la stratégie Swapping (S) est basée sur la décomposition suivante :

$$L(\phi_n^*) = L_n(\phi_n^*) + \underbrace{[L_x(\phi_n^*) - L_n(\phi_n^*)]}_{B(\phi_n^*)} + \underbrace{[L(\phi_n^*) - L_x(\phi_n^*)]}_{A(\phi_n^*)} \quad ,$$

où $B(\phi_n^*)$ est la différence étudiée à la partie précédente. Le terme $A(\phi_n^*)$ dépend d'objets de différentes natures:

- $P(Y = 1|X = x)$, la probabilité *a posteriori* pour des valeurs de X non observées. Pour faire de l'inférence sur cette quantité à partir de l'échantillon d'entraînement, nous devons faire des hypothèses sur la régularité de la fonction $P(Y = 1|X = x)$

- l'échantillon $((X_1,Y_1),...,(X_n,Y_n))$ dont nous disposons, qui peut être plus ou moins représentatif des valeurs possibles de x. On suppose en général qu'il n'y a pas de biais d'échantillonnage.
- le classificateur ϕ_n^* lui-même. En particulier, $A(\phi_n^*)$ peut dépendre de la complexité de la classe d'appartenance de ϕ_n^*. Toutefois, on peut supposer que cette complexité est essentiellement appréhendée dans le terme B, et qu'elle sera donc efficacement prise en compte par le critère Swapping.

Nous faisons donc l'hypothèse que $A(\phi_n^*)$ ne dépend pas de la complexité de ϕ_n^*, ce qui signifie que quel que soit l'ordre de grandeur du terme $A(\phi_n^*)$, ce terme n'est pas pertinent pour la sélection de modèles et peut donc être ignoré. Cette hypothèse semble raisonnable : nous avons déjà remarqué qu'en régression linéaire, les critères d'Akaike et de Mallows sont identiques. Le même critère pénalisé sert donc à estimer à la fois le biais conditionnel et le biais réel. Par ailleurs, plusieurs auteurs ont souligné le fait que l'estimation du biais conditionnel pour la comparaison de modèles est une stratégie pertinente, donnant en pratique des résultats satisfaisants. Ce point est notamment discuté dans Hastie *et al.* (1999), p.203.

3.2.2 Données de Kearns

On se base ici sur le plan de simulation de données de Kearns *et al.* (1997) pour étudier empiriquement les performances de la stratégie (S). Le modèle proposé par les auteurs est le suivant. On divise l'intervalle [0,1] en d intervalles égaux, auxquels on attribue alternativement les labels 0 et 1. Soit $((X_1,Y_1),...(X_n,Y_n))$ un n-échantillon i.i.d. où X_i désigne la position de l'observation i et Y_i son label. Les positions X_i sont tirées uniformément sur [0,1], et cette position tombe dans l'un des d intervalles définis précédemment. L'observation i reçoit pour label Y_i le label de son intervalle d'appartenance avec probabilité $1-\eta$, et le label alternatif avec probabilité η. Le paramètre η mesure le niveau de bruit du problème de classification. Pour tout k, on définit le classificateur $\phi_{n,k}^*$ qui minimise le risque empirique en découpant l'intervalle [0,1] en k segments de tailles quelconques. L'objectif de la sélection de modèles est de choisir le paramètre k. Pour chaque jeu de valeurs des paramètres n et η 50 simulations sont réalisées, et pour chaque simulation le nombre optimal de segments k_{opt} est calculé pour chaque critère de sélection. On calcule ensuite le taux d'erreur estimé et le vrai taux d'erreur de chaque critère. Ces valeurs sont moyennées sur l'ensemble des 50 simulations, et servent à comparer les performances des différents critères.

Dans l'article de Kearns, trois stratégies sont comparées : la méthode SRM de Vapnik, le Minimum Description Length (MDL : Rissanen (1987)) et la validation croisée (CV). Les simulations sont réalisées avec pour nombre d'intervalles $d = 100$. Le nombre d'observations n varie de 200 à 3000 et la valeur de η est prise dans [0.1, 0.2, 0.3]. Les conclusions de cette première étude montrent que les performances obtenues avec les critères SRM et MDL sont souvent très inférieures à celles obtenues avec la validation croisée. Le même plan de simulation est repris dans Bartlett *et al.* (2000), le cas $\eta = 0.4$ étant rajouté aux cas précédents. Les auteurs comparent plusieurs méthodes de pénalisation adaptative à la validation croisée et à la SRM, et concluent que les méthodes de pénalisation adaptative sont plus performantes que les critères pénalisés usuels, et donnent des résultats supérieurs à la validation croisée lorsque le niveau de bruit est élevé. Lorsque le niveau de bruit est faible, (CV) donne de meilleures performances.

Nous reprenons ici le plan de simulation utilisé dans ces deux articles, en y apportant les modifications suivantes :

- Nous avons pu constater qu'en réalisant 50 simulations d'un même jeu de données (i.e.

mêmes valeurs des paramètres η et n) et en calculant le taux d'erreur moyen sur les 50 simulations, les résultats obtenus sont très variables d'une série de simulations à l'autre. Nous avons donc effectué 200 simulations pour les valeurs de n allant de 20 à 100, et 100 simulations pour les valeurs de n allant de 100 à 500.

– Le temps de calcul étant élevé pour ces simulations, nous avons réduit le nombre de segments de d=100 dans les précédents articles à d=10 ici. Plusieurs simulations réalisées avec $d = 100$ nous ont permis de vérifier que les conclusions présentées ici sont stables, et ne sont pas dues à ce changement de paramètre.

3.2.3 Etude empirique du critère (S)

La figure 3.1 montre en ordonnée les différents risques EER, CER et TER, moyennés sur 100 simulations, pour η=0.2 et n=100, en fonction du nombre de segments k, en abscisse. On peut constater que pour les données de Kearns les courbes du CER et du TER sont parallèles, pour $k \geq d$. Ce comportement est observé pour toutes les valeurs possibles de n et η (résultats non présentés ici). On se trouve donc dans le cas où l'hypothèse que la quantité $A(\phi_n^*)$ ne dépend pas de la complexité de ϕ_n^* est vérifiée.

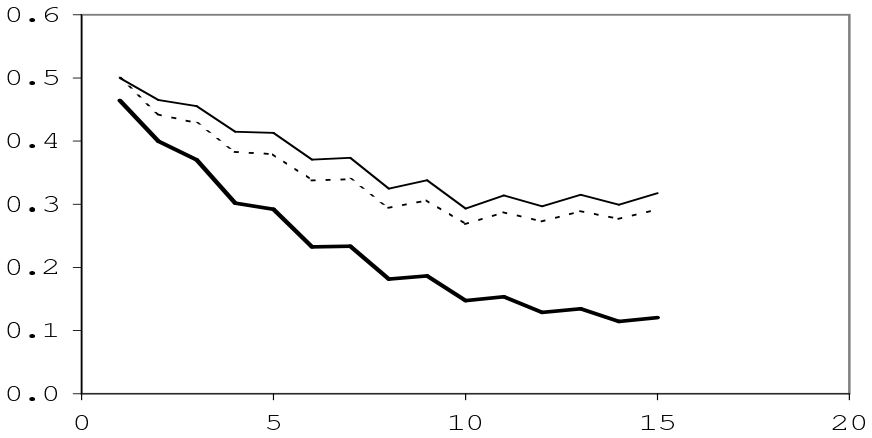


FIG. 3.1 – *EER (trait gras), CER (pointillés) and TER (trait simple) en fonction du nombre d'intervalles k. Les résultats sont moyennés sur 100 simulations, avec* $\eta = 0.2$ *et* $n = 100$.

La figure 3.2 compare le biais réel $E(TER - EER)$, le biais conditionnel et le biais estimé par la méthode (S). On constate que dans cet exemple, où η et fixé à 0.2, le biais est surestimé. Lorsque η=0.1, la surestimation du biais est plus forte. Cette surestimation est négligeable pour η=0.3 et devient une sous-estimation pour η=0.4 (non présenté ici).

La figure 3.3 montre le comportement du risque empirique et du risque pénalisé par Swapping, en fonction du nombre d'intervalles k. A gauche, deux simulations sont représentées séparément, et à droite les courbes correspondent à la moyenne sur 100 échantillons, avec $\eta =$ 0.2 et n=100. On constate que le risque empirique tend vers 0 lorsque le nombre de segments (qui quantifie ici la complexité de la règle de classification) augmente. A l'inverse, le risque pénalisé par Swapping diminue jusqu'à ce que le bon nombre de segments $k = 10$ soit atteint, puis augmente.

Nous nous intéressons maintenant à la comparaison des performances obtenues avec les critères (S) et (CV). Nous donnons aussi ici la performance de l'oracle (O), qui est ici défini

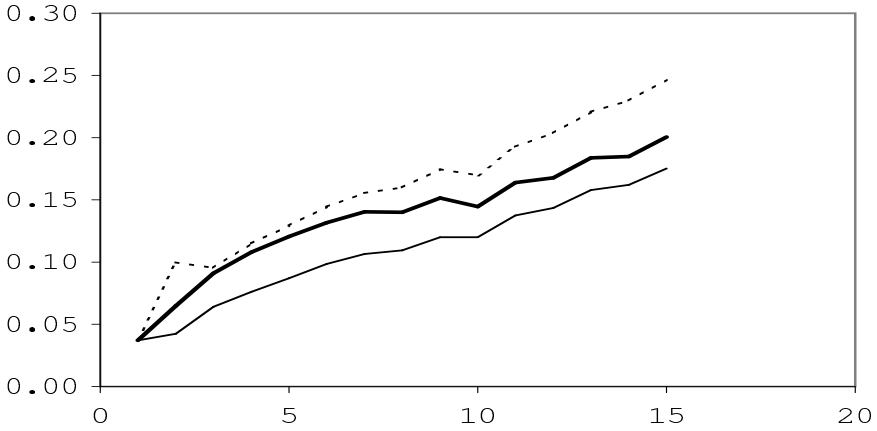


FIG. 3.2 – *Biais estimé par (S) (pointillés), biais conditionnel (trait simple) et biais réel (trait gras) en fonction du nombre d'intervalles k. Les résultats sont moyennés sur 100 simulations, avec $\eta = 0.2$ et $n = 100$. Pour $\eta = 0.2$ le biais conditionnel est surestimé.*

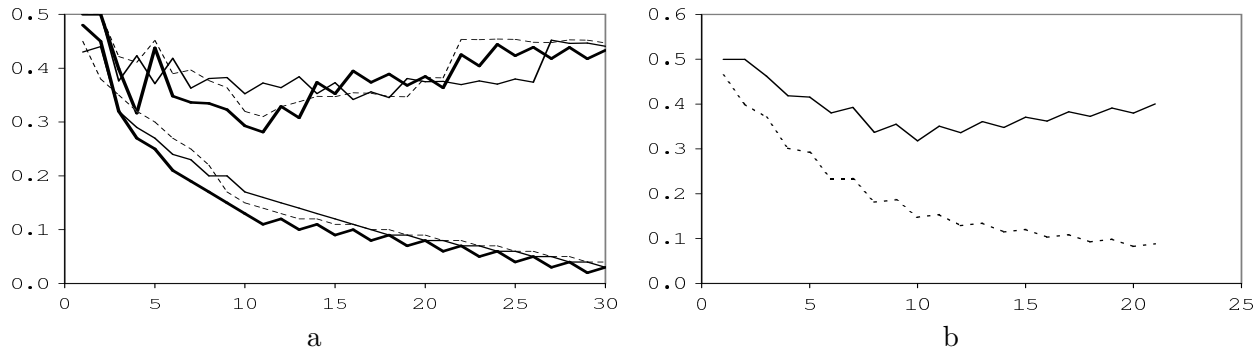


FIG. 3.3 – ***a:*** *EER et estimation par le critère (S) du CER en fonction du nombre d'intervalles k pour deux simulations.* ***b:*** *EER et estimation par le critère (S) du CER en fonction du nombre d'intervalles k, moyenne sur 100 simulations.*

pour chaque échantillon comme étant le classificateur minimisant le risque réel. Notons que ce classificateur optimal n'est pas toujours le classificateur à $k = d$ segments. Le classificateur obtenu par (O) est par définition le meilleur classificateur que l'on puisse sélectionner : les performances de (O) servent donc de référence. Nous présentons ici les résultats obtenus pour η=0.2. A partir des figures 3.4 et 3.5 et des études menées pour les autres valeurs de η (non présentées ici), nous tirons les conclusions conclusions suivantes :

- le critère (S) est plus performant que la validation croisée pour lorsque $\eta \leq 0.3$. On peut quantifier le gain relatif de (S) par rapport à (CV) par la formule $100(L_{CV} - L_S)/(L_{CV} - L_O)$. Lorsque cette quantité est positive (S) vainc (CV). Sur la figure 3.4 sont représentées les 4 courbes de gain relatif correspondant aux 4 valeurs de η considérées, tracées en fonction du nombre d'observations n. On constate que pour les valeurs de $\eta \leq 0.3$, le gain varie entre 20% et 80%. Lorsque $\eta = 0.4$, le gain est faible ou nul.
- La figure 3.5b montre la courbe moyenne de l'oracle, du TER et du TER estimé pour

(S), ainsi que la courbe moyen du TER et du TER estimé pour (CV). L'écart entre l'estimation du TER et le TER obtenu pour le classificateur sélectionné est plus faible pour (S) que pour (CV). L'estimation du TER par (S) peut être optimiste ou pessimiste, suivant les valeurs de η, alors que l'estimation par (CV) est toujours optimiste.

– La figure 3.5c montre les quantiles à 95% des écarts $L_S - L_O$ et $L_{CV} - L_O$, où L_C représente le TER obtenu pour le classificateur sélectionné avec le critère C. Le (S)-quantile est systématiquement plus faible que le (CV)-quantile.

La méthode Swapping est donc un bon compétiteur de la validation croisée. En particulier, l'estimation du TER par (S) est généralement plus précise que celle obtenue par (CV), ce qui est un point important : Bartlett *et al.* (2000) fait remarquer que "Good error estimation procedures provide good model selection methods".

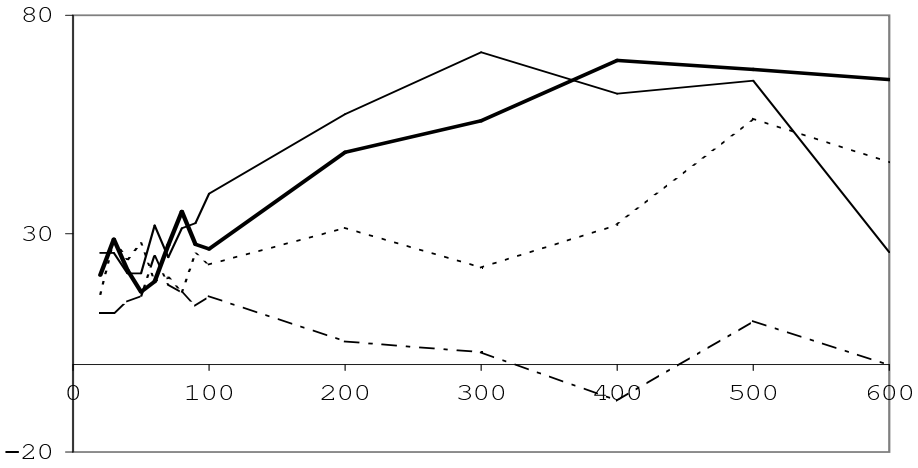


FIG. 3.4 – *Gain relatif de la procédure (S) par rapport à la procédure (CV), mesuré par* $(100(L_{CV} - L_S)/(L_{CV} - L_O)$*, pour* $\eta = 0.1$ *(trait solide),* $\eta = 0.2$ *(trait gras),* $\eta = 0.3$ *(pointillé),* $\eta = 0.4$ *(ligne brisée).*

3.3 Application à l'algorithme kNN

Nous nous intéressons maintenant à l'application du critère Swapping à l'algorithme des k plus proches voisins, noté kNN pour k Nearest Neighbors (Fix et Hodges (1991a), Fix et Hodges (1991b)). Cet algorithme est couramment employé dans divers champs disciplinaires, et bien que très simple, plusieurs analyses ont montré les bonnes performances de généralisation des kNN (Devroye *et al.* (1996), Hastie *et al.* (2001)). Nous rappelons brièvement le principe de l'algorithme :

– On cherche à prédire le label d'un point x_0.

– On détermine les k points les plus proches de x_0 parmi les points de l'échantillon d'entraînement.

– On attribue au point x_0 le label majoritaire parmi ses k plus proches voisins. C'est étape de "vote".

Diverses modifications peuvent être apportées à l'algorithme (pondération des votes, distances adaptatives...) mais fondamentalement les deux paramètres à fixer pour l'algorithme sont la distance utilisée et le nombre de voisins k. Dans ce chapitre, nous supposerons que la

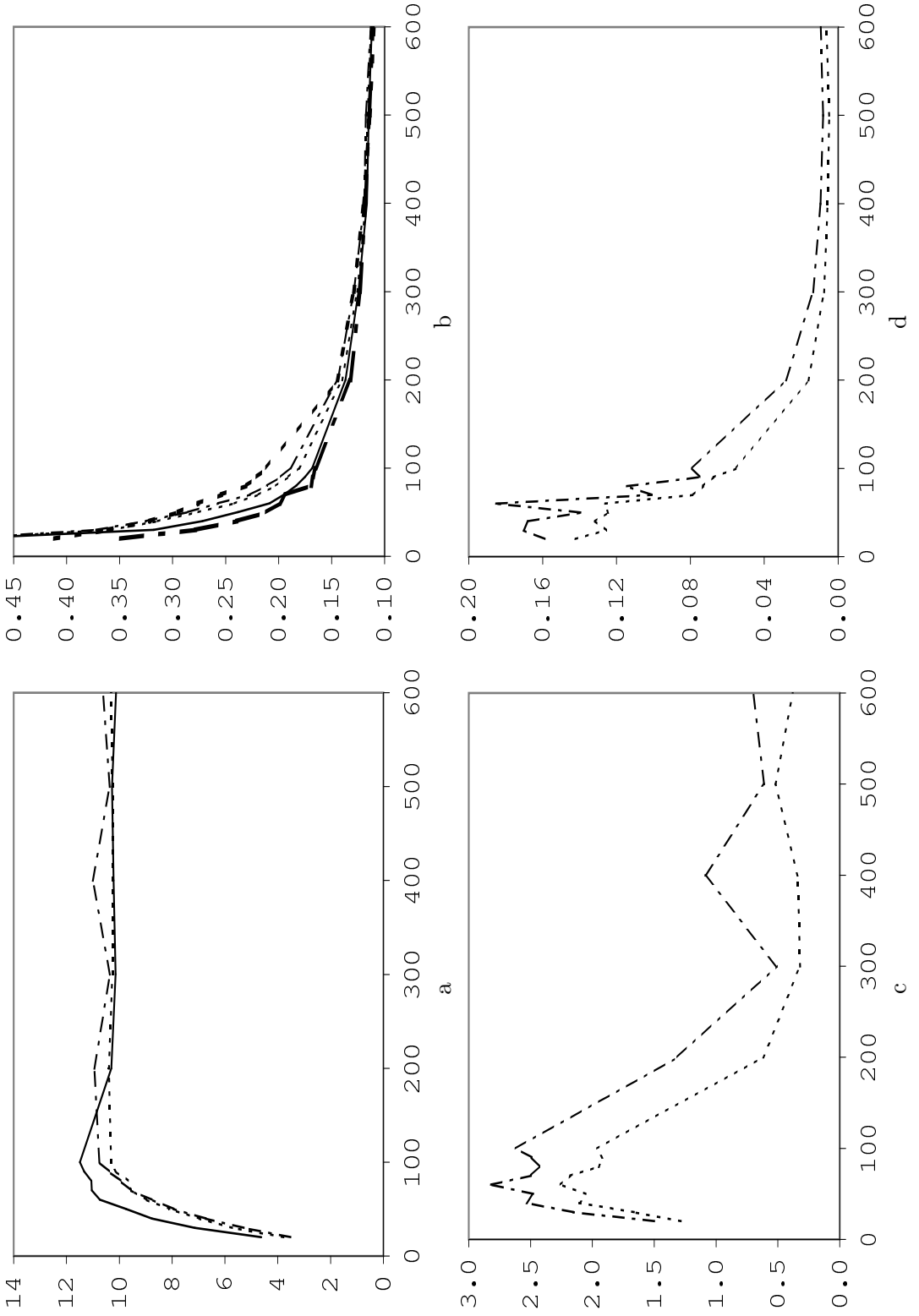


FIG. 3.5 – *Performances de la stratégie (S) pour $\eta = 0.2$.* ***a:*** *Nombre moyen d'intervalles k_O, k_{CV} et k_S obtenus avec (O), (CV) et (S) respectivement.* ***b:*** *Taux d'erreur réel moyen des classificateurs sélectionnés par stratégie la (O) (trait solide), (CV) (ligne brisée) et (S) (pointillé), respectivement. Les traits gras correspondent au TER de (CV) estimé par (CV), et au TER de (S) estimé par (S).* ***c:*** *Moyenne de $|k_O - k_{CV}|$ et $|k_O - k_S|$.* ***d:*** *Quantiles à 95% des distributions de $L_{CV} - L_O$ et $L_S - L_O$.*

distance est la distance euclidienne canonique, appliquée aux données centrées réduites. On ne s'intéresse donc qu'au choix de k.

Le paramètre k est classiquement fixé par validation croisée. L'utilisation de la validation croisée est particulièrement indiquée pour les kNN car son coût algorithmique, usuellement élevé, est ici du même ordre que le coût algorithmique du calcul du risque empirique. D'autres méthodes ont été proposées pour choisir k, en particulier plusieurs critères pénalisés comme AIC, BIC ou la SRM (Cherkassky et Ma (2003)). Dans toutes ces méthodes, le terme de pénalité ne dépend que de la complexité du modèle, c'est-à-dire ici de k. Nous avons déjà vu dans la partie précédente que ces critères ne sont pas aussi performants que la méthode (CV), qui est plus adaptative aux données. Ce comportement est confirmé par Hastie *et al.* (2001) dans le cas des kNN, où les auteurs soulignent en particulier la bonne estimation du taux d'erreur obtenu avec (CV). Ces différentes considérations montrent l'absence de méthodes concurrentes à (CV) pour le choix du nombre de voisins.

On souhaite maintenant comparer les méthodes (CV) et (S) pour le choix de k. Nous commençons par montrer que le coût algorithmique du Swapping n'est pas plus élevé que celui de (CV). Nous étudions ensuite les performances des deux méthodes sur des données simulées ainsi que sur plusieurs jeux de données réelles classiquement utilisés pour la comparaison de méthodes.

3.3.1 Calcul de la pénalité pour (S)

Afin d'éviter les cas d'égalité qui peuvent apparaître lorsque le nombre de voisins considérés est pair, nous suivons l'exemple de Fort et Lambert-Lacroix (2004) en ne considérant que les valeurs impaires de k. La méthode (S) nécessite, pour une valeur de k fixée, de calculer le terme de pénalité :

$$\frac{2}{n}\sum_{i=1}^{n} p_{x_i}(1-p_{x_i})[\phi_n^*(x_i,1)-\phi_n^*(x_i,0)] \ .$$

Nous montrons maintenant que dans le cas particulier de l'algorithme des kNN, le calcul de cette pénalité peut être réalisé en un temps réduit.

On s'intéresse au calcul de la différence $\phi_n^*(x_i,1)-\phi_n^*(x_i,0)$ en un point x_i. On note m le nombre de 1 parmi les k voisins du point considéré. La différence $\phi_n^*(x_i,1)-\phi_n^*(x_i,0)$ peut être facilement calculée en considérant l'argument suivant : lorsque l'on change le label du point x_i, on ne change pas la prédiction qui est faite en ce point, excepté dans le cas où x_i a pour label le label majoritaire et que la majorité est courte. On appelle majorité courte le cas où $m=(k-1)/2$ ou $m=(k+1)/2$ (rappelons que k est impair). En conséquence, le terme de différence prend la valeur 1 pour tous les points tels que $m=(k-1)/2$ ou $m=(k+1)/2$, et 0 pour l'ensemble des autres points. On peut donc facilement trouver l'ensemble des points participant à la pénalité, que l'on appellera par la suite "points limites", sans avoir à effectuer le changement de label et à relancer l'algorithme.

La figure 3.6 illustre l'identification des points limites. Le problème de classification consiste à classer en noir (1) ou en rouge (0), et l'on considère l'algorithme des 3 plus proches voisins. Les points limites sont les points cerclés. Il est intéressant de voir que ces points sont situés à la frontière entre les deux classes. Il est ainsi possible de collecter de l'information et de gagner en interprétation sur le problème de classification traité en étudiant ces points limites. Une discussion plus détaillée sur les points limites est présentée dans la partie 3.4.

Il reste à estimer les variances $p_{x_i}(1-p_{x_i})$. Remarquons qu'il suffit de calculer ces variances pour les points limites, puisque seuls ces derniers participent à la pénalisation. L'estimation

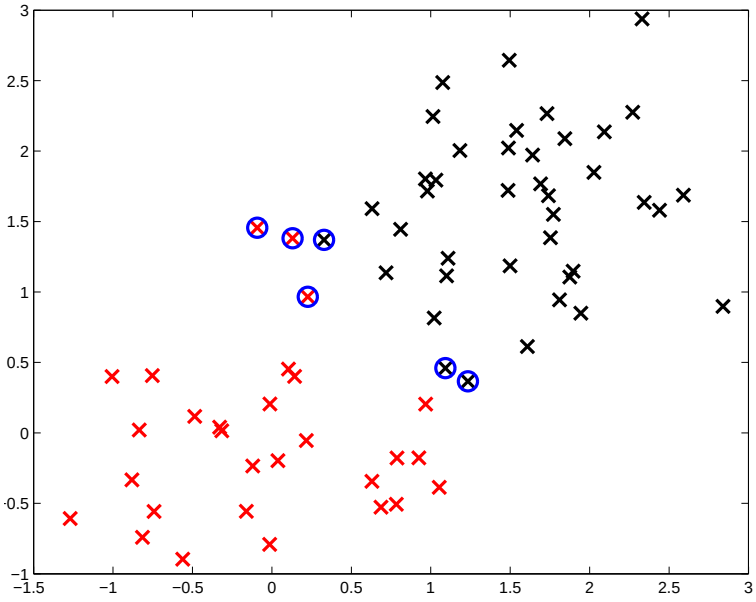


FIG. 3.6 – *Un exemple de calcul de la pénalisation Swapping. Problème à deux classes (rouge et noir), k =3. Les points cerclés sont les points limites.*

des probabilités *a posteriori* p_{x_i} peut être réalisée à l'aide de l'estimateur plug-in robuste proposé en partie 3.1.3. En un point quelconque, l'estimateur plug-in de l'algorithme kNN est m/k, l'estimateur plug-in robuste s'écrit donc :

$$\hat{p}_{x_i,R} = \frac{k \times (m/k) + n_0 \times 1/2}{k + n_0} = \frac{m + n_0/2}{k + n_0} \quad ,$$

et la variance est estimée par $\hat{p}_{x_i,R}(1 - \hat{p}_{x_i,R})$. Les calculs montrent que la variance estimée est identique pour tous les points limites et vaut

$$\hat{p}_{x_i,R}(1 - \hat{p}_{x_i,R}) = \frac{(k + n_0 + 1)(k + n_0 - 1)}{4(k + n_0)^2} \quad .$$

Ainsi, pour les kNN, le terme de pénalité peut s'écrire :

$$\frac{2}{n} \sum_{i=1}^{n} p_{x_i}(1 - p_{x_i})[\phi_n^*(x_i,1) - \phi_n^*(x_i,0)] = \frac{2N_\ell}{n} \times \frac{(k + n_0 + 1)(k + n_0 - 1)}{4(k + n_0)^2} \quad ,$$

où N_ℓ est le nombre de points limites. L'ensemble de ces considérations montre que la méthode Swapping est facilement implémentable pour les kNN, et que son coût algorithmique est comparable à celui de la validation croisée.

3.3.2 Données Simulées

Afin de comparer les résultats obtenus par (CV) et par (S), nous devons étudier le comportement de ces deux méthodes appliquées aux kNN et déterminer les paramètres qui peuvent influencer leurs performances respectives. Pour cela, nous avons simulé des jeux de données de n observations, décrites par p variables et le label 0 ou 1. Chaque variable est aléatoirement

distribuée selon une loi normale $\mathcal{N}(\mu,\sigma^2)$, où $\mu = 1$ pour les observations de label 1 et $\mu = 0$ pour les observations de label 0. Toutes les variables peuvent donc être également utilisées pour le problème de classification. Les valeurs possibles des différents paramètres sont :

- n=50, 100, 200
- p=5, 10, 20
- σ = 0.5, 1, 2.

Pour chaque valeur des paramètres, 100 jeux de données sont simulés, et le paramètre k est choisi par (CV) et par (S). Le taux d'erreur réel est ensuite estimé sur un échantillon indépendant de 10.000 observations. La table 3.1 montre les résultats obtenus, pour n=100. Les résultats obtenus pour n=50 et n=200 sont similaires (non présentés ici).

Paramètres		Swapping		Cross-Valid.	
p	Bruit	k	R	k	R
5	0.5	37.44	**1.49**	37.32	1.53
5	1	24.96	**15.32**	28.82	15.34
5	2	20.44	34.62	27.20	**33.36**
10	1	44.22	**0.08**	46.94	0.09
10	0.5	26.82	**7.38**	27.86	7.52
10	2	22.84	27.89	27.78	**27.37**
20	0.5	46.62	**0**	49.00	**0**
20	1	30.12	**2.1**	38.20	2.14
20	2	26.86	19.06	28.54	**19.05**

TAB. 3.1 – *Résultats sur données simulées, moyennés sur 100 échantillons de 100 observations. Pour chaque méthode (S) et (CV) et pour chaque valeur des paramètres, le nombre moyen de voisins k ainsi que le taux d'erreur réel R sont reportés. Le meilleur taux d'erreur obtenu est signalé en gras.*

Les conclusions sont comparables aux conclusions tirées sur les données de Kearns. Pour de faibles valeurs de σ, (S) donne de meilleures performances que (CV). On remarque que les écarts observés entre les deux méthodes sont toutefois plus petits que sur les données de Kearns.

3.3.3 Données réelles

Nous considérons maintenant divers jeux de données issus de la littérature. Nous donnons un bref descriptif de ces différents jeux ainsi que leur provenance. Tous les jeux de données considérés sont des problèmes de classification à deux classes, et les variables explicatives sont toujours considérées comme étant quantitatives. Mis à part le jeu de données Colon, tous les jeux de données sont disponibles sur la base de jeux de données UCI :

UCI Machine Learning Repository : *http : //www.ics.uci.edu/ mlearn/MLRepository.html*

1. **Cleveland** : 297 observations, 14 variables explicatives. L'objectif est de détecter la susceptibilité d'avoir une attaque cardiaque (classe 1). Données fournis par Andras Janosi, Hungarian Institute of Cardiology, Budapest.
2. **Ionosphere** : 351 radars, 34 variables explicatives. L'objectif est de classer les radars suivant leur capacité à détecter une structure dans la ionosphère. Les "bons" radars (classe 1) détectent la structure.

3. **Pima** : 768 individus, 8 variables explicatives. L'objectif est de diagnostiquer le diabète (classe 1) chez chacun des individus.
4. **Wisconsin** : 683 individus, 10 variables explicatives. L'objectif est de diagnostiquer le cancer du sein (classe 1).
5. **Colon** : 62 individus, 1991 variables explicatives. L'objectif est de diagnostiquer le cancer du colon (classe 1). Ce jeu de données initialement proposé par Alon *et al.* (1999) fait partie des données de biopuces classiquement utilisées en benchmarking. Le nombre de variables annoncé ici n'est pas le nombre de variables usuellement reporté pour ce jeux de données, généralement décrit comme ayant 2000 variables. En réalité, 3 de ces variables sont présentes en quadruple exemplaires parfaitement identiques, il faut donc retirer 3 de ces exemplaires de la table initiale. De plus, le nombre de variables étant important comparé au nombre d'observations, on réalise la classification soit sur l'intégralité des données, soit en ne conservant que 100, 500 ou 1000 variables. Ces variables sont sélectionnées suivant la significativité de leur statistique de test, lorsque l'on réalise un test de Student d'égalité des moyennes entre les populations 1 et 0. Ces données sont disponibles à l'adresse suivante :

$$http : //www.molbio.princeton.edu/colondata$$

La plupart de ces jeux de données nécessitent un pré-traitement avant d'être utilisés. Nous avons réalisé ces pré-traitements en suivant les recommandations des précédents utilisateurs. Nous ne détaillons pas ici l'ensemble de ces procédures, le lecteur pourra se reporter au descriptif des différents jeux de données disponible sur le site du UCI, et à Dudoit *et al.* (2002) pour le jeu de données Colon.

La performance des méthodes (CV) et (S) est évaluée par reéchantillonnage : pour chaque jeu de données étudié, 500 découpages des données en données d'apprentissage (90% des données pour Cleveland, Ionosphere, Pima et Wisconsin, et 70% des données pour Colon) et données test (30% des données pour Colon, 10% pour les autres) sont réalisés. La règle de classification est estimée sur les données d'apprentissage et évaluée sur les données test. Les 500 taux d'erreur ainsi obtenus sont moyennés.

La table 3.2 présente les résultat pour Cleveland, Ionosphere et Wisconsin. Les résultats pour Colon sont présentés dans la table 3.3. Les deux méthodes donnent des résultats comparables du point de vue des performances. Pour les données Ionosphere et Colon avec 1000 gènes, les performances de (S) sont moins bonnes que celles de (CV). Cela n'est pas surprenant pour les données Colon : nous avons précédemment montré sur les études de simulations que la validation croisée donne de meilleurs résultats que le Swapping lorsque le niveau de bruit est élevé. C'est le cas pour les données de biopuces. D'une part les données sont connues pour être entachées d'une forte variabilité technique, et d'autre part le nombre de gènes retenus ici pour la classification est élevé, et il est très vraisemblable que seule une faible proportion de ces gènes soit pertinente pour la classification, le reste des gènes pouvant alors être considéré comme une source supplémentaire de bruit. Les moins bonnes performances de (S) pour les données Ionosphere ont une toute autre explication. Plusieurs auteurs ont avancé que la valeur optimale du nombre de voisins est 1 pour ce jeu de données (Hechenbichler et Schliep (2004)). C'est effectivement la valeur choisie par (CV) dans la plupart des rééchantillonnages. En revanche, il est impossible de choisir 1 avec le critère (S). En effet, lorsque $k = 1$ toutes les observations de l'échantillon deviennent des points limites participant à la pénalité, qui est alors de l'ordre de 25%. Le critère (S) amène alors à choisir la plus petite valeur de k après 1, c'est-à-dire ici 3 (on ne considère que les k impairs). Cet exemple montre une des limites du critère (S). Notons toutefois que cette limite n'est pas très contraignante : la majorité des

	Swapping			Validation croisée		
Données	k	$\widehat{TER}$	TER	k	$\widehat{TER}$	TER
Cleveland	40.5	15.3	**16.4**	40.6	14.9	**16.4**
Wisconsin	19.3	2.86	**3.42**	19.4	2.77	3.45
Ionosphere	3	14.4	15.5	1.28	13.4	**13.7**

TAB. 3.2 – *Comparaison (S) et (CV) sur les jeux de données UCI. Pour chaque méthode (S) et (CV), le nombre moyen de voisins k et les taux d'erreur estimé $\widehat{TER}$ et réel TER sont reportés. Le meilleur taux d'erreur obtenu est signalé en gras.*

	Oracle		Swapping			CV		
Nb. Genes	k	TER	k	$\widehat{TER}$	TER	k	$\widehat{TER}$	TER
1000	6.5	14.3	12.0	24.8	21.8	11.5	12.5	**21.3**
500	6.7	13.4	11.8	23.6	**18.4**	15.2	10.3	19.0
100	4.8	12.4	11.9	22.5	**16.1**	19.2	9.0	16.3

TAB. 3.3 – *Comparaison (S) et (CV) sur le jeu de données Colon, pour différents nombre de gènes sélectionnés. Pour chaque méthode (S) et (CV), le nombre moyen de voisins k et les taux d'erreur estimé $\widehat{TER}$ et réel TER sont reportés. Le meilleur taux d'erreur obtenu est signalé en gras.*

problèmes de classification apparaissant dans la littérature sont des problèmes complexes, où le choix de la valeur 1 entraînerait un surajustement aux données.

Bien que les résultats entre les deux méthodes soient très comparables du point de vue du taux d'erreur réel, de réelles différences apparaissent lorsque l'on considère l'estimation du taux d'erreur. En effet, quel que soit le critère donnant les meilleures performances, la meilleure estimation du TER est systématiquement obtenue avec le Swapping. Sur les données Colon avec 1000 gènes par exemple, l'écart observé entre le TER et son estimation pour la validation croisée est de 10%, alors que pour ce même exemple l'écart est de 3% pour le Swapping. Le critère (S), bien qu'il soit construit comme un estimateur du CER, peut donc être considéré comme un bon estimateur du TER.

3.3.4 Une variante de la pénalité : l'algorithme S_0kNN

Nous avons vu à la partie précédente que dans le cas de l'algorithme kNN, le terme de pénalité de la méthode Swapping peut s'écrire

$$\frac{2N_\ell}{n} \times \frac{(k+n_0+1)(k+n_0-1)}{4(k+n_0)^2}$$

où N_ℓ le nombre de points limites. Le calcul de cette pénalité est très simple, mais nécessite le choix d'un paramètre de régularisation n_0. Ce paramètre intervient dans le calcul de la variance du bruit de classification, et a pour fonction de corriger l'estimateur plug-in. En effet, cet estimateur peut être considérablement biaisé en certains points de l'échantillon d'entraînement.

En réalité, les seuls points intervenants dans la pénalité étant les points limites, nous devons nous intéresser au comportement de l'estimateur plug-in pour ces points exclusivement. Aux

points limites, cet estimateur vaut

$$\hat{p}_{x_i} = \frac{k \pm 1}{2k} \ ,$$

et l'estimation de la variance est

$$\frac{(k+1)(k-1)}{4k^2} \ .$$

Considérons les valeurs de la variance pour les premières valeurs de k impaires :

$$\begin{aligned} k = 1 &\longrightarrow \hat{p}(1-\hat{p}) = 0 \\ k = 3 &\longrightarrow \hat{p}(1-\hat{p}) = 0.22 \\ k = 5 &\longrightarrow \hat{p}(1-\hat{p}) = 0.24 \\ &\vdots \end{aligned}$$

On constate que dès $k = 3$, la variance estimée α est très proche de la borne supérieure de la variance 0.25 : les variations de l'estimation de α sont plus petites que 0.01 lorsque k est supérieure à 5. Seul le cas où $k = 1$ donne un estimateur très biaisé du bruit de classification aux points limites, mais nous avons vu précédemment cette valeur de k ne peut pas être choisie avec le critère (S) classique. Nous interdisons donc ici cette valeur, i.e. nous ne considérons que les valeurs de $k \geq 3$. On considère ainsi une version du critère Swapping où l'estimateur plug-in n'est plus régularisé, et où $k \geq 3$. Nous appelons cette nouvelle version du Swapping S_0kNN, où S_0 désigne la pénalité Swapping avec n_0=0.

Le tableau 3.4 présente les résultats de S_0kNN sur les jeux de données étudiés dans la partie précédente. Tout d'abord, on remarque que bien que l'estimateur plug-in n'est pas régularisé, on ne choisit pas systématiquement la valeur $k = 3$. Le nombre de voisins considérés peut même monter jusqu'à k=47 pour le jeu de données Pima. Ensuite, on remarque que les deux méthodes S_0kNN et (CV) restent comparables du points de vue des performances. Le seul changement notable de ce point de vue est le renversement de situation pour le jeu de données Colon, où précédemment (S) réussissait mieux que (CV) pour les nombres de gènes faibles alors que S_0kNN vainc (CV) pour les nombres de gènes élevés. Nous avons vérifié que cette inversion n'était pas due à la procédure d'échantillonnage en réalisant 5000 échantillonnages. Les résultats restent les mêmes. L'amélioration vient de l'estimation du taux d'erreur : non seulement S_0kNN fournit de meilleures estimations que (CV), mais on peut constater que l'écart entre l'estimation et le taux d'erreur réel ne dépasse jamais les 3% pour S_0kNN. Ainsi cette version alternative, plus simple que la première, donne des résultats satisfaisants sur les jeux de données étudiés.

3.4 Discussion

Comme le montre le théorème 3.1.1 obtenue en partie 3.1, le critère Swapping peut être considéré comme un critère pénalisé par covariance, dont la théorie fut très récemment développée, en particulier par Efron (2004). La formule présentée au théorème 3.1.1 fut aussi dérivée par Efron (1986), bien que dans un cadre différent et avec une autre démonstration. L'estimateur Swapping fut proposé indépendamment par Efron (2004) et Daudin et Mary-Huard (2005), l'application aux kNN étant proposée dans cette dernière référence. D'autres méthodes d'estimation de la covariance

$$\frac{2}{n} \sum_{i=1}^{n} cov\left(I_{\{Y_i=1|X_i=x_i\}}, \Phi_n^*(x_i)\right)$$

	Swapping			CV		
Dataset	k	$\widehat{TER}$	TER	k	$\widehat{TER}$	TER
Cleveland	38.7	15.2	**16.7**	40.2	14.9	16.9
Pima	46.9	24	**25.2**	46.6	23.7	25.3
Wisconsin	19	2.9	**3.1**	19.4	2.8	**3.1**
Colon100	4.5	14.7	17.1	18.2	9.1	**15.9**
Colon500	3.7	16.1	18.9	15	10.5	**18.1**
Colon1000	4	19	**21.1**	11.7	12.9	21.4
Colon2000	4.5	25.2	**27.3**	7.9	22.5	28.2

TAB. 3.4 – *Comparaison S_0kNN et (CV) sur le jeu de données Colon, pour différents nombre de gènes sélectionnés. Pour chaque méthode (S) et (CV), le nombre moyen de voisins k et les taux d'erreur estimé $\widehat{TER}$ et réel TER sont reportés. Le meilleur taux d'erreur obtenu est signalé en gras.*

ont été proposées. Par exemple Tibshirani et Knight (1999) utilisent un estimateur basé sur la permutation des labels pour estimer ce terme. Pour chaque point de l'échantillon d'entraînement x, un label Y est attribué en tirant parmi les labels observés y_1,...,y_n sans remise. Cette méthode diffère de la notre puisqu'elle suppose implicitement que la probabilité pour une observation d'avoir le label 1 est reliée à la fréquence de ce label dans l'échantillon, ce qui n'est pas le cas de la méthode Swapping. Par ailleurs, les auteurs précisent que leur méthode n'est pas adaptée à l'algorithme des kNN, alors que le Swapping donne de bons résultats pour cet algorithme.

L'avantage de la méthode Swapping est son universalité : contrairement aux méthodes de pénalisation du risque empirique de type Vapnik, elle peut être appliquée à toutes les règles de classification. La seule quantité nécessaire au calcul de la pénalité est la probabilité *a posteriori* $P(Y = 1|X = x)$ en chacun des points limites, estimation qui peut ou non provenir de la règle de classification étudiée. Un futur travail est donc l'application de ce critère à d'autres méthodes de classification populaires, en particulier à l'algorithme CART, où le Swapping serait employé pour réaliser l'élagage de l'arbre. De manière plus générale, le Swapping peut être un concurrent de la validation croisée pour toutes les méthodes de classification nécessitant le réglage préalable de paramètres.

L'application du Swapping à l'algorithme des kNN donne des résultats intéressants et prometteurs. A notre connaissance, le Swapping est le premier critère pénalisé adaptatif développé pour cet algorithme. Nous avons vu que le principal apport du Swapping par rapport à la validation croisée est d'estimer plus finement le taux d'erreur réel, pour un coût algorithmique et des performances comparables. Nous avons aussi montré que la pénalité Swapping n'est en réalité basée que sur un petit nombre de points limites. Il est tentant d'établir une correspondance entre ces points limites obtenus pour les kNN et les points supports qui apparaissent dans la définition de l'hyperplan séparateur des SVM. Dans les deux cas, les algorithmes isolent des points difficiles à classer. Il est toutefois difficile de mener l'analogie plus loin. Considérons par exemple le cas de la figure 3.6. Supposons que le point noir apparaissant à l'extrême droite ait pour label "rouge". Ce point serait un vecteur support pour les SVM, mais ne serait pas un point limite pour le Swapping : il serait simplement mal classé. Les deux notions ne sont donc pas identiques. Pouvoir préciser la nature de ces deux notions et parvenir à les relier, tant du point de vue algorithmique que théorique, est donc une autre piste de recherche à explorer.

La diffusion des méthodes statistiques auprès des utilisateurs appliqués est une tâche importante, qui nécessite de proposer des algorithmes simples d'utilisation et propres à être employés sans être un spécialiste du domaine de l'apprentissage statistique. Nous avons proposé deux versions de la méthode Swapping. La première nécessite la calibration d'un paramètre de régularisation n_0. La deuxième version est plus "automatique", puisqu'elle peut être utilisée directement sans réglage de paramètre. En ce sens, cette dernière version promet d'être un outil adapté aux besoins des utilisateurs non statisticiens.

Chapitre 4

Critère pénalisé pour la sélection de variables

Nous avons présenté au chapitre 2 les différents arguments motivant la sélection de variables. Nous avons aussi introduit les deux types de méthodes de sélection, filter et wrapper, et étudié les méthodes filter basées sur les couvertures de Markov. Nous nous intéressons maintenant aux méthodes de type wrapper, et aux méthodes intégrées ("embedded methods"). Nous présentons un exemple de chacune de ces catégories.

Ainsi que nous les avons introduites dans le chapitre précédent, les méthodes wrapper quantifient la qualité d'un sous-ensemble de variables par le taux d'erreur réel obtenu en combinant ce sous-ensemble avec un algorithme de classification préalablement choisi. Plusieurs alternatives peuvent être envisagées pour estimer les taux d'erreur des différents classificateurs construits. Lorsque le nombre d'observations dont on dispose est grand, l'échantillon initial peut être divisé en un échantillon d'entraînement et un échantillon test, le premier servant à construire le classificateur et le deuxième à évaluer ses performances. Lorsque le nombre d'observations est trop faible, le taux d'erreur réel peut être estimé par validation croisée. Enfin, dans certains cas le taux d'erreur peut être simplement majoré par une borne supérieure calculable à partir des données (Guyon *et al.* (2002), Rakotomamonjy (2003), Zhu et Hastie (2003)). Cette dernière possibilité est illustrée par l'algorithme RFE (pour Recursive Feature Elimination), proposé par Guyon *et al.* (2002). La RFE est une méthode wrapper séquentielle dédiée aux SVM. Elle consiste à retirer à chaque étape la variable dont le poids dans la construction de l'hyperplan séparateur SVM est le plus petit. A chaque fois qu'une variable est retirée, un nouvel hyperplan est calculé, les variables restantes sont reclassées suivant leur poids dans le nouvel hyperplan, et le processus d'élimination continue. La justification heuristique d'une telle méthode repose sur la borne supérieure suivante de l'erreur par validation croisée L_{lo} :

$$L_{lo} \leq 4R^2||\mathbf{w}||_2^2 \ ,$$

où R est le rayon de la plus petite boule contenant les données dans l'espace de représentation, et $||\mathbf{w}||_2$ est la norme du vecteur orthonormal à l'hyperplan séparateur. La procédure RFE a donc pour objectif de diminuer $||\mathbf{w}||_2$ pour minimiser la borne supérieure sur L_{lo}.

Les méthodes intégrées sont appelées ainsi car elles fournissent spontanément une règle de classification qui ne dépend que d'une partie des variables disponibles dans l'échantillon d'entraînement. Comme les méthodes wrapper, elles sont par définition dédiées à un algorithme de classification donné, mais les temps de calcul sont considérablement plus courts, et peuvent être comparables aux temps de calcul des méthodes Filter (Guyon et Elisseeff (2003)).

C'est pourquoi beaucoup de chercheurs se sont récemment intéressés aux méthodes intégrées (Krishnapuram *et al.* (2004a), Weston *et al.* (2000), Lal *et al.*). La méthode intégrée la plus célèbre est certainement la méthode CART, qui sera étudiée en partie 4.4. Nous présentons ici la méthode Joint Classifier and Feature Optimization (JCFO, Krishnapuram *et al.* (2004b)) car elle illustre cette nouvelle famille des méthodes intégrées. Dans cette méthode, la construction du classificateur est basée sur un programme de minimisation d'un risque pénalisé, où la pénalité dépend à la fois de la complexité du classificateur et du nombre de variables sélectionnées. Du point de vue algorithmique, le programme de minimisation peut être reformulé comme un programme bayésien de maximisation *a posteriori* de la vraisemblance (MAP). Les termes de régularisation correspondent alors aux lois *a priori* posées sur les différents paramètres. Le passage par le formalisme bayésien permet l'application des méthodes de calcul usuellement employées dans ce contexte, comme l'algorithme EM par exemple.

Bien que la plupart des méthodes wrapper et des méthodes intégrées soient basées sur des heuristiques raisonnables, il n'existe que peu de résultats théoriques garantissant les performances de ces méthodes dans la littérature. Pour les méthodes de type JCFO par exemple, l'expression du critère pénalisé à optimiser est directement imposée par le choix des lois *a priori* posées sur les paramètres. Or ces lois sont choisies de manière pragmatique : le programme de maximisation est réalisable lorsque l'on prend les lois *a priori* proposées. Ainsi, dans Krishnapuram *et al.* (2004a) les auteurs concluent :

Can we obtain tight theoretical upper bounds on the error rate that can be used to motivate further improvements in algorithm? We are currently investigating different ways to address these issues.

Les exemples de la RFE et de la JCFO illustrent le manque de résultats théoriques en sélection de variables pour la classification. En l'absence de tels résultats, seules les procédures d'estimation par validation-croisée ou par échantillon test permettent de garantir localement les performances.

Nous présentons ici un critère de sélection de variables basé sur la minimisation du risque empirique pénalisé, et pour lequel les performances sont garanties par une inégalité oracle. Ce résultat est obtenu par l'adaptation de résultats de sélection de modèles développés par différents auteurs (Bartlett *et al.* (2000), Birgé et Massart (2001a), Devroye *et al.* (1996), Lugosi et Zeger (1995), Massart (2000)) au contexte de la sélection de variables. L'étude de ce critère va permettre une compréhension qualitative du rôle des différents paramètres intervenant dans la pénalisation, et donne les éléments théoriques nécessaires pour la construction de nouvelles méthodes wrapper ou intégrées performantes.

Ce chapitre est organisé de la manière suivante : nous introduisons en partie 4.1 le problème de sélection de variables comme un problème de sélection de modèles. La partie 4.2 est consacrée à la présentation du résultat principal et à sa démonstration. Une application théorique aux méthodes backward et forward est présentée en partie 4.3. La partie 4.4 propose une analyse de l'algorithme CART en tant que méthode de sélection de variables intégrée basée sur le résultat de la partie 4.2. Enfin, la partie 4.5 donne quelques éléments de discussion sur les résultats présentés.

Le travail présenté ici fera l'objet d'une publication en 2006, sous le titre "A penalized criterion for variable selection in classification" (T. Mary-Huard, S. Robin and J.J. Daudin) dans Journal of Multivariate Analysis.

4.1 Contexte

4.1.1 Classification

Certaines de nos notations changent par rapport aux parties précédentes pour des raisons techniques qui sont expliquées en partie 4.2. Afin de ne pas perdre le lecteur, nous reprenons brièvement le problème de classification détaillé au chapitre 1 en introduisant au fur et à mesure les nouvelles notations.

On cherche à construire un classificateur Φ sur la base de données d'entraînement (X_i,Y_i), $i = 1,...,n$, où Y_i est un label binaire (0 ou 1) et X_i est un vecteur à p variables. Le classificateur de Bayes, introduit dans la première partie de ce manuscrit, est ici désigné par Φ^B et défini par :

$$\Phi^B(x) = \begin{cases} 1 & \text{si } E\left(Y = 1|X = x\right) > 1/2 \\ 0 & \text{sinon.} \end{cases}$$

Ce classificateur dépend de la fonction de régression $\Phi^*(x) = E\left(Y = 1|X = x\right)$. Cette fonction de régression est en pratique inconnue, puisque l'on ne connaît pas P, la loi jointe du couple (X,Y).

L'estimation de Φ^B peut être réalisée par minimisation du risque empirique. On considère une classe $\mathcal{C}$ de classificateurs, parmi lesquels nous devons choisir un candidat pour estimer Φ^B. Par définition, on a

$$\Phi^* = \underset{\Phi\in\mathcal{L}^2}{\text{Argmin}}\, E\left(Y - \Phi(X)\right)^2 \quad ,$$

on peut donc estimer Φ^B dans la classe $\mathcal{C}$ par

$$\begin{aligned} \hat{\Phi} &= \underset{\Phi\in\mathcal{C}}{\text{Argmin}}\, \frac{1}{n}\sum_{i=1}^{n}(Y_i - \Phi(X_i))^2 \\ &= \underset{\Phi\in\mathcal{C}}{\text{Argmin}}\, L_n(\Phi) \quad . \end{aligned} \tag{4.1}$$

On remarque que l'équation (4.1) ne correspond pas à la présentation usuelle de l'ERM : la perte $hard - loss$ usuellement employée est ici remplacée par la perte quadratique. Malgré ce changement, le minimiseur $\hat{\Phi}$ obtenu est rigoureusement identique au minimiseur classique, puisque la minimisation de la perte quadratique est réalisée sur un ensemble de fonctions qui ne prennent pour valeurs que 0 ou 1. $\hat{\Phi}$ est donc le minimiseur du risque empirique.

Nous avons vu au chapitre 1 que la stratégie ERM est une stratégie pertinente, à condition que la classe $\mathcal{C}$ soit bien choisie, i.e. établisse un bon compromis entre biais et variance. Le choix de cette classe est donc crucial, et comme nous ne disposons pas d'information *a priori* pour le choix de cette classe, nous allons considérer plusieurs classes $\mathcal{C}_m$, $m = 1,...,M$ plutôt qu'une seule. L'ERM est appliquée à chaque classe pour obtenir un candidat $\hat{\Phi}_m$, puis le choix entre les différents candidats se fait par minimisation d'un critère pénalisé de la forme

$$L_n(\hat{\Phi}_m) + pen(\mathcal{C}_m,n) \quad , \tag{4.2}$$

où $pen(\mathcal{C}_m,n)$ dépend de la complexité de la classe $\mathcal{C}_m$ et du nombre d'observations. L'idée est ici d'adapter cette stratégie de sélection de modèles au problème de sélection de variables. Pour cela, nous devons spécifier les classes $\mathcal{C}_m$ à considérer.

4.1.2 Sélection de variables

Nous n'avons pour l'instant imposé aucune forme spécifique aux classes $\mathcal{C}_m$, ni aucune contrainte sur leur complexité. Dans le contexte de la sélection de variables, on définit un classificateur Φ comme étant une fonction de complexité k (par exemple Φ est un polynôme de degré k) appliquée à un sous-ensemble de j variables choisies parmi les p variables constituant X. Il existe C_p^j sous-ensembles de j variables parmi p, que l'on indexe par ℓ. On définit ainsi $\mathcal{C}_{k,j,\ell}$ comme étant la classe composée de l'ensemble des classificateurs de complexité k, qui ne prennent en compte pour la classification que les j variables de X figurant dans le sous-ensemble ℓ. En résumé :

- $j = 1,...,p$ désigne le nombre de variables,
- $\ell = 1,...,\mathrm{C}_p^j$ désigne un sous-ensemble particulier à j variables,
- $k = 1,...,\infty$ désigne la complexité de Φ.

Nous devons aussi définir une mesure de la complexité d'une classe. Dans la partie 1.2, cette mesure était donnée par la dimension de Vapnik de la classe. Nous définissons ici l'entropie de Vapnik-Chervonenkis de la classe $\mathcal{C}_{k,j,\ell}$:

$$H_{k,j,\ell} = \log |\{A(\Phi) \cap \{X_1^\ell, \ldots, X_n^\ell\},\ \Phi \in \mathcal{C}_{k,j,\ell}\}|,$$

où X_i^ℓ est le vecteur aléatoire contenant les j coordonnées $X_i^{(1)},...X_i^{(j)}$ du sous-ensemble ℓ, et $A(\Phi) = \{X^\ell : \Phi(X^\ell) = 1\}$. Pour finir, nous définissons la fonction de perte

$$l(\Phi^*,\Phi) = \mathbb{E}[\Phi^*(X) - \Phi(X)]^2 \ , \tag{4.3}$$

qui mesure la performance relative du classificateur Φ comparée à la fonction de régression.

4.2 Résultat principal

4.2.1 Présentation du résultat principal et commentaires

Pour chaque classe $\mathcal{C}_{k,j,\ell}$ définie dans la partie précédente, on détermine le minimiseur du risque empirique $\hat{\Phi}_{k,j,\ell}$. Le choix entre les différents candidats est ensuite réalisé en minimisant un critère pénalisé de la forme

$$L_n(\hat{\Phi}_{k,j,\ell}) + pen(\mathcal{C}_{k,j,\ell},n)$$

Le théorème suivant donne la forme explicite de la pénalité, ainsi que le risque associé à la stratégie proposée.

Théorème 4.2.1. *On définit $\hat{\Phi}_{opt}$ comme étant le minimiseur du risque empirique pénalisé*

$$\hat{\Phi}_{opt} = \operatorname*{Argmin}_{k,j,\ell} L_n(\hat{\Phi}_{k,j,\ell}) + pen(\mathcal{C}_{k,j,\ell},n),$$

où

$$pen(\mathcal{C}_{k,j,\ell},n) = K_1' \frac{H_{k,j,\ell}}{n} + K_2' \left[\frac{\log \mathrm{C}_p^j}{n} + \frac{2\log(j+k)}{n} \right] \tag{4.4}$$

pour des constantes K_1' et K_2' données. Alors, l'inégalité oracle suivante garantit la performance du classificateur sélectionné $\hat{\Phi}_{opt}$:

$$\mathbb{E}[l(\Phi^*,\hat{\Phi}_{opt})] \leq C(1+\log(p)) \left\{ \inf_{k,j,\ell} \left[\inf_{\Phi \in \mathcal{C}_{k,j,\ell}} l(\Phi^*,\Phi) + K_1' \frac{\mathbb{E}\left(H_{j,k,\ell}\right)}{n} + C''' \frac{\log(k+j)+j}{n} \right] \right\} \tag{4.5}$$

où C et C''' sont des constantes qui dépendent de K_1' et K_2'. La démonstration de ce résultat est donnée dans la partie suivante.

Remarque 1 Si l'on considère la définition (4.4), on constate que le nombre de classes à j variables apparaît explicitement dans le terme de pénalité. Ce résultat est en réalité tout à fait intuitif, puisqu'il illustre le fait que trouver un sous-ensemble optimal de 10 variables parmi 100 est plus facile que de trouver un sous-ensemble optimal de 10 variables parmi 10.000. Ainsi, dans les problèmes de grande dimension qui nous intéressent, ce terme est prépondérant dans la pénalisation.

Remarque 2 La plupart des méthodes de sélection de variables proposées supposent que la complexité k des différentes classes de fonctions est identique. Par exemple, lorsque l'on réalise la RFE, le noyau utilisé pour les SVM doit être spécifié avant de réaliser la sélection de variables. A l'inverse, pour notre résultat les complexités de chaque classes peuvent être différentes et peuvent prendre autant de valeurs que nécessaire (k peut varier de 1 à $+\infty$). De ce fait, il est possible de faire varier conjointement le nombre de variables et la complexité : on pourra ainsi comparer et préférer un classificateur basé sur un petit sous-ensemble de variables avec de fortes interactions à un sous-ensemble plus large mais avec des interactions plus faibles.

Remarque 3 La forme de l'inégalité oracle obtenue précédemment est semblable à la forme obtenue dans d'autres contextes où le nombre de modèles par dimension est crucial, comme la détection de ruptures Lebarbier (2005) ou l'estimation de densité Castellan (2000). De plus, le facteur $(1 + \log(p))$ qui apparaît dans l'inégalité (4.5) est aussi présent dans le résultat de sélection de variables dans le contexte de la régression présenté dans Birgé et Massart (2001a). Dans cet article, il est montré que la forme de l'inégalité est optimale du point de vue minimax, suggérant que le terme $(1 + \log(p))$ qui apparaît ici est aussi optimal. Du point de vue minimax la majoration obtenue est fine, et seules les constantes K_1', K_2', C and C''' qui apparaissent dans la pénalité et l'inégalité oracle peuvent être améliorées. En effet, ces constantes résultent de l'inégalité de concentration plus ou moins fine choisie pour contrôler le risque sur un modèle donné (voir la démonstration) et des majorations effectuées tout au long de la démonstration.

4.2.2 Démonstration

Le résultat présenté ici est une adaptation d'un résultat de sélection de modèles démontré dans Massart (2000). Les notations adoptées ici diffèrent de celles du papier original, mais pour faciliter la lecture de la démonstration et la confrontation avec le résultat initial, la désignation des constantes est identique à celle utilisée dans l'article original. Bien qu'il ne soit pas ici possible de reprendre l'intégralité de la démonstration originale, nous en présentons les principales étapes que nous commentons brièvement.

Le résultat initial est le suivant. On définit $\mathcal{C}_k$ comme une classe de fonctions $\Phi : \mathbb{R}^p \to \{0,1\}$, on désigne par H_k l'entropie de Vapnik-Chervonenkis associée à cette classe et on définit

$$\hat{\Phi}_{opt} = \underset{k}{\text{Argmin}} \left[L_n(\hat{\Phi}_k) + K_1' \frac{H_k}{n} + K_2' \frac{x_k}{n} \right] \tag{4.6}$$

où K_1' et K_2' sont des constantes données, et où les x_k sont des poids associés à chaque classe. Si le nombre de classe est au plus de n, on peut choisir des poids x_k égaux entre eux et valant

$\log(n)$. On obtient alors l'inégalité oracle suivante :

$$\mathbb{E}[l(\Phi^*,\hat{\Phi}_{opt})] \leq C \left\{ \inf_k \left[\inf_{\Phi \in \mathcal{C}_k} l(\Phi^*,\Phi) + K_1' \frac{\mathbb{E}\left(H_k\right)}{n} \right] + C' \frac{1+\log(n)}{n} \right\} \tag{4.7}$$

où les constantes C et C' sont connues et dépendent de K_1' et K_2'.

La démonstration d'un tel résultat de sélection de modèles peut se décomposer en deux grandes étapes (on pourra se référer au chapitre 1 ou à Birgé et Massart (2001a) pour plus de détails). La première consiste à contrôler le risque sur un modèle (*capacity control*) et la seconde à établir la borne sur l'ensemble des modèles simultanément (*union bound*). On introduit les notation suivantes :

$$\begin{aligned} \bar{\gamma}_n(\Phi) &= \frac{1}{n}\sum_{i=1}^{n} \gamma(\Phi,X_i,Y_i) - E[\gamma(\Phi,X,Y)] \\ \Phi_m &= \underset{\Phi \in \mathcal{C}_m}{\text{Argmin}}\ E[\gamma(\Phi,X,Y) - \gamma(\Phi^*,X,Y)] \quad . \end{aligned}$$

La première étape consiste à contrôler le risque sur un modèle en proposant une borne probabiliste pour la quantité

$$\sup_{\Phi \in \mathcal{C}_{m'}} |\bar{\gamma}_n(\Phi) - \bar{\gamma}_n(\Phi_m)| \tag{4.8}$$

pour une classe $\mathcal{C}_m$ donnée. On utilise pour cela les inégalités de concentration. Dans la théorie de Vapnik présentée au chapitre 1, γ représente la fonction de perte $hard-loss$, et le contrôle de (4.8) peut être ramené au problème du contrôle de $\sup_{\Phi \in \mathcal{C}_{m'}} \bar{\gamma}_n(\Phi)$. Nous avons vu qu'un tel contrôle universel sur la classe $\mathcal{C}_m$ résulte en une inégalité oracle dont la vitesse de convergence est de l'ordre de $1/\sqrt{n}$. Dans Massart (2000), l'auteur montre qu'il est possible de contrôler plus finement le terme (4.8) en remplaçant la fonction de perte $hard - loss$ par la fonction de perte quadratique et en utilisant l'inégalité de Talagrand. Ce contrôle plus fin permet d'obtenir des décroissances en $1/n$ et non plus en $1/\sqrt{n}$.

Dans la deuxième étape, le contrôle précédent est réalisé sur tous les modèles simultanément, et un poids x_m est ajouté à chaque borne supérieure afin de garantir toutes les majorations avec une probabilité globale de $1 - e^{-\xi}$. Dans le résultat initial, on fait l'hypothèse que le nombre de classes considérées n'est pas trop grand : ce nombre est supposé inférieur à n. Ainsi, il suffit de choisir des poids $x_m = \log(n)$ pour garantir la condition nécessaire suivante :

$$\sum_m e^{-x_m} \leq \Sigma < \infty \quad .$$

Dans le cadre de la sélection de variables, le nombre de classes considérées est élevé, et le nombre de classes de même complexité peut être élevé aussi. Plusieurs auteurs ont déjà souligné le fait que dans une telle situation il est préférable de ne pas prendre des poids identiques pour toutes les classes (Lebarbier (2005), Castellan (2000)). Nous modifions donc le résultat initial à cette étape.

Avec les notations introduites en partie 4.1, on désigne par $\mathcal{C}_{k,j,\ell}$ la classe des fonctions de complexité k, où seules j variables parmi les p initiales sont utilisées. Dans le contexte de la sélection de variables, les poids doivent être modifiés de manière à prendre en compte le nombre important de classes considérées. Par ailleurs, on souhaite que des classes de même

complexité, c'est-à-dire ayant des valeurs de k et j identiques, aient des poids identiques. La condition nécessaire précédente devient alors :

$$\sum_{k=1}^{\infty}\sum_{j=1}^{p} \mathrm{C}_p^j \exp\left(-x_{j,k}\right) \leq \Sigma.$$

On choisit alors les poids suivants :

$$x_{j,k} = \frac{\log \mathrm{C}_p^j}{n} + \frac{2\log(j+k)}{n}$$

Avec ces nouveaux poids, l'inégalité (4.7) peut être reformulée de la manière suivante :

$$\mathbb{E}[l(\Phi^*,\hat{\Phi}_{opt})] \leq C\left[\inf_{k,j,\ell}\left(\inf_{\Phi\in\mathcal{C}_{k,j,\ell}} l(\Phi^*,\Phi) + K_1'\frac{\mathbb{E}\left(H_{k,j,\ell}\right)}{n}\right) + C''\frac{1+2\log(k+j)+\log \mathrm{C}_p^j}{n}\right]$$

On utilise l'inégalité $\log \mathrm{C}_p^j \leq j(1+\log(\frac{p}{j}))$ pour obtenir :

$$\begin{aligned}
\mathbb{E}[l(\Phi^*,\hat{\Phi}_{opt})] &\leq C\left[\inf_{k,j,\ell}\left(\inf_{\Phi\in\mathcal{C}_{k,j,\ell}} l(\Phi^*,\Phi) + K_1'\frac{\mathbb{E}\left(H_{k,j,\ell}\right)}{n} + C''\frac{1+2\log(k+j)+j(1+\log(\frac{p}{j}))}{n}\right)\right] \\
&\leq C\left[\inf_{k,j,\ell}\left(\inf_{\Phi\in\mathcal{C}_{k,j,\ell}} l(\Phi^*,\Phi) + K_1'\frac{\mathbb{E}\left(H_{k,j,\ell}\right)}{n} + C'''(1+\log(p))\frac{\log(k+j)+j}{n}\right)\right] \\
&\leq C(1+\log(p))\left[\inf_{k,j,\ell}\left(\inf_{\Phi\in\mathcal{C}_{k,j,\ell}} l(\Phi^*,\Phi) + K_1'\frac{\mathbb{E}\left(H_{k,j,\ell}\right)}{n} + C'''\frac{\log(k+j)+j}{n}\right)\right]
\end{aligned}$$

pour des constantes C et C''' appropriées, ce qui achève la démonstration.

4.3 Application à la sélection séquentielle backward

En théorie, la sélection de variables exhaustive est optimale, puisque tous les sous-ensembles possibles de variables sont considérés. En pratique, la sélection exhaustive est généralement impossible à réaliser car le nombre de modèles à visiter est de 2^p : le coût algorithmique d'une procédure exhaustive serait trop important. On emploie des algorithmes parcimonieux, qui ne visitent qu'un nombre restreint de sous-ensembles. Les algorithmes parcimonieux les plus couramment utilisés sont les algorithmes de types backward, forward ou Stepwise (voir Kohavi et John (1997) et le chapitre 2 de ce document), proposés par différents auteurs. Ces algorithmes ont été largement adoptés car leurs performances sont empiriquement établies sur un grand nombre de données d'application (Guyon *et al.* (2002), Zhu et Hastie (2003)). Nous reprenons rapidement ici le principe de l'algorithme backward. Dans cette partie, nous supposerons le paramètre de complexité k fixé.

L'algorithme backward prend pour jeu de données initial l'ensemble des variables et procède de manière récursive. A chaque étape, la variable la moins pertinente au sens d'un critère de qualité préalablement défini est retirée. Dans l'exemple de la RFE présentée en introduction, le critère de qualité est le poids associé à chaque variable dans l'hyperplan. Lorsque toutes les variables ont été retirées, on obtient un classement des variables de la plus pertinente (dernière retirée) à la moins pertinente (première retirée). Le lecteur intéressé consultera Rakotomamonjy (2003) pour une étude détaillée sur les procédures backward appliquées aux SVM, illustrée par de nombreux exemples de critères de qualité. La difficulté est

ensuite de fixer le nombre r de variables à retenir pour construire le classificateur final. Une méthode possible pour choisir ce nombre de variables est d'estimer le taux d'erreur de chaque classificateur construit avec les j meilleures variables de la liste, soit par validation-croisée, soit par échantillon test. Toutefois, ces deux alternatives nécessitent pour la première beaucoup de calculs supplémentaires et pour la deuxième des données supplémentaires. Bien souvent, le nombre de variables retenues est donc fixé à l'avance, et non choisi en fonction des données. Notre objectif est ici de proposer quelques pistes pour l'élaboration d'un critère d'arrêt pour les procédures de sélection backward basées sur la minimisation du risque empirique.

On considère la procédure backward suivante : on commence avec toutes les variables, et à chaque étape on retire la variable dont l'absence provoque la plus petite dégradation du taux d'erreur empirique. Une fois les variables classées, on note $\hat{\Phi}_j$ le classificateur contenant les j premières variables du classement. On aimerait déterminer un critère d'arrêt pénalisé de la forme

$$Crit(j) = L_n(\hat{\Phi}_j) + \text{pen}(Z,j), \;\; Z = \{(X_1,Y_1),\ldots,(X_n,Y_n)\}, \;\; j = 1,\ldots,p.$$

En application directe du théorème 4.2.1, on peut formuler la proposition suivante :

Propriété 4.3.1. *Le critère d'arrêt optimal pour la sélection backward doit satisfaire :*

$$Crit(j) < L_n(\hat{\Phi}_j) + K_1\frac{H_j}{n} + K_2\frac{\log \mathrm{C}_p^j}{n} \quad . \tag{4.9}$$

On obtient ainsi une borne supérieure pour le critère d'arrêt. La proposition suivante donne une borne inférieure :

Propriété 4.3.2. *Le critère d'arrêt optimal pour la sélection backward doit satisfaire :*

$$Crit(j) > L_n(\hat{\Phi}_j) + K_1\frac{H_j}{n} + K_2\frac{\log(j+1)}{n} \quad . \tag{4.10}$$

Démonstration. La démonstration des deux propositions tient au décompte des modèles visités.
A chaque étape t, le comportement de la sélection backward est le suivant : on considère le modèle à $t+1$ variables obtenu à l'étape précédente, et on détermine le meilleur modèle à t variables choisies parmi les $t+1$ précédentes. On remarque que :

- à chaque étape, c'est-à-dire pour un nombre fixé t de variables, le nombre de modèles considérés est plus petit que le nombre de modèles visités en sélection exhaustive,
- tous les modèles visités en sélection backward sont aussi visités en sélection exhaustive.

De ces deux constatations on conclut que la pénalité (4.4) proposée pour la sélection exhaustive est trop forte pour la sélection backward, ce qui justifie la proposition 4.3.1. Remarquons qu'entre le théorème 4.2.1 et la proposition 4.3.1 les indices k et ℓ ont disparu : le premier est fixé, et le deuxième disparaît puisqu'en sélection backward on ne considère qu'un seul sous-ensemble à j variables. On peut choisir comme poids $x_m = \log \mathrm{C}_p^j + \log(p)$ et omettre la dernière partie des poids pour la comparaison.

La démonstration de la proposition 4.3.2 est du même ordre que la précédente. A l'étape t, le nombre de classes visitées depuis le début est $t+1$. Si la séquence des classes $\mathcal{C}_t$, où l'on considère les t premières variables, était fixée et non pas déduite des données, alors nous

n'aurions considéré qu'un sous-modèle à t variables pour chaque t, et en reprenant la preuve du théorème 4.2.1 nous aurions obtenu le critère pénalisé suivant :

$$L_n(\hat{\Phi}_t^*) + K_1' \frac{H_t}{n} + K_2' \frac{\log(t+1)}{n} \quad .$$

Appliquer ce critère dédié à une séquence déterministe à la sélection backward est clairement optimiste puisque nous ne prenons pas en compte le fait qu'à chaque étape t, t variables sont choisies de manière optimale parmi les $t+1$ de l'étape précédente. □

Il est clair que l'on peut traiter de manière identique la sélection forward à partir des mêmes considérations, et obtenir là aussi des bornes supérieure et inférieure pour le critère d'arrêt. Dans ce cas, la borne de la proposition 4.3.1 serait la même pour les deux critères d'arrêt, et la borne inférieure serait modifiée de la manière suivante :

$$L_n(\hat{\Phi}_j) + K_1 \frac{H_j}{n} + K_2 \frac{\log(p-j+1)}{n}.$$

4.4 Une illustration : l'algorithme CART

4.4.1 La méthode CART

Parallèlement à l'obtention de critères d'arrêt pour les méthodes de sélection de variables de type backward ou forward, le théorème 4.2.1 fournit aussi un cadre rigoureux pour la compréhension d'algorithmes déjà existants. Nous illustrons ce propos en analysant ici le fonctionnement de l'algorithme CART de Breiman *et al.* (1984), à la lumière du théorème 4.2.1.

La facilité d'interprétation de la règle de décision fournie par CART a rendu cet algorithme très populaire, et plusieurs auteurs l'ont appliqué avec succès à des problèmes de nature très différentes, comme la détection de spam (Hastie *et al.* (2001)) ou le diagnostic médical (Ripley et Hjort (1995)). Nous présentons ici le principe de l'algorithme.

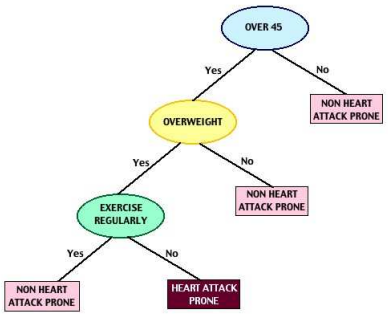


FIG. 4.1 – *Un exemple d'arbre de classification. Si le patient répond oui à la première question puis non à la deuxième, il sera classé comme peu enclin aux attaques cardiaques.*

Considérons l'exemple du diagnostic d'attaque cardiaque. Lorsqu'un patient est admis au centre médical, le diagnostic pour ce patient doit être établi sur la base d'une série de questions successives auxquelles le patient doit répondre par oui ou par non. Ce type de règle de classification peut être représenté graphiquement par un arbre binaire (Figure 4.1). Chaque noeud non terminal de l'arbre correspond à une question. Chaque noeud terminal,

appelé feuille, correspond à un diagnostic, "sujet enclin à faire une attaque" ou "sujet non enclin". Tout nouveau patient est classé dans l'une des feuilles de l'arbre en fonction de ses réponses aux questions.

L'étape d'apprentissage a pour objectif la construction de l'arbre, ce qui nécessite :

- de définir à chaque noeud une question, c'est-à-dire de choisir une variable et un seuil. Dans la Figure 4.1, la première question "Age > 45" est établie en choisissant la variable "Age" et le seuil "45" ;
- de choisir une taille d'arbre, cette taille étant mesurée par le nombre de feuilles. Généralement, on construit dans un premier temps un arbre complet, c'est-à-dire un arbre suffisamment grand pour classer sans erreur toutes les données de l'échantillon d'entraînement, puis on considère plusieurs sous-arbres issus de l'arbre complet. Dans l'exemple de la figure 4.1, l'arbre retenu a une taille de 4. Le choix d'un sous-arbre est un problème de sélection de modèles.

En pratique, comment construit-on un tel arbre de classification ? On rappelle que l'on dispose d'un échantillon E d'entraînement de n observations (X_i,Y_i). On commence par déterminer la première question à poser. Pour cela, on essaie toutes les combinaisons possibles d'une variable X^j et d'un seuil t_j pour former la question "$X^j > t_j$". Chaque question possible sépare l'échantillon d'entraînement en deux sous-échantillons, E_{oui} et E_{non}, suivant les réponses obtenues à la question. On cherche la combinaison variable-seuil telle que les deux sous-échantillons E_{oui} et E_{non} obtenus (ceux qui répondent "oui" à la question et les autres) soient le plus purs possible au sens du label à prédire. La pureté d'une population E peut être quantifiée par différents indicateurs :

$$\begin{array}{lll} I(E) = & \hat{\pi}_0 \ln(\hat{\pi}_0) + \hat{\pi}_1 \ln(\hat{\pi}_1) & \text{déviance,} \\ I(E) = & \hat{\pi}_0 \hat{\pi}_1 & \text{indice de Gini,} \\ I(E) = & \min(\hat{\pi}_0, \hat{\pi}_1) & \text{erreur de classification.} \end{array}$$

On choisira donc la question qui optimise la réduction d'impureté

$$I(E) - \frac{n_{oui}}{n} I(E_{oui}) - \frac{n_{non}}{n} I(E_{non}) \quad ,$$

où n_{oui} et n_{non} sont les tailles des échantillons E_{oui} et E_{non}. Une fois la première question définie, on peut appliquer aux deux sous-échantillons résultants E_{oui} et E_{non} la même procédure, et ainsi de suite jusqu'à ce qu'en chaque feuille les observations soient toutes de même label.

Dans cette partie, l'indicateur de pureté considéré est le taux d'erreur de classification. On peut ainsi voir la construction d'un arbre comme une heuristique pour déterminer le minimiseur du risque empirique, et utiliser les résultats des parties précédentes.

Par définition, l'arbre complet ne fait aucune erreur de classement sur les données d'entraînement (sauf cas d'égalité)). De manière générale, on peut toujours construire un arbre de taille n discriminant parfaitement les n observations d'entraînement. Ces deux remarques illustrent clairement la possibilité que l'arbre complet puisse être surajusté. Il faut alors l'élaguer, ou plus simplement choisir entre l'arbre complet et ses sous-arbres. De toutes les règles proposées pour sélectionner un sous-arbre optimal, seule la procédure de sélection proposée par Breiman et ses coauteurs est aujourd'hui couramment employée. Cette procédure consiste à considérer une suite de sous-arbres emboîtés $T_1 \preceq T_2 \preceq ... \preceq T_{max}$, où T_{max} est l'arbre complet, et à choisir le sous-arbre T_{opt} minimisant un critère pénalisé de la forme

$$L_n(T_i) + \alpha |T_i| \quad , \tag{4.11}$$

où $|T_i|$ est la taille de l'arbre T_i et où α est choisi par validation croisée Q-fold (Q étant généralement fixé à 10). Le lecteur intéressé consultera Breiman *et al.* (1984) pour une présentation détaillée de la démarche. La procédure de Breiman a fait l'objet de plusieurs études théoriques dans le cadre de la sélection de modèles, on se reportera par exemple aux travaux de Nobel (2002) et de Gey et Nedelec (2005). Dans la suite, nous désignerons cette procédure par (P).

4.4.2 L'approche sélection de variables

La règle de classification que produit CART ne dépend que d'un sous-ensemble des variables initiales. Dans le cas de données de grande dimension où $p >> n$, cette sélection devient sévère puisque le classificateur final ne dépendra au plus que de $n-1$ variables. C'est pourquoi plusieurs auteurs considèrent l'algorithme CART comme une méthode intégrée de sélection de variables (Guyon et Elisseeff (2003), Lal *et al.*). Toutefois, lors de l'étape de sélection de l'arbre, il n'apparaît pas que la sélection de variables soit prise en compte. En effet, dans le critère pénalisé (4.11) précédent, la pénalisation ne prend explicitement en compte que la taille de l'arbre $|T|$, c'est-à-dire la complexité de la règle considérée. Le fait que l'arbre est choisi parmi tous les arbres de taille $|T|$ qu'il soit possible de construire à partir de p variables ne semble pas pris en compte. En particulier, le paramètre p n'apparaît pas dans la pénalisation.

En appliquant les résultats de la partie 4.2, nous allons montrer que l'étape de sélection de variables est en réalité prise en compte dans la stratégie de pénalisation par validation croisée Q-fold à travers le coefficient α, et que la méthode CART est effectivement une méthode de sélection de variables intégrée.

Nous commençons par définir les classes de classificateur que nous allons considérer. Un arbre est défini par sa configuration (c'est-à-dire la disposition de ses noeuds), et par les variables correspondant à chaque noeud. Pour une taille d'arbre $|T|$ fixée, le nombre de configurations différentes est

$$N_{conf} = \frac{1}{N+1} \mathrm{C}_{2N}^{N} \quad ,$$

où $N = 2|T| - 1$ est le nombre total de noeuds de l'arbre. Pour un arbre donné, il faut ensuite déterminer en chaque noeud la variable qui va être utilisée pour établir la question. Il y a donc pour une configuration donnée $p^{|T|-1}$ choix possibles de variables. Contrairement à la partie précédente, nous comptons ici le nombre de listes de variables ordonnées avec remise que l'on peut construire à partir de p variables. La liste est avec remise car une variable peut servir à la construction de plusieurs noeuds. La liste est ordonnée car une même liste de variables peut générer des arbres différents, suivant la manière d'associer une variable à un noeud. On définit ainsi une classe $\mathcal{C}_{|T|,k,\ell}$ comme l'ensemble des arbres à $|T|$ feuilles de configuration k fixée, construits à partir de la liste ordonnée de variables ℓ, où $k = 1,...,N_{conf}$ et $\ell = 1,...,p^{|T|-1}$. Dans une classe donnée, les seuls éléments restant à fixer pour spécifier complètement l'arbre sont donc les seuils associés à chaque noeud. Avec cette définition des classes, le nombre de classes d'arbres de même taille $|T|$ est donc

$$N_{|T|} = \frac{1}{N+1} \mathrm{C}_{2N}^{N} p^{|T|-1} \quad ,$$

Pour appliquer le théorème 4.2.1, il nous faut encore mesurer la complexité de chacune des classes. Nous proposons ici une majoration de cette complexité par la dimension de Vapnik d'un arbre. Une estimation plus raffinée de cette complexité peut être trouvée dans Gey et

Nedelec (2005). De manière triviale, il est toujours possible de trouver dans la classe $\mathcal{C}_{|T|,k,\ell}$ un arbre qui sépare parfaitement $|T|$ points, quelle que soit la liste ℓ considérée. La dimension de Vapnik de cette classe est donc $|T|$, et on a :

$$H_{|T|,k,\ell} \leq (|T|) \log n \ .$$

Nous pouvons donc maintenant utiliser le théorème 4.2.1 pour proposer une pénalité adaptée pour l'algorithme CART. D'après l'expression 4.4, pour un arbre $|T|$ la pénalité doit être de la forme

$$pen(\mathcal{C}_{|T|,k,\ell},n) = K_1 \frac{(|T|) \log n}{n} + K_2 \frac{\log\left((N+1)^{-1} \mathrm{C}_{2N}^{N} p^{|T|-1}\right)}{n} \quad .$$

On a :

$$\begin{aligned} pen(\mathcal{C}_{|T|,k,\ell},n) &\leq K_1' \frac{|T| \log n}{n} + K_2 \frac{\log\left((N+1)^{-1} \mathrm{C}_{2N}^{N} p^{|T|-1}\right)}{n} \\ &\leq K_1' \frac{|T| \log n}{n} + K_2 \frac{|T| \log(p) + \log\left(\mathrm{C}_{2N}^{N}\right) - \log(N+1)}{n} \\ &\leq K_1' \frac{|T| \log n}{n} + K_2' \frac{|T| \log p}{n} + 2N \log(\frac{2N}{N}) \\ &\leq \left(K_1'' \frac{\log n}{n} + K_2'' \frac{\log p}{n} \right) |T| \quad , \end{aligned}$$

puisque $N = 2|T| - 1$. Ainsi, dans le cadre de la classification pour l'algorithme CART, on choisira l'arbre de classification minimisant un critère de la forme

$$L_n(T) + \left(K_1'' \frac{\log n}{n} + K_2'' \frac{\log p}{n} \right) |T| \quad . \tag{4.12}$$

Si l'on compare cette expression à la procédure de pénalisation (4.11), on remarque que l'on retrouve ici une pénalisation du risque empirique par un terme proportionnel à la taille de l'arbre. La procédure de sélection (P) est donc parfaitement valide dans le cadre d'une sélection simultanée de la complexité et des variables de l'arbre. Plus précisément, l'expression (4.12) montre que la constante de multiplication α de la procédure (P) doit être une fonction affine de $\log p$. Cette relation particulière entre le paramètre p et la constante α va maintenant nous permettre de valider à la fois la forme de la pénalité (4.12) et la procédure CART comme méthode de sélection de variables intégrée.

4.4.3 Simulations

Nous cherchons à confirmer les résultats de la partie précédente par une étude de simulation. Le principe général de cette étude est le suivant. On commence par simuler un problème de classification simple, où le label Y d'un individu dépend de 3 variables X^1, X^2 et X^3. Puis l'on ajoute à ces 3 variables un certain nombre de variables supplémentaires N_{suppl}, non informatives. Le nombre total de variables est donc $p = N_{suppl} + 3$, et lorsque N_{suppl} augmente, la quantité d'information utile pour le classement reste constante. Pour un nombre total de variables p donné, on simule 400 échantillons. On réalise ensuite pour chaque échantillon la construction de l'arbre, puis son élagage suivant la procédure (P). Chaque procédure d'élagage conduit à l'estimation par validation croisée Q-fold du paramètre α. L'objectif est de calculer la valeur moyenne de α pour différentes valeurs de p, puis d'étudier le comportement de cette valeur moyenne en fonction de p. Nous présentons ici les résultats obtenus à partir de 3 méthodes différentes de simulation des données.

Plan de simulation "CART"

Pour ces simulations, les variables simulées sont indépendantes, et de même loi $\mathcal{N}(0,1)$. Le label est déduit des variables X^1, X^2 et X^3 de la manière suivante :

- si $X^1 > 0.2$ et $X^2 > 0$, alors $Y = 1$,
- si $X^1 < 0.2$ et $X^3 > 0$, alors $Y = 1$,
- sinon $Y = 0$.

On bruite ensuite les données en leur ajoutant un bruit blanc de variance σ^2. Comme on le voit, la règle de classification optimale est exactement de la forme des règles de décision de l'algorithme CART, i.e. le classificateur de Bayes fait partie de l'une des classes d'arbres considérées. Le problème de classification est donc facile.

Plan de simulation "Somme"

Dans ce deuxième plan, les variables sont simulées de la même manière que précédemment. Seule la règle pour générer les labels est différente :

- si $X^1 + X^2 + X^3 > 0$, alors $Y = 1$,
- sinon $Y = 0$.

Les données sont ensuite bruitées comme dans le plan CART. Cette fois, la règle de classification optimale n'a pas la forme d'une règle de décision CART. Le problème de classification est donc plus difficile que dans le cas précédent.

Plan de simulation "Correlations"

Dans ce troisième plan, les 3 variables informatives sont simulées de manière indépendante et suivent des lois $\mathcal{N}(0,1)$. On crée ensuite une variable moyenne

$$X_4 = (X^1 + X^2 + X^3)/3 \ .$$

Chaque variable supplémentaire X_i est ensuite simulée de la manière suivante :

$$X^i = X^4 + \varepsilon_i \quad ,$$

où ε_i est un bruit blanc de variance σ^2. Les données supplémentaires sont donc faiblement corrélées avec les variables informatives, et fortement corrélées entre elles. Enfin, la règle d'appartenance est

- si $(X^1)^2 + (X^2)^2 + (X^3)^2 > 2.5$, alors $Y = 1$,
- sinon $Y = 0$.

Le problème est donc compliqué, car les données sont corrélées et la règle d'appartenance est difficile à estimer par un arbre.

Résultats

Nous montrons ici les résultats obtenus pour $n = 100$ individus par échantillon et $\sigma^2 = 0.2$ (d'autres valeurs de ces paramètres donnent des résultats similaires). Les résultats sont représentés en figure 4.2.

Chaque graphique correspond à l'un des trois plans de simulation. L'abscisse des points représente le logarithme du nombre total de variables p, et l'ordonnée la valeur moyenne du paramètre α. On observe clairement une relation linéaire entre la valeur α sélectionnée et le

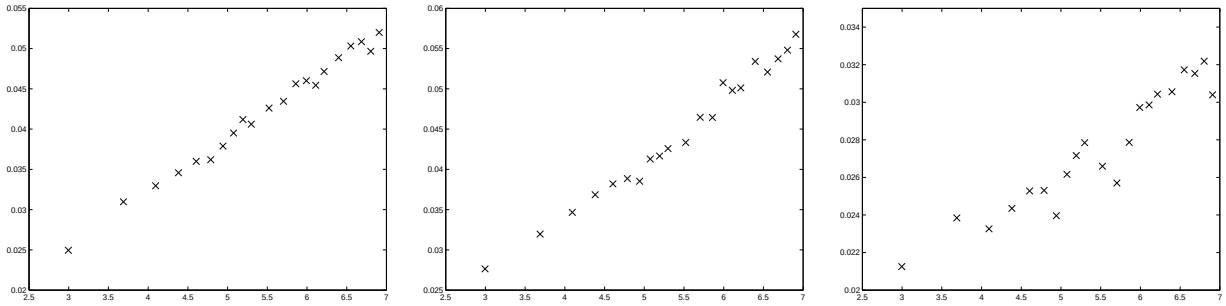


FIG. 4.2 – *Valeur moyenne de α (en ordonnée) en fonction du logarithme du nombre de variables p (en abscisse), pour les 3 types de simulation*

logarithme du nombre de variables. Cette observation apparaît moins clairement pour le plan de simulations "Corrélations" car la variabilité des coefficients α choisis est plus grande. En augmentant le nombre de simulations jusqu'à 1000, on obtient une courbe comparable à celles obtenues pour les deux autres plans.

La première conséquence de cette observation est que la procédure (P) est une vraie étape de régularisation qui tient compte à la fois de la complexité de l'arbre mais aussi de la sélection de variables. La sélection de variables n'est pas explicite, car le terme de pénalisation prenant en compte le nombre de classes de même complexité est proportionnel à $|T|$. Ce terme apparaît donc implicitement dans l'étape d'estimation du paramètre α, comme le montre la figure 4.2. Ceci démontre que l'algorithme CART, au même titre que la méthode JCFO présentée en introduction, est une méthode de classification avec sélection de variables intégrée.

La seconde conséquence est que le décompte des classes proposé est validé. En partie 4.2, chaque classe $\mathcal{C}_{k,j,\ell}$ est en particulier définie par la liste ℓ des j variables intervenant dans la construction des classificateurs. On considérait alors des listes sans remise et non ordonnées. A l'inverse, nous avons proposé pour CART de considérer les listes de variables comme ordonnées, avec remise. La relation linéaire entre α et $\log(p)$ observée confirme cette manière de compter les modèles. Cette originalité, particulière à CART, vient du fait qu'une même variable peut être associée à plusieurs noeuds.

4.5 Discussion

Afin d'obtenir une inégalité oracle minimax, nous avons ici utilisé un résultat de Massart (2000). Nous avons vu que l'utilisation de ce résultat impose le choix d'une fonction de perte quadratique. De ce fait, le risque est ici défini comme une distance moyenne à la fonction de régression Φ^*, et non au classificateur de Bayes Φ^B. Il est possible d'obtenir le même type de résultat en prenant la fonction de perte $hard-loss$ usuelle, et en s'appuyant sur des résultats de sélection de modèles plus récents, comme ceux disponibles dans Massart et Nédélec (2006) par exemple. Du point de vue de la sélection de variables le résultat obtenu serait comparable à celui du théorème 4.2.1, au sens où l'on montrerait encore que la pénalité doit explicitement prendre en compte le nombre de modèles de même dimension visité, et que le terme $\dfrac{\log \mathrm{C}_p^j}{n}$ peut être un terme prépondérant lorsque le nombre de variables est grand.

Il est important de noter que la démarche reste applicable lorsque l'on passe à d'autres fonctions de perte convexes. On sait en effet qu'en pratique la plupart des algorithmes existants

minimisent non pas une fonction de perte de type $hard - loss$, mais des fonctions de perte convexes. C'est en particulier le cas des SVM, où la fonction de perte utilisée est la fonction $hinge - loss$ (Smola et Schölkopf (1998)). Le résultat présenté ici pourrait être étendu au cas de la perte $hinge - loss$, en s'appuyant par exemple sur les travaux de Blanchard *et al.*, où les auteurs proposent une interprétation de type sélection de modèles de l'algorithme SVM.

Le résultat obtenu ici, bien qu'optimal du point de vue minimax, n'est pas directement utilisable. En effet les constantes qui apparaissent dans la pénalisation sont *a priori* bien trop grossières pour une application directe du résultat. Cette difficulté est récurrente dans les résultats de sélection de modèles, et diverses solutions ont été proposées dans la littérature. Les plus courantes sont basées sur la calibration de constantes : les constantes qui apparaissent dans la pénalité étant des constantes universelles, il est possible de les estimer en se basant sur des simulations. Cette possibilité a notamment été explorée par Birgé et Rozenholc (2002), où une première heuristique de calibration est proposée, puis par Birgé et Massart (2001b) et Gey et Lebarbier (2002), qui utilisent l'heuristique de pente pour la calibration. Ces heuristiques pourraient ici être adaptées pour obtenir une estimation plus fine des constantes K_1' et K_2'.

Enfin, les résultats présentés dans ce chapitre devraient aussi trouver une application pratique. Nous avons montré en partie 4.3 comment obtenir des bornes supérieures et inférieures pour les algorithmes de sélection de variables séquentiels. Ces bornes peuvent se révéler assez grossières, mais il n'existe actuellement aucun résultat analytique sur la manière de fixer le critère d'arrêt pour les méthodes backward et forward. Le nombre de variables sélectionnées est en général choisi par rééchantillonnage, qui est une procédure coûteuse en termes de temps de calcul. Le fait de disposer de bornes pour le critère d'arrêt pourrait donc servir à réduire la fenêtre des valeurs pour lesquels le rééchantillonnage est réalisé, et ainsi faire gagner un temps de calcul considérable.

Le théorème 4.2.1 représente aussi un cadre rigoureux pour l'analyse et le développement de méthodes de sélection intégrées. L'approche par sélection de modèles nous a déjà permis d'expliquer comment la sélection de variables est réalisée dans l'algorithme CART, et comment la procédure de régularisation usuelle prend implicitement le nombre de modèles visités en compte dans la constante de pénalisation. Il est bien sûr possible d'appliquer notre raisonnement à l'analyse d'autres méthodes de sélection de variables. Si l'on reprend l'article de Weston *et al.* (2000) par exemple, on constate que la sélection de variables proposée est basée sur une pénalisation du risque par la norme ℓ_0, ce qui revient à compter le nombre de variables. On constate que dans une telle pénalisation le nombre de modèles de même dimension n'est pas pris en compte. Les méthodes qui dérivent de la pénalisation ℓ_0 pourraient donc être modifiées en prenant en compte la forme du critère obtenue ici.

Chapitre 5

Agrégation supervisée

5.1 Introduction

Les limites de la sélection de variables

Dans la partie 2, nous avons présenté les motivations classiquement avancées par les auteurs pour faire de la sélection de variables. Trois arguments étaient proposés :

- La sélection de variables, en prenant en compte la complexité du classificateur construit, va permettre de choisir un classificateur plus performant et plus robuste du point de vue du taux d'erreur,
- Travailler avec une règle simplifiée devrait réduire les temps de calcul,
- En réduisant considérablement le nombre de variables, on obtiendra une règle de décision plus simple à interpréter, i.e. les variables gardées seront des variables explicatives autant que prédictives.

De ces trois objectifs, seul les deux premiers nous sont apparus accessibles. L'inégalité oracle démontrée au chapitre 4 apporte une validation théorique de l'efficacité de la stratégie de sélection de variables. Du point de vue empirique, les résultats très satisfaisants obtenus par plusieurs auteurs montrent que cette stratégie est valable, et permet d'améliorer le taux d'erreur de classement et les temps de calcul.

Il apparaît que l'objectif d'interprétation n'est pas accessible par la sélection de variables. Cette constatation est autant théorique que pratique. Du point de vue théorique, nous avons montré l'intérêt de construire des critères de sélection de variables adaptatifs. Ce choix ne se justifie que par l'objectif d'amélioration du taux d'erreur du classificateur, et nulle part l'objectif d'interprétation n'est pris en compte. Du point de vue pratique, on peut se reporter à l'article de Michiels *et al.* (2005). Dans cette étude, les auteurs s'intéressent aux expériences de biopuces consacrées au diagnostic du cancer publiées dans les années 1995-2003. Chacune de ces expériences avait pour objectif de construire une règle de classification à partir d'un échantillon d'entraînement, et de valider cette règle en estimant son taux d'erreur de classement sur un échantillon test. En reprenant ces expériences et en rééchantillonnant les données d'entraînement et de test, Michiels *et al.* ont montré que les listes des gènes identifiés comme prédicteurs du cancer sont instables, et changent radicalement en fonction du rééchantillonnage. Cette expérience montre qu'en pratique, la sélection de variables ne garantit en aucun cas que la règle de classification est interprétable lorsque l'on travaille avec des données de grande dimension.

Instabilité et redondance

Afin de déterminer ce qui rend les méthodes de sélection de variables instables, nous reprenons le raisonnement présenté au chapitre 2 concernant les méthodes de type Filter.

Les méthodes Filter sont développées selon un double objectif : évincer les variables non informatives, et gérer les problèmes de redondance. Identifier les variables non informatives et les retirer est un objectif pertinent. Mis à part le fait que l'identification de ces variables peut être plus ou moins aisée, cette étape ne peut *a priori* pas rendre compte de l'instabilité de la liste des variables sélectionnées, et donc des difficultés d'interprétation de la règle. L'incapacité des méthodes de sélection de variables à rendre la règle de décision simple à interpréter tient donc essentiellement au traitement de la redondance. En effet, si deux variables apportent la même information, seul l'échantillonnage guidera le choix de l'une par rapport à l'autre. Lorsque les données sont de grande dimension, la redondance peut être très forte, et les cas d'ex aequo entre variables fréquents. Les données d'Alon illustrent ce problème. La Figure 5.1 présente le profil de quatres gènes, tous classés dans la liste des 10 gènes les plus différentiellement exprimés dans l'expérience. On constate que les profils sont très similaires, et les coefficients de corrélation entre ces quatre gènes sont tous supérieurs à 0.85. Lors d'une étude de classification, et suivant le découpage en échantillon d'entraînement / échantillon test, l'un de ces gènes sera sélectionnée, et les autres disqualifiées. On explique ainsi l'instabilité décrite dans l'article de Michiels *et al.* (2005).

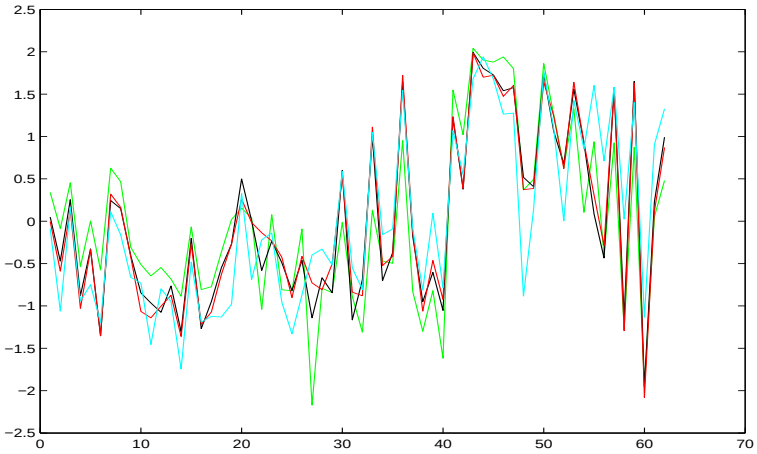


FIG. 5.1 – *Profils d'expression normalisés des gènes classés 3 et 11 dans l'expérience d' Alon. Source : Krishnapuram* et al. *(2004a)*

Un nouveau traitement de la redondance

Dans l'argumentaire développé dans les articles de sélection de variables filter, les variables redondantes son présentées comme un facteur potentiel de dégradation des performances du classificateur. C'est pourquoi les variables redondantes doivent être évincées, au même titre que les variables non informatives (cf. chapitre 2.3.1). Ce point de vue est clairement illustré par l'exemple présenté dans l'article de Yu et Liu (2004a), que nous traduisons ici : "Soient X^1,...,X^5 des variables booléennes. On suppose que la règle de classification est une fonction de X^1 et X^2. Par ailleurs, on suppose que $X^2 = \bar{X^3}$ et que $X^4 = \bar{X^5}$. Pour classer, X^1 est

indispensable, l'une des deux variables X^2, X^3 peut être supprimée à condition de garder l'autre, et l'on peut supprimer les variables X^4 et X^5, qui ne sont pas informatives. En présence de milliers de variables, [...] l'apprentissage sera amélioré si l'on se restreint à garder des variables pertinentes et non redondantes."

Il est clair qu'un tel raisonnement ne prend pas en compte le fait que les données sont généralement bruitées. Dans l'exemple présenté, il suffirait de rajouter un bruit centré autour de chaque variable pour conclure que la moyenne des variables X^2 et $\bar{X}^3$ serait plus efficace pour la prédiction que de considérer l'une d'entre elles uniquement. En outre, plusieurs variables présentant le même profil désignent une tendance marquée : l'information est plus fiable lorsqu'elle est corroborée par plusieurs sources. Cette tendance est donc susceptible d'être aussi exprimée dans de futures observations à classer.

Si la redondance peut être utilisée avec profit pour améliorer les performances d'un algorithme de classification et gagner en clarté d'interprétation, nous avons vu qu'il est difficile de la prendre en compte dans l'étape de sélection de variables. Notre objectif est donc de découpler le traitement des variables redondantes de l'étape de sélection de variables. On peut résumer ce point de vue de la manière suivante : dans l'exemple de Yu et Liu, il existe 3 groupes distincts de variables. La variable X^1 forme à elle seule un groupe unique, apportant une information pertinente pour le label. Le deuxième groupe comporte les variables X^2 et X^3, qui apportent une même information redondante sur le label. Enfin, le troisième groupe contient les variables X^4 et X^5, non informatives pour le label. La méthode de traitement de la redondance a donc pour tâche de retrouver ces trois groupes, et de synthétiser l'information contenue dans chacun d'entre eux, par exemple en prenant les moyennes de X^2 et $\bar{X}^3$, et de X^4 et $\bar{X}^5$ respectivement pour les groupes 2 et 3. L'étape de sélection servira ensuite à éliminer le troisième groupe, non informatif pour le label. Nous concevons donc ici le traitement de la redondance comme une étape préalable et complémentaire à la sélection de variables, et non comme une méthode alternative. Pratiquement, cette étape consistera en deux opérations : regrouper les variables redondantes du point de vue de l'information qu'elles apportent sur le statut, et synthétiser l'information portée par chaque groupe de variables. Nous parlerons de méthodes d'agrégation supervisée, traduction du terme "supervised clustering" proposé dans Dettling et Bühlmann (2002). On prendra garde de distinguer dans la suite

- les méthodes d'agrégation supervisée (notre méthode), qui visent à regrouper des variables en fonction de leur information sur le label,
- les méthodes d'agrégation ou de classification classiques, qui agrègent en l'absence de superviseur,
- la classification supervisée, où l'on cherche à prédire le label d'un individu à partir des variables disponibles,
- la compression (supervisée ou non), où l'on ne cherche pas à former des groupes disjoints de variables.

Quel type d'agrégation supervisée ?

Le regroupement de variables est un problème classique en statistique : on trouve déjà dans les années 70 plusieurs articles sur le sujet (Anderberg (1973), Harman (1973)). En revanche, le problème du regroupement de variables supervisé n'est intensivement étudié que depuis une période récente. Cet intérêt récent s'explique, comme pour les méthodes de sélection de variables, par la nécessité pour le statisticien de traiter des jeux de données caractérisés par un nombre de variables de plus en plus grand. Nous détaillons ici quelques uns des travaux

sur le sujet.

L'une des premières méthodes d'agrégation supervisée fut proposée par Hastie *et al.* (1999). Dans cet article, les auteurs présentent une méthode basée sur une modification de leur algorithme de GeneShaving. L'algorithme construit une combinaison linéaire des variables très corrélées avec le statut. La combinaison linéaire est ensuite "nettoyée" (c'est la partie shaving), c'est-à-dire que les variables les moins corrélées avec la combinaison obtenue sont retirées. On construit ensuite une seconde combinaison linéaire orthogonale à la première, et ainsi de suite.

Dettling et Bühlmann (2002) et Dettling (2003) présentent deux algorithmes, Wilma (pour *Wil*coxon and *Ma*rgin criteria) et Pelora (pour *Pe*nalized *Lo*gistic *R*egression *A*nalysis), tous deux basés sur l'objectif de créer de petits groupes de variables à caractère hautement prédictif. Le caractère prédictif d'une variable ou groupe de variables est basé sur le critère donnant le nom à la méthode : la statistique de Wilcoxon ou la régression logistique pénalisée. Un petit nombre de groupes est ensuite utilisé avec différents algorithmes de classification.

Le travail de Jornsten et Yu (2003), bien que souvent présenté comme un exemple d'agrégation supervisée, ne fait pas partie à proprement parler de ce type de méthode. Les auteurs proposent un cadre unifié pour réaliser, dans une même étape, un regroupement non supervisé des variables et une sélection des meilleurs groupes pour la prédiction. L'avantage de ce cadre est qu'il permet aux auteurs d'élaborer un critère pénalisé de sélection de modèles qui permet de sélectionner simultanément un modèle à K groupes et les meilleurs barycentres de groupes pour prédire. Toutefois, le statut n'intervient pas dans la création des groupes.

L'algorithme de Xu et Zhang réalise itérativement un regroupement de variables puis une sélection des meilleurs groupes, et fonctionne, selon les auteurs, comme une sélection Backward. L'homogénéité des groupes est mesurée par le critère Silhouette, et leur pouvoir discriminant par leur coefficient dans l'hyperplan obtenu en utilisant l'algorithme SVM linéaire.

Enfin, Diaz-Uriarte (2004) propose aussi une méthode où les variables sont regroupées en "composants" (signature components). Les composants sont créés un à un, et doivent être informatifs vis-à-vis du statut : un composant non relié au statut n'est pas pris en compte.

Quels sont les points communs entre ces méthodes ? Le premier est que toutes combinent le regroupement et la sélection des groupes. Traiter spécifiquement la redondance des variables, en dehors de toute considération de leur pertinence, n'est donc pas l'objectif de ces méthodes. On peut à ce propos remarquer que la manière dont la redondance est traitée par les différents algorithmes proposés n'est pas toujours claire. L'autre point commun existant est que toutes les méthodes décrites (exception faite de Jornsten et Yu (2003)) sont de type filter : l'étape de regroupement ne nécessite pas la connaissance de l'algorithme de classification qui sera ensuite appliqué. Bien qu'il n'y ait pas *a priori* de supériorité des méthodes wrapper sur les méthodes filter, cet aspect pose ici problème. En effet, il n'y a pas de cohérence entre la méthode de création des groupes et la méthode de sélection des groupes. Il semble par exemple difficile de justifier la combinaison du critère Silhouette et de la classification SVM dans le travail de Xu et Zhang.

Comme les méthodes présentées ci-dessus, la méthode que nous proposons repose sur le principe de regroupement des variables, combinée à une étape de sélection des groupes. Les deux spécificités de notre méthode sont les suivantes :

- l'objectif de la méthode proposée sera de regrouper des variables redondantes concernant le statut, quelle que soit leur pertinence.
- la méthode sera de type Wrapper : le critère de redondance, ainsi que la manière de synthétiser l'information dans chaque groupe, sera basée sur l'analyse de la méthode de classification choisie. Cela assurera la cohérence entre la méthode d'agrégation supervisée

et la méthode de sélection Wrapper utilisée ensuite.

Nous présentons la stratégie de traitement de la redondance pour la méthode de classification kNN. Cette première application sera l'occasion de présenter la stratégie globale d'agrégation supervisée. En partie 5.2 nous définissons la notion de redondance pour les kNN, ainsi que l'objectif d'agrégation, présenté comme un programme de maximisation de critère. La partie 5.3 est consacrée à une première étude de la méthode sur données simulées. La partie 5.4 montre les résultats obtenus en appliquant la méthode à des données fonctionnelles et des données génomiques. Les extensions possibles de la méthode d'agrégation présentée à d'autres distances et au cadre de la régression sont présentées en partie 5.5. Enfin, nous montrons en partie 5.6 comment la stratégie de traitement de la redondance peut être appliquée à d'autres méthodes de classification supervisée, et nous présentons l'agrégation pour l'algorithme CART.

5.2 Agrégation supervisée pour les kNN

5.2.1 Le programme de minimisation

On rappelle que l'algorithme des kNN procède comme suit : lorsque l'on désire classer un nouveau point, on détermine les k observations les plus proches de ce point dans l'échantillon d'entraînement, et on classe suivant le label majoritaire parmi ses voisins. Il y a donc formellement deux étapes : la recherche des voisins, appelée étape de tesselation, et la révélation de leurs labels. Le point crucial est ici que les variables X^1,...,X^p n'interviennent que dans la première étape, et les labels Y uniquement dans la deuxième. On va ainsi pouvoir définir la redondance entre variables en ne considérant que l'étape de tesselation : deux variables induisant les mêmes tesselations induiront les mêmes classifications. Ceci nous permet de définir la redondance pour les kNN :

Définition 5.2.1. *Deux variables sont redondantes pour l'algorithme kNN si elles engendrent les mêmes contributions aux distances entre individus.*

On constate ici une spécificité des kNN : bien que la redondance étudiée soit en principe une redondance de l'information pour le label Y, elle ne s'exprime qu'en fonction des variables explicatives X. L'objectif de la méthode d'agrégation proposée sera donc d'agréger les variables en conservant au mieux les distances entre individus.

Afin de décrire rigoureusement l'objectif de traitement de la redondance pour les kNN, nous introduisons maintenant les notions de répartition et d'agrégation. On appelle répartition et l'on note $\mathcal{C} = \mathcal{C}_1 \bigcup \mathcal{C}_2 ... \bigcup \mathcal{C}_{N_C}$ une répartition des p variables initiales en N_C groupes disjoints. Nous pouvons alors définir une agrégation comme suit :

Définition 5.2.2. *On appelle agrégation une procédure consistant à choisir une répartition en N_C groupes et à résumer l'information contenue dans chaque groupe $\mathcal{C}_i$ par une combinaison linéaire des variables du groupe :*

$$Z^{\mathcal{C}_i} = \sum_{X^j \in \mathcal{C}_i} \alpha_j X^j \quad .$$

Nous cherchons donc l'agrégation conservant au mieux les distances entre individus. Pour quantifier la conservation des distances, on utilise la notion d'inertie engendrée par un en-

semble de variables :

$$I(X^1,...,X^p) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n} d^2(X_i,X_j) \quad .$$

Bien que d'autres mesures globales sur les distances existent, l'inertie est certainement la mesure la plus employée par les statisticiens. Plusieurs méthodes populaires, comme l'Analyse en Composantes Principales (ACP) ou la classification non supervisée par distance de Ward sont basées sur l'inertie, et les propriétés de cette mesure ont été largement étudiées (voir Saporta (1990) par exemple). En particulier, on sait que l'inertie d'un nuage de points diminue lorsque l'on agrège les variables, cette propriété nous permettant de définir la perte pour les kNN :

Définition 5.2.3. *La perte kNN pour une agrégation de taille N_C donnée est définie par la perte d'inertie engendrée par cette agrégation*

$$P_{kNN}\left(\left\{X^1,...,X^p\right\}, \left\{Z^{\mathcal{C}_1},...,Z^{\mathcal{C}_{N_C}}\right\}\right) = I(X^1,...,X^p) - I(Z^{\mathcal{C}_1},...,Z^{\mathcal{C}_{N_C}}) \quad .$$

A partir de cette perte et de la définition de la procédure d'agrégation que nous avons données, nous pouvons donc décrire l'objectif de traitement de la redondance pour les kNN comme un programme de minimisation de la manière suivante :

Proposition 4. *Soient $X_1,...,X_n$ n observations, pour lesquelles p variables $X^1,...,X^p$ sont mesurées. Soit N_C un entier fixé tel que $N_C < p$. Trouver le meilleur regroupement de variables pour l'algorithme des kNN consiste à résoudre le programme de minimisation*

$$\min_{\mathcal{C}\in\mathbb{A}} P_{kNN}\left(\left\{X^1,...,X^p\right\}, \left\{Z^{\mathcal{C}_1},...,Z^{\mathcal{C}_{N_C}}\right\}\right) \quad , \tag{5.1}$$

où $\mathbb{A}$ est l'ensemble des partitions de $\{X^1,...,X^p\}$ en N_C groupes.

Le programme de minimisation est ici défini de manière générale, mais la résolution pratique de ce programme de minimisation passe par la spécification d'une distance entre individus. Nous supposons maintenant que la distance d est la distance euclidienne canonique, appliquée aux données X centrées réduites :

$$d^2(X_i,X_j) = \sum_{\ell=1}^{p}(X_i^\ell - X_j^\ell)^2 \quad .$$

L'inertie calculée en utilisant cette distance est alors additive pour les variables :

$$\begin{aligned}
I(X^1,...,X^p) &= \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n} d^2(X_i,X_j) \\
&= \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{p}(X_i^k - X_j^k)^2 \\
&= \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N_C}\sum_{X^{k_1},...,X^{k_q}\in\mathcal{C}_k}(X_i^{k_1} - X_j^{k_1})^2 \\
&= \sum_{k=1}^{N_C}\sum_{X^{k_1},...,X^{k_q}\in\mathcal{C}_k} I(X^{k_1},...,X^{k_q})
\end{aligned}$$

Par conséquent, cette propriété d'additivité s'applique aussi à la perte kNN, ce qui nous permet de simplifier le problème (5.1) par :

$$\begin{aligned} & \min_{\mathcal{C}\in\mathbb{A}} \sum_{i=1}^{N_C} \sum_{X^{i_1},\ldots,X^{i_q}\in\mathcal{C}_i} P_{kNN}\left(\left\{X^{i_1},\ldots,X^{i_q}\right\}, Z^{\mathcal{C}_i}\right) \\ = & \min_{\mathcal{C}\in\mathbb{A}} \sum_{i=1}^{N_C} \sum_{X^{i_1},\ldots,X^{i_q}\in\mathcal{C}_i} P_{kNN}\left(\left\{X^{i_1},\ldots,X^{i_q}\right\}, X^{\mathcal{C}_i}\right) \quad , \end{aligned} \tag{5.2}$$

où $X^{\mathcal{C}_i}$ est la première composante principale de l'ACP réalisée sur les variables du groupe $\mathcal{C}_i$, cette dernière simplification (5.2) provenant directement de l'application des résultats classiques obtenus en ACP.

En théorie, l'objectif est donc parfaitement décrit par le problème de minimisation de l'équation 5.2. En pratique, se posent toutes les difficultés classiques liées à la résolution d'un tel problème : la complexité algorithmique du programme de minimisation et le choix du paramètre N_C. Pour le premier point, nous présentons dans la partie suivante une adaptation de l'algorithme de classification hiérarchique ascendante (CAH) pour la résolution heuristique du problème de maximisation. Pour le deuxième point, nous présentons en partie 5.4.1 une méthode de sélection du nombre de groupes basée sur la méthode de resubstitution K-fold.

5.2.2 Algorithme de regroupement

La résolution du problème de minimisation (5.1) passe par l'exploration de l'ensemble des répartitions possibles de p variables en N_C groupes. On sait que le nombre de ces répartitions est très grand : on dénombre par exemple plus de 10^{47} répartitions de 100 variables en 3 groupes. Il n'existe donc pas de méthode algorithmique susceptible de visiter l'ensemble des répartitions en un temps raisonnable, en particulier lorsque le nombre de variables est grand. On utilise alors des algorithmes permettant d'approcher la répartition optimale. Plusieurs méthodes ont été proposées, les plus célèbres étant les nuées dynamiques (K-means) ou la classification hiérarchique ascendante (CAH). L'objectif étant de faciliter l'interprétation des résultats d'une classification supervisée, on recherche ici une méthode d'agrégation qui soit stable : le même algorithme appliqué plusieurs fois à un même jeu de données doit fournir des résultats identiques. On sait que l'algorithme des nuées dynamiques doit être initialisé par le choix de N_C centres de groupes, ce qui rend les résultats de cette méthode très instables. Nous choisirons donc ici la méthode CAH pour regrouper les variables. Nous présentons rapidement le principe général de cet algorithme.

La CAH est une méthode de classification itérative. A l'étape initiale, les p variables constituent des classes individuelles. On calcule les distances deux à deux entre variables, et les deux variables les plus proches sont réunies en une classe. La distance entre cette nouvelle classe et les $p-2$ variables restantes est ensuite calculée, et à nouveau les deux éléments (classes ou variables) les plus proches sont réunis. Ce processus est réitéré jusqu'à ce qu'il ne reste plus qu'une unique classe regroupant toutes les variables.

La mise en application de cette procédure nécessite donc la spécification d'une distance entre variables, et entre groupes de variables. La définition de ces distances est directement déduite du programme de minimisation (5.2), i.e. de la définition de la redondance 5.2.1 :

Définition 5.2.4. *La distance entre deux variables est mesurée par la perte d'inertie engendrée lorsque ces deux variables sont remplacées par le premier axe de leur ACP.*

L'algorithme procède donc comme suit : on calcule pour chaque couple de variables la perte d'inertie associée à leur agrégation. On remarque qu'à l'étape t on dispose de $p-t+1$ variables à classer dans $p-t$ groupes, et que l'on a la décomposition suivante :

$$\sum_{i=1}^{p-t+1}\sum_{j=1}^{t} d^2(X_i,X_j) = \sum_{i=1}^{p-t+1}\sum_{j=1}^{t}\sum_{k=1}^{p} d^2(X_i^k,X_j^k) \quad .$$

L'agrégation ne concernant que deux variables X^{k_1} et X^{k_1}, les sommes des carrés concernant les autres variables ne sont pas affectées, et la perte d'inertie I de l'étape considérée est mesurée par λ_2, la deuxième valeur propre associée à la matrice de variance-covariance des deux variables. Remplacer les deux variables X^{k_1} et X^{k_2} par le premier axe de leur ACP engendre la perte d'inertie minimale lorsque l'on agrège ces deux variables. De plus, on choisit à chaque étape le couple (X^{k_1}, X^{k_2}) de valeur propre associée λ_2 minimum parmi tous les couples possibles. On réalise ainsi la plus petite perte d'inertie possible à cette étape. La procédure est donc localement optimale : à chaque étape on réalise l'agrégation optimale au sens de la perte d'inertie, conditionnellement aux étapes précédentes.

La définition proposée pour la distance entre variables permet une interprétation simple de la perte à chaque étape. L'analyse de la perte peut servir à fournir un premier diagnostic sur la difficulté de la tâche d'agrégation. La partie suivante présente les différentes utilisations qui peuvent être faites de la courbe d'agrégation, tracée à partir de la perte à chaque étape.

5.2.3 La courbe d'agrégation

L'utilisation d'un algorithme d'agrégation suppose d'une part qu'il existe une redondance dans les données, et d'autre part que cette redondance est facilement détectable. Nous verrons dans la partie 5.3 qu'il est préjudiciable d'agréger les variables lorsque l'une de ces deux hypothèses n'est pas satisfaite. Idéalement, le choix d'agréger ou non les variables à disposition est basée sur la bonne connaissance que l'on a des données. Néanmoins, cette connaissance n'est pas toujours suffisante. Dans l'exemple des données de biopuces, on sait qu'il existe une grande redondance entre gènes : la survie d'une cellule repose sur des mécanismes (assurant la reproduction de la cellule, ou la synthèse de protéines essentielles à sa survie) entraînant une forte corégulation des gènes entre eux. Mais on sait aussi que la variabilité technique liée à cette technologie est forte, et peut perturber l'identification de cette corégulation. Dans de tels cas, on souhaite disposer de critères indiquant si la tâche d'agrégation est pertinente ou non. On peut alors se baser sur la courbe d'agrégation pour déterminer la difficulté de la tâche.

La courbe d'agrégation représente en ordonnée la perte d'inertie associée à l'étape t en fonction de t en abscisse. On fera bien attention au fait que lorsque le nombre d'étapes t augmente, le nombre de groupes de variables diminue. Cette courbe permet donc de voir l'évolution de la dégradation de l'inertie au fil des étapes, et de repérer un éventuel "décrochage" de la courbe, c'est-à-dire un point de la courbe à partir duquel la perte d'inertie est significativement plus forte qu'auparavant. Ce décrochage désigne alors l'étape à partir de laquelle on réalise des fusions entre variables peu redondantes, i.e. l'étape à partir de laquelle il n'est plus pertinent d'agréger. Un exemple de décrochage est présenté en partie 5.3.2.

On considère aussi la courbe d'agrégation cumulée, représentant pour chaque étape la perte d'inertie depuis le début du processus d'agrégation. La forme globale de cette courbe donne une idée de la difficulté de la tâche d'agrégation. S'il n'y a aucune redondance à traiter, toutes les agrégations engendrent le même ordre de perte d'inertie, et la courbe d'agrégation

cumulée est linéaire. A l'inverse, lorsque l'information est très redondante, les pertes d'inertie correspondant aux premières étapes (où l'on regroupe les variables redondantes) seront bien plus faibles que celles engendrées par les dernières étapes (ou l'on regroupe des variables non redondantes). La courbe d'agrégation cumulée sera alors creusée, comme sur la figure 5.2.

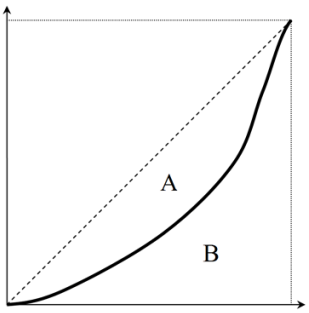


FIG. 5.2 – *Courbe d'agrégation, en trait plein. La droite en pointillé représente la courbe d'agrégation que l'on obtiendrait sous l'hypothèse qu'il n'y a pas de redondance. Le rapport des aires A et B mesure l'écart entre cette hypothèse et la réalité.*

L'incurvation de la courbe d'agrégation cumulée peut être mesurée par l'indice de Gini, défini graphiquement sur la figure 5.2 comme deux fois le rapport des surfaces A et B. L'indice varie entre 1 et 0, cette dernière valeur étant atteinte lorsque la courbe de concentration est très peu creusée, c'est-à-dire lorsque l'agrégation est difficile ou inutile. Dans la suite, nous nous servirons des deux courbes d'agrégation (cumulée et non cumulée) pour mesurer la difficulté de la tâche d'agrégation et proposer un critère d'arrêt pour l'algorithme d'agrégation.

5.3 Une première étude de l'agrégation

Nous présentons maintenant l'application de l'algorithme décrit précédemment à des données simulées. Nous étudions tout d'abord le comportement de l'algorithme d'agrégation en fonction de différents facteurs, comme le bruit d'agrégation ou la difficulté du problème de classification. Nous vérifierons que l'algorithme est capable d'identifier les groupes de variables redondantes. Dans un deuxième temps, nous établissons que la conservation de la redondance via l'agrégation peut servir l'objectif d'amélioration des performances de classification, au même titre que la sélection de variables. Nous comparons en particulier les taux d'erreur obtenus après sélection avec les taux d'erreur obtenus après agrégation-sélection.

5.3.1 Premier plan de simulation

Nous commençons par considérer un plan de simulation très simple, avec pour objectif de décrire le comportement de l'algorithme d'agrégation supervisée pour les kNN.

Dans ce premier plan, nous simulons des variables de trois types différents : très informatives (concernant le statut), peu informatives et non informatives. On commence par simuler un statut (0 ou 1) pour N_{ind} individus. On crée ensuite trois variables de référence, appelées dans la suite prototypes. Le premier prototype est simulé selon une loi normale centrée de variance 1 et représente le prototype non informatif. Le second est simulé suivant une loi normale $\mathcal{N}(0,1)$ pour les observations de label 0, et suivant une loi normale $\mathcal{N}(\mu,1)$ pour les observations de label 1. C'est le prototype informatif. La difficulté de la tâche de classification

est donc quantifiée par le paramètre μ. Enfin, un troisième prototype est créé en prenant la moyenne des deux premiers, pour obtenir un prototype faiblement informatif.

Chacun des trois prototypes créés est copié en dix exemplaires bruités afin de créer des groupes de variables redondantes. Ces versions bruitées sont obtenues en prenant chaque prototype, et en lui ajoutant un bruit gaussien centré de variance σ_B^2. La difficulté de la tâche de regroupement est donc quantifiée par le paramètre σ_B^2. On obtient finalement un échantillon de N_{ind} individus, décrits par 31 variables (10 de chaque type et le label). Notons que les prototypes ne sont pas inclus dans l'échantillon. Les caractéristiques des trois types de variables sont résumées dans les tableaux 5.1 et 5.2.

	$X\|Y=0$	$X\|Y=1$	X	Ttest
Inf	$\mathcal{N}(0,1+\sigma_B^2)$	$\mathcal{N}(\mu,1+\sigma_B^2)$	$\mathcal{N}(\frac{\mu}{2},\frac{\mu^2}{4}+1+\sigma_B^2)$	$\sqrt{n}\frac{\mu}{\sqrt{1+\sigma_B^2}}$
Peu Inf	$\mathcal{N}(0,1+\sigma_B^2)$	$\mathcal{N}(\frac{\mu}{2},1+\sigma_B^2)$	$\mathcal{N}(\frac{\mu}{4},\frac{\mu^2/4+2}{4}+\sigma_B^2)$	$\sqrt{n}\frac{\mu}{2\sqrt{1+\sigma_B^2}}$
Non Inf	$\mathcal{N}(0,1+\sigma_B^2)$	$\mathcal{N}(0,1+\sigma_B^2)$	$\mathcal{N}(0,1+\sigma_B^2)$	0

TAB. 5.1 – *Distributions conditionnelles et non conditionnelles des trois types de variables simulées. La dernière colonne donne la valeur théorique de la statistique de test de Student pour une comparaison de moyenne. Cette dernière valeur sert à quantifier le pouvoir discriminant de chaque type de variable.*

	Inf	Peu Inf	Non Inf
Inf	$\frac{\mu^2+4}{\mu^2+4+4\sigma_B^2}$	$\frac{\mu^2+4}{\sqrt{\mu^2+4+4\sigma_B^2}\sqrt{\mu^2+8+16\sigma_B^2}}$	0
Peu Inf		$\frac{\mu^2+8}{\mu^2+8+16\sigma_B^2}$	$\frac{2}{\sqrt{1+\sigma_B^2}\sqrt{\mu^2+8+16\sigma_B^2}}$
Non Inf			$\frac{1}{1+\sigma_B^2}$

TAB. 5.2 – *Table des corrélations théoriques entre les trois types de variables.*

Cent échantillons sont simulés de cette manière. Chaque échantillon i est successivement utilisé comme échantillon d'entraînement. Dans un premier temps les variables de cet échantillon sont agrégées en utilisant l'algorithme CAH décrit précédemment jusqu'à ce qu'il n'y ait plus que N_C variables agrégées. Puis, on sélectionne N_S variables agrégées à l'aide d'une méthode séquentielle forward de sélection de variables. Les variables sont sélectionnées au fur et à mesure par minimisation du critère swapping. On se base sur le critère swapping plutôt que sur le risque empirique pour choisir les variables afin de rendre la sélection plus robuste : entre deux variables ayant le même taux d'erreur empirique, le critère sélectionnera la variable minimisant le nombre de points limites, i.e. le nombre de points pour lesquels le classement semble incertain (cf. partie Swapping). Enfin, la règle de classification kNN construite à partir des N_S variables agrégées sélectionnées est utilisée pour classer les observations de l'échantillon $i+1$. Ce dernier sert donc d'échantillon test pour estimer le taux d'erreur final du classificateur construit.

Dans cette étude, le nombre de copies (10) par prototype est un paramètre fixé. Les paramètres d'intérêt sont les suivants :

- la moyenne des variables informatives chez les individus de label 1 : $\mu = [2,4,6]$,
- le bruit d'agrégation : $\sigma_B^2 = [1,2,5]$,
- le nombre de variables après agrégation : $N_C = 3$, 6, ..., 27 et 30,
- le nombre de variables après sélection $N_S = [1,2,3,5,10,15,20]$,

– le nombre de voisins k et le nombre d'individus par échantillon N_{ind}.

Notons que k et N_{ind} ne sont pas ici des paramètres d'intérêt à proprement parler. On désire toutefois vérifier que les conclusions de l'étude sont stables pour différentes valeurs de ces paramètres. On travaillera avec un nombre d'individus compris entre 50 et 200, ce qui correspond à l'ordre de grandeur de la grande majorité des expériences de biopuces. On considérera les valeurs 3, 5 et 7 pour k, valeurs par défaut employées dans plusieurs études.

5.3.2 Les résultats

Classement des variables

On s'intéresse tout d'abord à l'étape d'agrégation. Afin de déterminer si l'algorithme d'agrégation classe les variables suivant leur appartenance, on considère la partition en trois groupes obtenue à partir des données, et on compte le nombre moyen de variables correctement classées sur les 100 simulations. Pour les différentes valeurs de μ et de σ_B^2, on obtient la table 5.3.

	σ_B^2		
μ	1	2	5
2	29.7	23.8	16.4
4	29.5	25.8	18.4
6	27.1	26.6	20.9

TAB. 5.3 – *Nombre moyen de variables correctement classées, en fonction des paramètres μ et σ_B^2.*

La première colonne semble contradictoire au premier abord : lorsque le bruit d'agrégation est très faible, le nombre de variables bien classées diminue lorsque μ augmente. Ce comportement s'explique par les corrélations induites entre les trois types de variables par les valeurs des paramètres. Les expressions des corrélations sont données dans la table 5.2. En prenant comme valeurs des paramètres $\mu = 6$ et $\sigma_B^2 = 1$ par exemple, on obtient les valeurs relatées en table 5.4.

D'après la table 5.4, il est possible de décrire le comportement théorique de l'algorithme. Durant les 9 premières étapes, les variables informatives sont agrégées les unes avec les autres. Durant les 9 étapes suivantes, les variables peu informatives sont agrégées ensemble. Enfin, à l'étape 19, les groupes informatif et peu informatif sont agrégés, car la corrélation entre les variables de ces deux groupes est plus forte qu'entre deux variables du groupe non informatif. Enfin, les variables non informatives sont agrégées. Ainsi, lorsque l'on sélectionne 3 groupes, on dispose théoriquement d'un groupe de 20 variables (informatives et peu informatives), et

	Inf	Peu Inf	Non Inf
Inf	0.90	0.73	0
Peu Inf		0.77	0.18
Non Inf			0.5

TAB. 5.4 – *Table des corrélations entre les trois types de variables, obtenues pour les valeurs $\mu = 6$ et $\sigma_B^2 = 1$.*

de deux groupes distincts de variables non informatives. On ne retrouve pas la répartition intuitive suivant les trois groupes initiaux.

En pratique, le phénomène décrit ici est effectivement systématiquement observé. Ceci montre que même lorsque le problème de classification est facile, les variables discriminantes facilement identifiables et le vrai nombre de groupes de variables N_C connu, l'agrégation optimale n'est pas obtenue pour la valeur N_C.

Ce point étant précisé, les résultats tirés de la table 5.3 sont par ailleurs raisonnables. Lorsque le problème d'agrégation est facile, c'est-à-dire lorsque le bruit d'agrégation est faible ou que les variables informatives ont un profil très différent des variables non informatives (μ élevé), l'algorithme retrouve les groupes d'appartenance. Les performances de classement se dégradent lorsque le bruit augmente ou lorsque les profil des variables informatives et non informatives sont proches. Ainsi, pour des problèmes d'agrégation faciles il ne devrait pas y avoir de dégradation des performances dues au regroupement de variables ne faisant pas partie du même groupe.

Dans le cas de simulations, il est aisé de déterminer si le problème d'agrégation est complexe ou non. Lorsque l'on dispose de données réelles, une telle information n'est pas accessible. Les courbes d'agrégation ou le critère de Gini peuvent alors servir d'outils de diagnostic pour déterminer si le problème de classification est difficile. La figure 5.3 présente deux courbes d'agrégation cumulées, correspondant chacune à une simulation pour $\mu = 6$ et $\sigma_B^2 = 1$ et $\mu = 2$ et $\sigma_B^2 = 5$ respectivement :

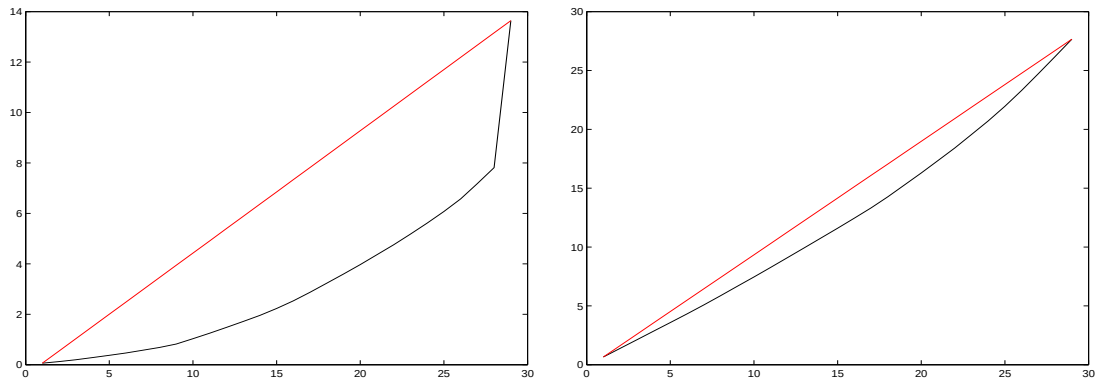


FIG. 5.3 – *Deux courbes d'agrégation cumulées. A gauche, le problème d'agrégation est facile ($\mu = 6$ et $\sigma_B^2 = 1$), et le point de rupture apparaît lorsque l'on agrège des variables de groupes différents. A droite, le problème d'agrégation est difficile ($\mu = 2$ et $\sigma_B^2 = 5$) et il n'y a pas de rupture claire.*

On observe clairement une rupture dans la courbe pour le problème facile, correspondant à l'étape où l'on commence à agréger des variables qui ne font pas partie du même groupe. Lorsque le bruit d'agrégation est élevé (cas difficile), la rupture n'est plus observée. Nous présentons ici la table 5.5 des indices de Gini, moyennés sur 100 simulations, pour différentes valeurs de μ et de σ_B^2 :

On remarque que l'indice se dégrade particulièrement lorsque le bruit d'agrégation

	σ_B^2		
μ	1	2	5
2	0.35	0.18	0.13
4	0.45	0.22	0.13
6	0.53	0.29	0.13

TAB. 5.5 – *Indices de Gini moyens, pour différentes valeurs de μ et de σ_B^2*

augmente.

Les résultats d'agrégation obtenus dans cette première étude sont donc satisfaisants. Lorsque les données ne sont pas trop bruités, l'algorithme retrouve les groupes et facilite ainsi l'interprétation des données. Le critère de Gini et les courbes d'agrégation apportent une première information sur les données, nous permettant de distinguer les cas où l'agrégation est susceptible d'améliorer les résultats.

Influence des différents paramètres sur les performances

On s'intéresse maintenant à l'impact des différents paramètres du problème sur les performances de classification. Pour cela on réalise une analyse de la variance, où la variable à expliquer est le taux d'erreur réel du classificateur obtenu pour chaque simulation, régressée sur les valeurs des différents paramètres du modèle μ, σ_B^2, N_C, N_S, k et N_{ind}, ainsi que sur leurs interactions (les variables explicatives sont toutes considérées comme qualitatives). Le résultat de l'analyse de la variance est résumé figure 5.4. Pour chaque paramètre, les carrés moyens (colonne MS) mesurent la contribution de ce paramètre à la variance de la variable d'intérêt.

On remarque tout d'abord que le niveau d'agrégation N_C, ainsi que l'interaction entre le niveau d'agrégation et le nombre de variables sélectionnées $N_C \times N_S$ font partie des 11 facteurs les plus importants de la liste présentée. L'étape d'agrégation a donc un impact sur la qualité des résultats, mais aussi sur les performances de l'étape de sélection qui lui succède. Par ailleurs, nous constatons que les interactions faisant intervenir le nombre d'observations N_{ind} et l'une des variables d'intérêt N_C ou N_S n'apparaissent qu'en fin de tableau (à partir de la $23^{\text{ème}}$ place), et sont associées à des carrés moyens plus faibles. On en conclut que les conclusions tirées sur l'agrégation ou sur la sélection ne dépendent pas du nombre d'individus dans l'échantillon. Nous présenterons donc les résultats pour $N_{ind} = 100$, les résultats obtenus pour d'autres valeurs de N_{ind} étant similaires. Un raisonnement similaire montre que le nombre de voisins k choisi pour l'algorithme des kNN ne change pas les conclusions sur l'agrégation et la sélection. Nous fixons donc aussi la valeur de k à 3 dans la présentation des résultats et dans les simulations suivantes.

Comparaison Agrégation / Agrégation-Sélection

On s'intéresse aux performances de classification obtenues avec la stratégie d'agrégation/sélection, comparées à celles obtenues avec la stratégie de sélection. La figure 5.5 présente les courbes de performance obtenues suivant les valeurs des paramètres μ et σ_B^2. Sur chaque graphique, une courbe correspond à une valeur fixée de N_S. Par exemple, la courbe violette montre l'évolution du taux de performance de la règle de classification construite

Obs	Source	DF	MS
1	DiffClass	2	2194.034908
2	BruitComp	2	926.490198
3	BruitComp*DiffClass	4	35.669314
4	Nind	2	13.352009
5	Nsel	6	12.442147
6	Ncomp	9	10.060104
7	BruitComp*Nsel	12	7.125284
8	Nvois	2	6.107672
9	Nind*BruitComp	4	2.049204
10	BruitComp*Ncomp	18	1.881973
11	Nsel*Ncomp	40	1.134335
12	Nind*BruitCo*DiffCla	8	0.695532
13	BruitCo*DiffCla*Nsel	24	0.638543
14	BruitC*DiffCla*Ncomp	36	0.626955
15	Nvois*DiffClass	4	0.611855
16	Nvois*BruitComp	4	0.410664
17	Nvois*BruitC*DiffCla	8	0.275900
18	Nind*DiffClass	4	0.213487
19	DiffClass*Ncomp	18	0.129872
20	DiffClass*Nsel	12	0.096352
21	BruitComp*Nsel*Ncomp	80	0.081840
22	Nind*Nvois*DiffClass	8	0.067999
23	Nind*Nsel	12	0.054892
24	Nind*Ncomp	18	0.035859
25	DiffClass*Nsel*Ncomp	80	0.034160
26	Nind*BruitComp*Ncomp	36	0.031615
27	Nind*Nvois	4	0.027713
28	Nind*Nvois*BruitComp	8	0.015260
29	Nind*BruitComp*Nsel	24	0.012745
30	Nind*Nsel*Ncomp	80	0.012533
31	Nvois*Ncomp	18	0.009023
32	Nind*DiffClass*Ncomp	36	0.009023
33	Nvois*Nsel	12	0.008978
34	Nvois*DiffClass*Nsel	24	0.005041
35	Nind*DiffClass*Nsel	24	0.004822
36	Nind*Nvois*Ncomp	36	0.002783

FIG. 5.4 – *Table de l'analyse de la variance pour l'analyse du risque en fonction des paramètres μ, σ_B^2, N_C, N_S, k et N_{ind}. Les interactions sont considérées jusqu'à l'ordre 3. Les interactions non significatives au seuil de 5% ne sont pas représentées.*

lorsqu'une seule variable est sélectionnée. Le taux d'agrégation augmente de droite à gauche : l'extrémité droite de la courbe violette représente le taux d'erreur réel de la règle construite avec une variable choisie parmi 30 (pas d'agrégation), et l'extrémité gauche le taux d'erreur pour une règle construite avec une variable choisie parmi 3, issues de l'agrégation maximale en 3 groupes.

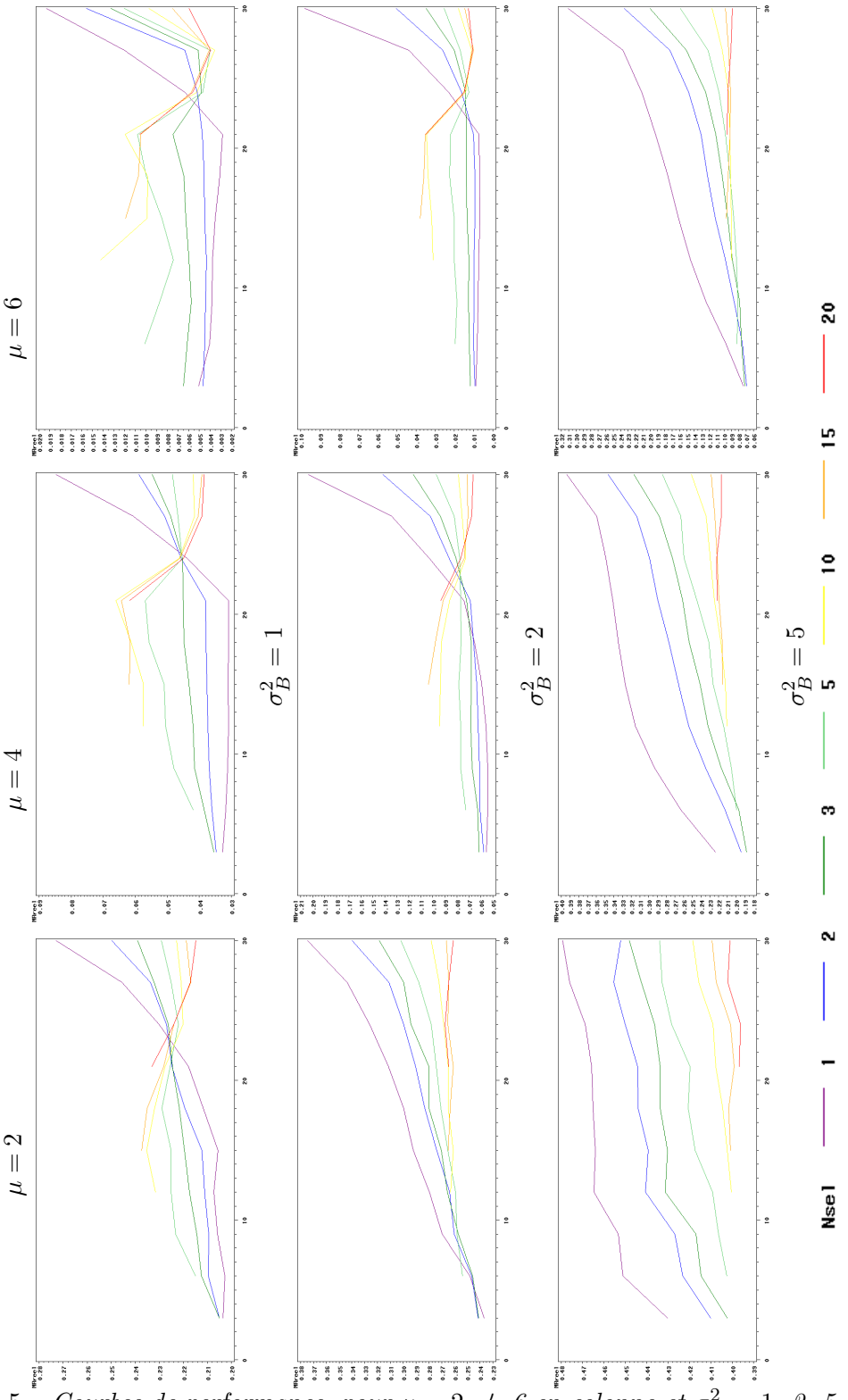
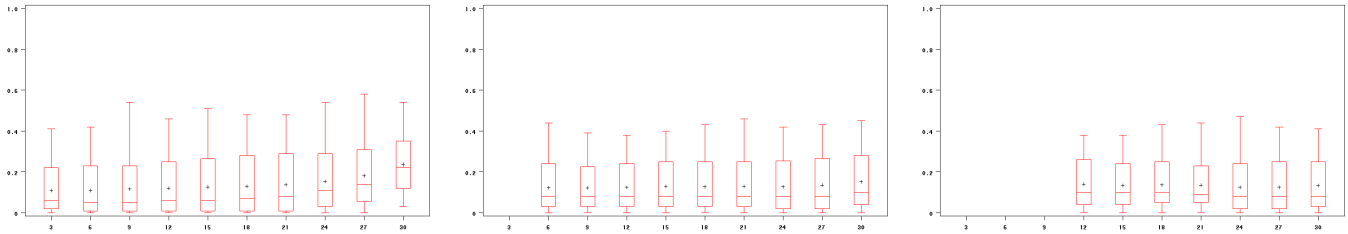


FIG. 5.5 – *Courbes de performance, pour $\mu = 2$, 4, 6 en colonne et $\sigma_B^2 = 1$, 2, 5 en ligne.*

On constate premièrement que dans toutes les configurations, excepté dans le cas le plus difficile où $\mu = 2$ et $\sigma_B^2 = 5$, le meilleur taux d'erreur enregistré est toujours obtenue pour une combinaison agrégation/sélection, et non par sélection uniquement. Ce premier résultat confirme qu'il existe une manière pertinente d'utiliser la redondance entre variables, et que la présence de la redondance n'altère pas systématiquement les performances du classificateur construit. Pour des valeurs de σ_B^2 inférieures à 2, la meilleure règle de classification est toujours construite en sélectionnant une seule variable après agrégation. L'agrégation réalise donc l'objectif souhaité : elle synthétise l'information présente dans les données, ce qui la rend plus robuste et facilite l'étape de sélection. Comme le montrent les graphiques, plus le bruit d'agrégation est élevé plus la valeur de N_C optimale observée est faible, ce qui semble raisonnable. En effet, plus l'on agrège plus le risque de se tromper dans les regroupements est élevé. Lorsque la redondance est difficile à identifier, il ne faut pas trop agréger afin de ne pas dégrader les performances. Ce point sera plus amplement illustré dans la partie suivante. Remarquons toutefois que les performances restent toujours meilleures pour les valeurs de N_C les plus élevées.

Si l'on considère les courbes obtenues lorsque l'on sélectionne un grand nombre de variables, on constate qu'elles montrent généralement une dégradation des résultats rapide lorsque l'on agrège. Cette dégradation s'explique ici par la "quantité d'information" disponible. Il n'y a ici que 10 variables informatives, et 10 variables moyennement informatives. Ainsi, dans le meilleur des cas il ne faudrait pas sélectionner plus de 20 variables. Les courbes atteignent donc un minimum lorsque la totalité de l'information pertinente est prise en compte, puis remontent par conséquence de l'ajout de variables non pertinentes, à travers l'agrégation.

La figure 5.6 montre les Box-plots pour le taux d'erreur, tracés pour les différents taux d'agrégation et pour N_S=1, 5 et 10. Il apparaît que l'agrégation n'influe généralement pas sur la variabilité des résultats : les box-plots sont à peu près tous de la même taille, avec une légère diminution au fur et à mesure de l'agrégation pour les faibles valeurs de N_S. L'agrégation ne stabilise donc que peu les résultats. Les comportements observés pour les autres valeurs de N_S sont similaires (non montrés ici).



$N_S = 1$ $N_S = 5$ $N_S = 10$

FIG. 5.6 – *Box-plots du taux d'erreur de généralisation, pour différentes valeurs de N_S. L'abscisse représente le taux d'agrégation.*

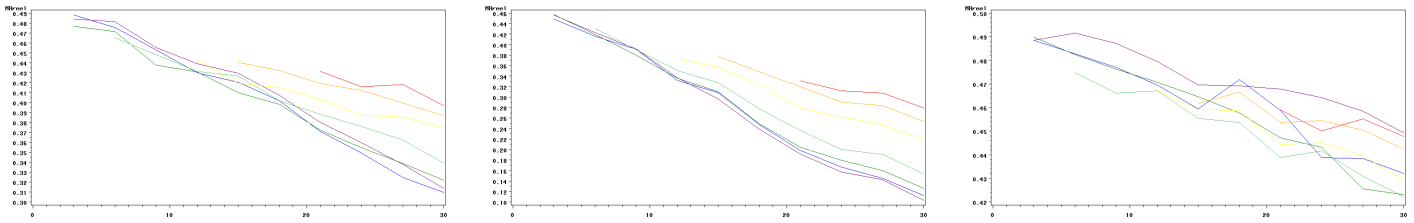
5.3.3 Autres plans de simulations

Dans cette partie, nous fixons les paramètres k et N_{ind} respectivement à 3 et 100. On cherche maintenant à savoir si l'agrégation supervisée apporte un gain en performance dans différentes conditions. Pour cela, nous considérons plusieurs modèles de simulation. Le nombre total de variables dans tous ces modèles est fixé à 30.

Pas de redondance pour les variables

On crée une seule variable informative, et 29 variables non informatives toutes créées indépendamment les unes des autres, de moyenne nulle et de variance σ_B^2. Il n'y a aucune information à agréger, le taux d'agrégation "idéal" est donc 30, et le nombre de variables à sélectionner 1.

La figure 5.7 présente les résultats obtenus pour (μ,σ_B^2)=(2,1), (4,1) et (2,2). On obtient des résultats similaires pour d'autres valeurs.



(μ,σ_B^2) = (2,1) (μ,σ_B^2) = (4,1) (μ,σ_B^2) = (2,2)

FIG. 5.7 – *Courbes de performance, pour différentes valeurs de μ et σ_B^2, en l'absence de toute redondance.*

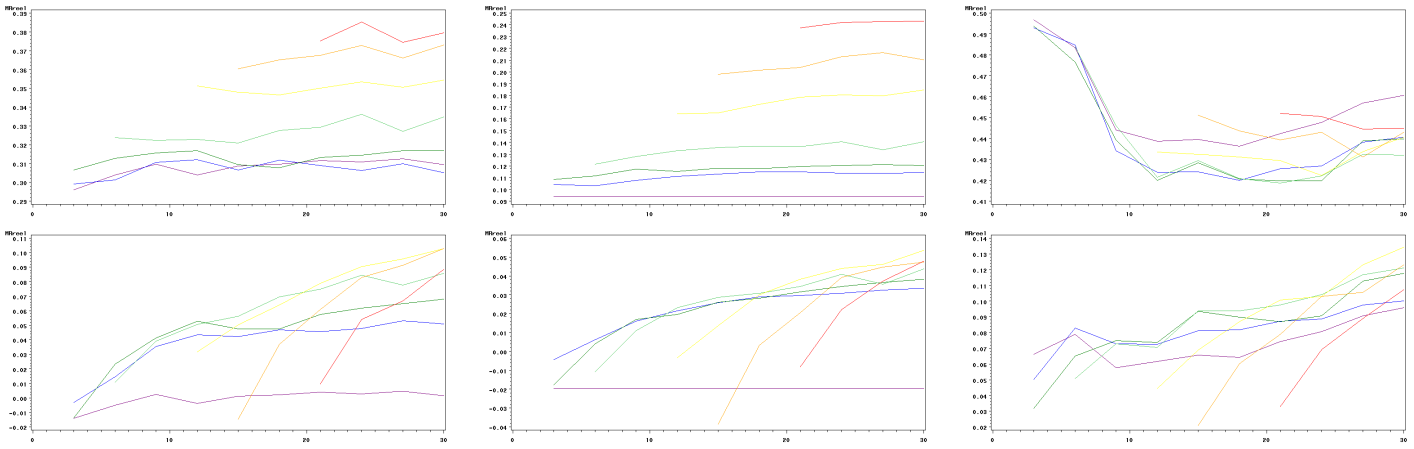
L'information est rapidement dégradée, même pour des valeurs élevées de μ. Il ne faut donc pas agréger lorsqu'il n'y a pas de redondance. Bien sûr ce résultat était attendu, mais il montre qu'il est important de disposer d'outils de diagnostic pour déterminer s'il faut agréger ou non. L'indice moyen de Gini dans le cas $\mu = 6$ et $\sigma_B^2 = 2$ est de 0.13, bien en dessous de la valeur observée pour le cas le plus défavorable de la première étude. L'indice de Gini montre bien ici que l'agrégation est soit trop difficile, soit inadaptée.

Pas de redondance pour la variable explicative

On crée une seule variable informative, et 29 copies d'un prototype non informatif. Il y a donc de la redondance pour ces variables, et l'agrégation est potentiellement utile. Idéalement, l'algorithme d'agrégation devrait classer les 29 variables non informatives dans une même classe et la variable informative dans une classe à part. Le taux d'agrégation optimal serait alors de 2 et le nombre de variables sélectionnées de 1.

Si l'on reprend les expressions de la table 5.2, les corrélations entre la variable informative et les variables non informatives sont nulles, et les corrélations entre variables non informatives sont de $1/(1 + \sigma_B^2)$. Nous fixons ici les valeurs de σ_B^2 à 0.11, 1 et 2, ce qui correspond à des corrélations de 0.9, 0.5 et 0.33. Les résultats pour les couples (μ, σ_B^2) =(2,1), (4,1) et (2,2) sont présentés en figure 5.8.

Les courbes de performance diffèrent du cas non informatif précédent lorsque le bruit d'agrégation n'est pas trop important. Lors des premières étapes d'agrégation, on ne dégrade pas les performances du classificateur. On ne les améliore pas non plus, puisque l'algorithme n'agrège en théorie que les variables non informatives. On constate par ailleurs que plus le bruit d'agrégation augmente, plus l'on prend le risque en agrégeant de dégrader l'information. Ainsi, pour les valeurs (μ, σ_B^2) = (2,2), la courbe de performance montre qu'en moyenne le taux d'erreur augmente lorsque le taux d'agrégation est supérieur à 10. Cette dégradation est due au fait que l'unique variable informative est agrégée avec des variables non informatives. Cette observation est confirmée lorsque l'on s'intéresse au nombre moyen d'étapes de l'algorithme CAH avant que la variable informative ne soit regroupée. Lorsque le bruit d'agrégation est



$(\mu, \sigma_B^2) = (2,1)$ $(\mu, \sigma_B^2) = (4,1)$ $(\mu, \sigma_B^2) = (2,2)$

FIG. 5.8 – *Courbes de performance, pour différentes valeurs de μ et σ_B^2, en l'absence de redondance pour la variable informative.*

de 0.11, la variable informative est agrégée avec l'ensemble des autres variables à la dernière étape, i.e. l'information sur le label est conservée. Lorsque le bruit de compression est de 2, la variable informative est en moyenne agrégée avec une autre variable à l'étape 20.

Le gain vient ici de l'écart entre le taux d'erreur calculé sur l'échantillon test et le taux d'erreur réel, calculé sur l'échantillon de validation. La deuxième série de courbes en figure 5.8 montre que cet écart diminue lorsque l'on agrège. Le gain en précision varie suivant les valeurs des paramètres μ, σ_B^2 et N_S entre 1 et 10%. Ici encore, les résultats sont variables suivant le niveau de bruit des données. Pour σ_B^2=5 par exemple, l'agrégation dégrade systématiquement les résultats. Ici encore le choix d'agréger ou non doit être guidé par l'indice de Gini. Ce critère varie ici entre 0.13 et 0.72. Pour les cas représentés en figure 5.8, le critère varie entre 0.13 et 0.15. Un faible indicateur de Gini n'est donc pas toujours le signe de l'inefficacité de l'agrégation.

Ainsi même lorsque la redondance entre variables ne concerne que les variables non informatives, il peut être intéressant d'agréger afin d'obtenir de meilleures estimations du taux d'erreur final.

5.4 Applications aux données réelles

5.4.1 Estimation des performances, choix des paramètres et échantillonnage

Comparaison de méthodes

Dans cette partie, nous appliquons l'algorithme d'agrégation à différents jeux de données issus de la littérature. Bien que notre objectif soit d'améliorer l'interprétation des résultats issus d'une analyse de classification supervisée, il est important de quantifier l'impact de la stratégie d'agrégation sur le taux d'erreur du classificateur construit, et de comparer les performances de ce classificateur aux performances obtenues par d'autres méthodes sur le même jeu de données. Comparer plusieurs méthodes implique le choix d'une méthode d'estimation du taux d'erreur des classificateurs construits. Différentes stratégies de rééchantillonnage peuvent

alors être envisagées :

- la validation par "Hold-out", ou estimation par échantillon de validation, qui consiste à exclure de l'échantillon initial n_v observations, qui serviront ensuite à estimer la règle construite sur l'échantillon d'entraînement constitué des $n-n_v$ données. Cette procédure peut être réalisée q fois, en changeant à chaque fois d'échantillon de validation,
- la validation croisée "K-fold", qui consiste à diviser l'échantillon initial en K groupes d'observations d'effectifs égaux, chaque groupe servant à tour de rôle d'échantillon de validation,
- la validation croisée "Leave-One-Out" (LOO), où chaque individu sert à tour de rôle d'échantillon de validation, les $n-1$ observations restantes servant d'échantillon test.

Toutes ces procédures, couramment utilisées en pratique, sont éligibles pour comparer les règles de classification. Toutefois, il est important de noter que les résultats obtenus avec chacune d'entre elles peuvent être très différents, en particulier lorsque les données sont de grande dimension. Nous avons déjà vu sur un exemple (donnée Colon) au chapitre 3 que les taux d'erreur obtenus par validation croisée et par Hold-out peuvent varier de 9%. De ce fait, nous avons choisi de systématiquement reprendre la méthode de rééchantillonnage proposée par les auteurs de l'article de référence, afin de pouvoir comparer rigoureusement les résultats. Le plan de rééchantillonnage sera donc spécifique à chaque jeu de données présenté.

Choix des paramètres

La partie précédente nous a permis d'étudier le comportement de l'algorithme d'agrégation dans différentes situations. Cette étude a en particulier démontré que le choix des paramètres N_C et N_S est critique : une mauvaise spécification du taux d'agrégation ou de sélection est susceptible de dégrader considérablement les performances du classificateur construit, mais aussi l'interprétation des résultats obtenus. Il est donc nécessaire de proposer une méthode rigoureuse pour le choix de ces deux paramètres. Ce choix sera lui aussi basé sur une procédure de rééchantillonnage. Là encore, lorsque le problème du choix des paramètres est traité dans l'article où sont initialement présentées les données, nous reprenons le plan d'échantillonnage proposé.

Dans la suite, les plans de rééchantillonnage seront désignés par le triplet $[n_e,n_t,n_v]$, où

- n_e représente la taille de l'échantillon d'entraînement, qui sera utilisé pour construire les différentes règles de classification,
- n_t représente la taille de l'échantillon test, utilisé pour déterminer les paramètres N_C et N_S optimaux pour la règle de classification,
- n_v représente la taille de l'échantillon de validation, servant à estimer le taux d'erreur de la règle finalement sélectionnée. C'est ce dernier taux d'erreur qui sera utilisé pour comparer la procédure d'agrégation-sélection proposée ici aux méthodes concurrentes.

A propos du paramètre N_C

Le choix conjoint des paramètres N_C et N_S ne devrait pas être uniquement basé sur la minimisation du taux d'erreur de la règle de classification. Améliorer le taux d'erreur en identifiant les variables (ou les groupes de variables) non informatives pour le label est bien l'objectif de la sélection de variables, en revanche ce n'est pas celui de la procédure d'agrégation. Il est donc souhaitable que le choix du paramètre N_C ne soit pas uniquement basé sur la minimisation du taux d'erreur de la règle, mais prenne en compte l'objectif d'interprétation. Au delà d'un certain nombre d'étapes, l'algorithme d'agrégation regroupe des variables qui

ne sont pas redondantes, et dégrade l'information contenue dans les données. On doit donc fixer une limite à la procédure d'agrégation. Cette limite peut être trouvée en considérant la courbe d'agrégation non cumulé. On observe que la perte est stable sur les premières étapes,

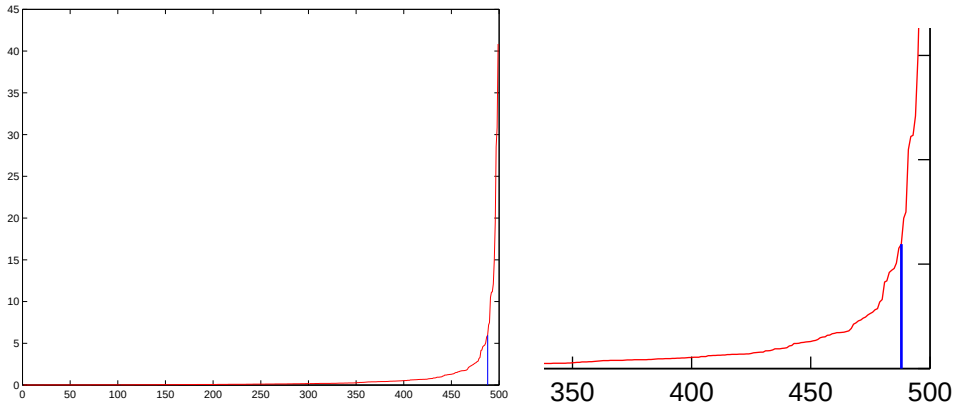


FIG. 5.9 – *Détermination du critère d'arrêt pour la procédure d'agrégation. La figure de droite montre le comportement localement instable de la courbe d'agrégation, justifiant de travailler sur l'enveloppe convexe de la courbe.*

puis se dégrade rapidement : on agrège alors des groupes de variables qui ne sont plus homogènes. L'objectif est d'identifier le taux d'agrégation N_C^* à partir duquel la perte d'inertie enregistrée devient trop forte. Le choix du couple (N_C, N_S) se fera alors par minimisation du taux d'erreur, pour les valeurs de N_C supérieures ou égales à N_C^*. Dans l'exemple de la figure 5.9, il y a 500 variables initiales. Les étapes supérieures à l'étape $t = 485$ correspondent à des agrégations en moins de $p - t + 1 = 16$ groupes. La perte mesurée pour ces dernières agrégation est très forte, on agrège des groupes non homogènes. On ne considérera donc que les agrégations en $N_C^* = 16$ groupes ou plus.

Plusieurs méthodes peuvent être envisagées pour détecter la valeur de décrochage N_C^*. Lavielle (1999) propose un critère d'arrêt basé sur le calcul des dérivées secondes. Pour chaque étape Q, la différence de pente

$$Z_{Q+1} + Z_{Q-1} - 2 \times Z_Q$$

est calculée, où Z_Q est la valeur du critère à l'étape Q. On arrête l'agrégation lorsque le critère est maximum, où lorsqu'il dépasse un seuil fixé par l'utilisateur. Une alternative possible est de calculer à chaque étape l'angle entre les dérivées premières aux points Q et $Q + 1$, et d'arrêter l'agrégation lorsque cet angle est maximal.

La difficulté majeure pour l'application de ces critères réside dans la forme localement irrégulière de la courbe d'agrégation. La figure 5.9 (à droite) illustre ce problème d'irrégularité. Afin d'éviter les risques d'identification de maximum locaux du critère, Lavielle (1999) recommande de travailler sur l'enveloppe convexe de la courbe d'agrégation, plutôt que sur la courbe directement. Nous avons observé qu'en pratique travailler sur l'enveloppe convexe permet effectivement une stabilisation des résultats. En particulier, les différents critères d'arrêt envisagés donnent alors des résultats très similaires. Dans la suite, le critère d'arrêt utilisé sera la rupture de pente, calculé sur l'enveloppe convexe de la courbe d'agrégation.

Nous présenterons donc pour chaque jeu de données les résultats obtenus

– par classification kNN brute (sans sélection ni agrégation),

– par classification kNN avec sélection,
– par classification kNN avec agrégation et sélection.

5.4.2 Reconnaissance vocale

Présentation des données

Nous considérons ici un premier jeu de données réelles, initialement proposé par Biau *et al.* (2005). Ce jeu de données est composé de 55 enregistrements du mot "boat" et de 45 enregistrements du mot "goat". Pour chaque enregistrement le signal analogique est discrétisé en une série de 8192 points. Chaque signal ainsi obtenu est ensuite décomposé en série de Fourier sur une base de 500 périodes :

$$f(t) = \sum_{j=-500}^{500} A_r e^{i\frac{2\pi j}{p}t} \quad .$$

Les données sont donc de la forme (X,Y), où $X^j = |A_j|$ est l'énergie associée à la période angulaire $\frac{2\pi j}{p}$, et $Y = 1$ si le mot est Boat, 0 sinon. L'objectif est de discriminer les deux types de mots en utilisant l'algorithme kNN, en se basant sur les énergies de la décomposition de Fourier.

Nous présentons tout d'abord une rapide analyse descriptive des données GOAT-BOAT. La figure 5.10 montre la densité spectrale moyenne obtenue pour l'ensemble des données. En abscisse sont représentées les 500 périodes angulaires (notées ici de 1 à 500), en ordonnée la valeur moyenne de l'énergie associé à chaque période. Le code couleur est fonction de la statistique du test de Student de comparaison de deux populations, les couleurs chaudes représentant une statistique de test élevée. Cette représentation ne nous renseigne que sur le pouvoir discriminant des variables (énergies) prises une par une, sans considérer les possibles interactions entre variables. Nous pouvons toutefois en retirer qu'il existe clairement deux gammes de période, représentées ici en rouge, de pouvoir discriminant très élevé. Dans la première gamme (16-37) la valeur moyenne de la statistique de Student est de 10.4, et de 9.1 dans la seconde gamme (91-95). Le problème de classification devrait donc être simple.

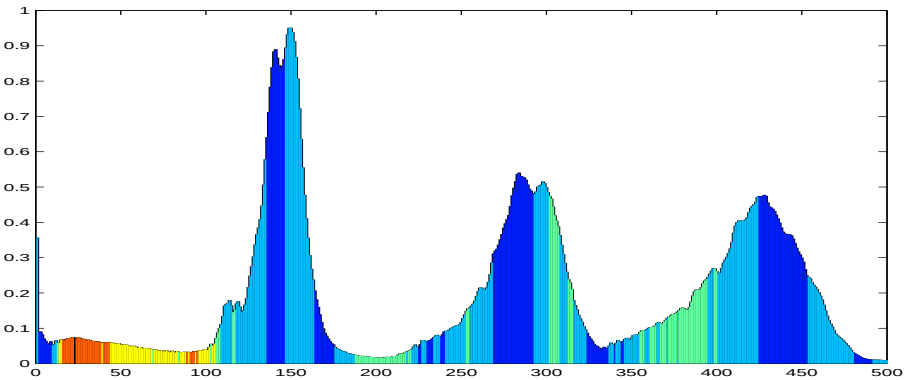


FIG. 5.10 – *Spectre moyen du jeu de données Goat / Boat. La couleur est fonction de la statistique du test de Student associée à chaque coefficient. On remarque deux zones discriminantes : 16-43 et 89-95.*

Ce jeu de données sur la reconnaissance vocale a déjà été employé par différents auteurs. Biau *et al.* (2005) réalisent la classification par kNN en se basant sur les d premiers coefficients

de la série de Fourier. Les auteurs sélectionnent le nombre de voisins k et le nombre de coefficients d par critère pénalisé, et relatent un taux d'erreur de 21% sur ces données. Tuleau (2005) montre que les résultats obtenus par Biau peuvent être améliorés en modifiant le terme de pénalité employé dans la procédure de sélection des paramètres, et obtient un taux d'erreur de l'ordre de 15%. Enfin, dans Rossi et Villa (2006), les auteurs utilisent une adaptation de l'algorithme des SVM pour données fonctionnelles, et obtiennent un taux d'erreur de 8% en utilisant un noyau gaussien. Remarquons que dans ces différentes utilisations du jeu de données, la représentation des données choisie ne correspond pas à la représentation en série de Fourier choisie ici (voir Biau *et al.* (2005)), et que le changement de représentation améliore considérablement les résultats de classification. C'est pourquoi le taux d'erreur de référence sera le taux d'erreur obtenu par classification kNN brute sur les 500 énergies.

Néanmoins, afin de pouvoir comparer nos résultats avec ceux de Biau *et al.* (2005) et Rossi et Villa (2006), nous estimerons les taux d'erreur des différentes règles calculées en réalisant 100 tirages du plan d'échantillonnage [50,49,1] utilisé par ces auteurs.

Résultats

Les données spectrales sont des données très structurées, pour lesquelles on s'attend à ce qu'une même information soit portée pas une plage de fréquence plutôt que par quelques fréquences isolées. Ce comportement est effectivement illustré par la figure 5.10, qui montre clairement qu'il existe deux gammes de fréquences informatives. On s'attend donc à ce que l'algorithme d'agrégation retrouve cette structure en plages de fréquence. La figure 5.11 montre ici le résultat de la procédure d'agrégation puis de sélection sur un échantillonnage. Le comportement observé correspond bien au comportement attendu.

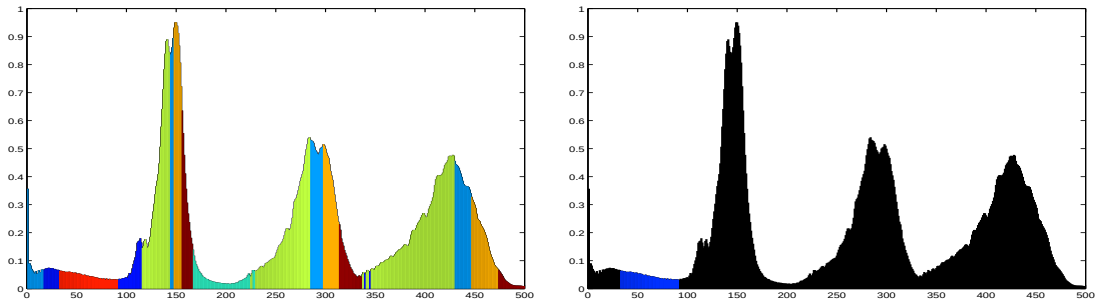


FIG. 5.11 – *Représentation des données après agrégation (gauche), puis après sélection (droite). L'agrégation permet d'obtenir une représentation en gammes de périodes. La sélection choisit ensuite la gamme la plus pertinente correspondant à la gamme identifiée dans l'étude descriptive.*

Afin de vérifier si la procédure d'agrégation permet effectivement de rendre les résultats de sélection de variables plus robustes, nous nous intéressons maintenant à la fréquence avec laquelle les différentes périodes angulaires ont été sélectionnées. La figure 5.12 donne une représentation synthétique de ces fréquences de sélection. On remarque ici que la procédure d'agrégation, alliée à la procédure de sélection, permet une détection claire des gammes de périodes informatives, cette détection étant moins évidente lorsque seule la sélection est employée. Sur 100 échantillonnages, les périodes de la gamme 16-37 sont sélectionnées entre 89 et 95 fois par la procédure d'agrégation / sélection. Ces mêmes périodes sont sélectionnées moins de 17 fois par la procédure de sélection, mises à part pour les périodes 28, 29, 30 et

Méthode	N_C	N_S	Tx Erreur
kNN brut	-	-	5%
Sélection	-	6.7	7%
Agrég.-Sélect.	19.5	5.2	0%

TAB. 5.6 – *Nombre de groupes après agrégation, nombre de variables sélectionnées et taux d'erreur des différentes règles (moyenne sur 100 rééchantillonnages)*

31, sélectionnées respectivement 34, 23, 49 et 84 fois. Les périodes de la deuxième gamme informative (91-95) ne sont jamais sélectionnées par la procédure de sélection (entre 0 et 2 fois), et 90 fois par la procédure d'agrégation / sélection.

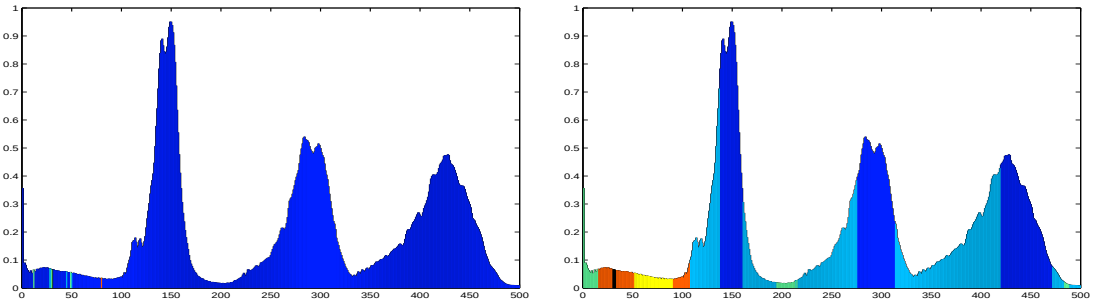


FIG. 5.12 – *Spectres moyens du jeu de données Goat / Boat. La couleur est fonction de la fréquence avec laquelle la longueur d'onde a été choisie respectivement par la procédure de sélection simple à gauche, et par la procédure d'agrégation / sélection à droite.*

Nous nous intéressons maintenant au taux d'erreur des différentes règles de classification construites. Les performances de différents classificateurs sont résumées dans le tableau 5.6. On remarque que le taux d'erreur obtenu avec la sélection de variables est plus élevé que celui obtenu avec les kNN bruts. Ce résultat surprenant peut venir de la grande variabilité de la procédure LOO. Si l'on considère l'agrégation, on constate que le taux d'agrégation moyen est de 19.5 : la taille des données est donc réduite d'un facteur 25 après la procédure d'agrégation. On obtient un taux d'erreur par LOO de 0%, ce qui confirme d'une part que l'information pertinente n'a pas été dégradée par la procédure d'agrégation, et d'autre part que la forte réduction de dimension a facilité l'étape de sélection.

5.4.3 Données de biopuces

Présentation des données

Nous considérons maintenant les données de biopuces issues de Golub *et al.* (1999). L'objectif de cette étude est de distinguer les formes de leucémie AML et ALL sur la base des expressions de gènes mesurées dans des cellules prélevées dans la moelle osseuse du patient. L'étude porte sur 78 individus pour lesquels l'expression de 3571 gènes (après normalisation et filtrage) est mesurée par la technique des puces à ADN. Les données sont donc de la forme (X,Y), où X^j est la mesure d'expression du gène j, et $Y = 1$ si la leucémie est de type AML, 0 sinon.

Les utilisations de ce jeu de données dans la littérature sont très nombreuses, et suivant les auteurs différents plans de rééchantillonnages furent proposés. Nous choisissons ici d'estimer

Méthode de classification	**Taux d'erreur LOO**
Adaboost (Ben-Dor *et al.* (2000))	4.2%
SVM (Noyau quadratique, Ben-Dor *et al.* (2000))	4.2%
Regression logistique (Krishnapuram *et al.* (2004b))	2.8%
Régression probit (Krishnapuram *et al.* (2004b))	2.8%
JCFO (noyau linéaire) (Krishnapuram *et al.* (2004b))	0%

TAB. 5.7 – *Taux d'erreur de différentes méthodes de classification appliquées au jeu de données Golub.*

le taux d'erreur final par LOO afin de comparer nos résultats à ceux de Krishnapuram *et al.* (2004b), ainsi qu'à l'ensemble des résultats présentés dans cet article. Par ailleurs, la sélection des paramètres se fera ici par Hold-out sur un échantillon test de 26 personnes. Le plan d'échantillonnage est donc [45,26,1]. Les résultats obtenus dans d'autres études sont présentés dans la table 5.7.

Il apparaît que les taux d'erreur obtenus par les différentes méthodes sont tous proches de 0%, il n'y a donc rien à gagner du point de vue des performances de classement. Par ailleurs, la stabilité de la liste des gènes sélectionnés n'est pas étudiée dans les articles cités. Nous ne pourrons donc pas comparer de ce point de vue les méthodes présentées dans ces articles à la notre.

Résultats

Les résultats obtenus pour ce jeu de données sont résumés dans la table 5.8.

Méthode	N_C	N_S	Tx Erreur
kNN brut	-	-	2.8%
Sélection	-	8.6	7%
Agrég.-Sélect.	22.5	3.8	2.8%

TAB. 5.8 – *Nombre de groupes après agrégation, nombre de variables sélectionnées et taux d'erreur des différentes règles pour le jeu de données Golub (moyenne sur 100 rééchantillonnages)*

La première constatation est que la règle de classification kNN brut donne des résultats très satisfaisants. En comparaison, la sélection de variables dégrade les performances de 4%. Il est difficile d'expliquer une telle perte de performance. Elle peut être due à l'instabilité de la procédure de sélection de variables. La procédure d'agrégation-sélection donne des performances identiques aux kNN bruts, mais ne base son classement que sur une partie des variables.

Afin de déterminer la pertinence de la liste des gènes identifiés par les différentes procédures, nous nous référons au travail de Su *et al.* (2003). Dans cet article, les auteurs proposent une liste de 100 gènes déterminés comme importants pour l'objectif de discrimination par différentes méthodes. Cette liste, que nous désignerons dans la suite comme la liste Su2003, nous servira de référence pour la comparaison entre nos résultats et les résultats d'analyses antérieures du même jeu de données.

Nom	Fonction
D10495_at	PRKCD Protein kinase C, delta
HT1612_at	Macmarcks
J05243_at	SPTAN1 Spectrin, alpha, non-erythrocytic 1 (alpha-fodrin)
L47738_at	Inducible protein mRNA
M11722_at	Terminal transferase mRNA
M22960_at	PPGB Protective protein for beta-galactosidase (galactosialidosis)
M29696_at	IL7R Interleukin 7 receptor
M55150_at	FAH Fumarylacetoacetate
M63138_at	CTSD Cathepsin D (lysosomal aspartyl protease)
M63379_at	CLU Clusterin (compl. lysis inhibitor; testosterone-repressed prostate message 2; apolipoprotein J)
M63959_at	LRPAP1 Low density lipoprotein-related protein-ass. protein 1 (alpha-2-macroglobulin receptor-ass. protein 1
M84526_at	DF D component of complement (adipsin)
M92357_at	B94 PROTEIN
U46499_at	GLUTATHIONE S-TRANSFERASE, MICROSOMAL
X07743_at	PLECKSTRIN
X51521_at	VIL2 Villin 2 (ezrin)
X52056_at	SPI1 Spleen focus forming virus (SFFV) proviral integr. oncogene spi1
X59417_at	PROTEASOME IOTA CHAIN
X61587_at	ARHG Ras homolog gene family, member G (rho G)
X95735_at	Zyxin
Y08612_at	RABAPTIN-5 protein
D26156_s_at	Transcriptional activator hSNF2b
U05572_s_at	MANB Mannosidase alpha-B (lysosomal)
M31211_s_at	MYL1 Myosin light chain (alkali)
M98399_s_at	CD36 CD36 antigen (collagen type I receptor, thrombospondin receptor)
M31523_at	TCF3 Transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)
U29175_at	Transcriptional activator hSNF2b

TAB. 5.9 – *Liste des gènes identifiés par la stratégie d'agrégation-sélection et recensés dans la littérature.*

Avec la procédure d'agrégation-sélection, nous avons identifié une liste de 40 gènes sélectionnées plus de 60 fois sur les 100 échantillonnages. La plupart de ces gènes font partie de la liste Su2003, leur nom et leurs fonctions sont reportés dans la table 5.9. Si l'on relance la procédure de validation croisée sur ces 40 gènes, le taux d'erreur est de 1.4%.

En comparaison, lorsque l'on réalise une simple étape de sélection, seuls 9 gènes sont sélectionnés plus de 10 fois. Parmi ces 9 gènes, 4 seulement font partie de la liste des 40 identifiés précédemment. Les 5 autres ne sont pas recensés dans liste Su2003.

5.5 Extensions pour les kNN

5.5.1 Application à la régression

Bien que nous ayons présenté les kNN dans le cadre de la classification supervisée, cet algorithme peut aussi être employé pour la prédiction d'une variable Y continue. Dans ce cas, l'étape de tesselation reste inchangée, seule l'étape de prédiction est modifiée : la valeur de Y est estimée en prenant la valeur moyenne de Y sur les k plus proches voisins. Dans ce nouveau cadre, la séparation entre les phases de tesselation et de révélation des valeurs Y_i demeure. L'algorithme d'agrégation présenté précédemment peut ainsi être directement appliqué dans le cadre de la régression. Nous présentons dans le paragraphe suivant une application de la stratégie d'agrégation à des données continues de spectrométrie.

Présentation des données

Nous considérons des données de spectrométrie utilisées pour déterminer le taux de graisse d'un morceau de viande de porc. Nous reprenons ici la présentation de ce jeu de données proposée par Borggaard et Thodberg (1992). Les données utilisées ont été obtenues grâce à un Tecator Infratec Food and Feed Analyzer (Tecator) travaillant dans le proche infrarouge (longueurs d'onde comprises entre 850 et 1050 nanomètres). Les spectres correspondent à l'absorbance mesurée pour 100 longueurs d'onde. Chaque spectre est normalisé en deux étapes : la ligne de base du spectre est estimée puis soustraite aux données, puis l'aire sous la courbe de chaque spectre est ramenée à 1. Les données normalisées sont données en figure 5.13.

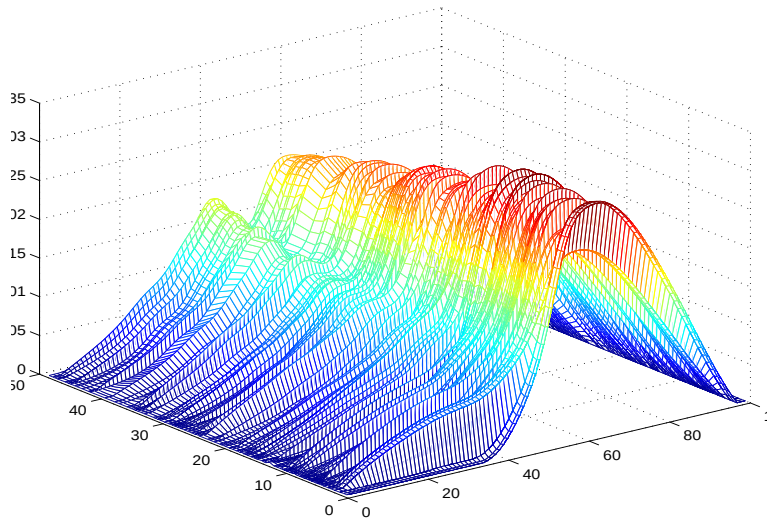


FIG. 5.13 – *Données Tecator. L'abscisse représente les longueurs d'onde allant de 850 à 1050 nanomètres, l'ordonnée représente le taux de graisse allant de 0% à 50 %, et l'axe z l'absorbance.*

On dispose donc de 215 observations (X,Y) où X^j est l'absorbance de la j^{eme} longueur d'onde, et Y le pourcentage de graisse du morceau considéré.

La qualité de l'estimation du taux de graisse est quantifiée par l'erreur moyenne de

Méthode	N_C	N_S	EMP
kNN brut	-	-	3.0
Sélection	-	6.7	2.58
Agrég.-Sélect.	60.5	6.7	2.80

TAB. 5.10 – *Nombre de groupes après agrégation, nombre de variables sélectionnées et erreur moyenne de prédiction des différentes règles (moyenne sur 100 rééchantillonnages)*

prédiction, définie par

$$EMP = \frac{1}{n_v} \sum_{i \in E_v} (Y_i - \hat{Y}_i)^2 \quad ,$$

où E_v est l'échantillon de validation. La première étude de ces données fut publiée par Borggaard et Thodberg (1992), où les auteurs proposent différents algorithmes de réseaux de neurones pour estimer le taux de graisse. Les EMP varient entre 0.65 et 10.88, suivant les méthodes. Amato *et al.* (2005) traitent le problème avec une méthode de décomposition dans une base ondelettes, et obtiennent un EMP de 0.05.

Nous utilisons ici un plan de rééchantillonnage où 170 observations sont utilisées comme données d'entraînement, et 45 comme données de validation. Il n'y a pas d'échantillon test, la sélection des paramètres est réalisée par validation K-fold. La validation K-fold consiste à diviser l'échantillon d'entraînement en K sous-échantillons de taille égale. Ces sous-échantillons jouent ensuite tour à tour le rôle de données test. On choisit ici $K = 5$, et nous notons ce plan de rééchantillonnage dans la suite [170,(5),45].

Résultats

Dans le cas des données Tecator, il semble qu'il existe un longueur particulièrement discriminante. En effet, la procédure de sélection de variables sélectionne la longueur d'onde 932 dans plus de 80% des rééchantillonnages et les longueurs d'onde voisines 928 et 930 plus de 70 fois. Ces longueurs d'onde avaient déjà été identifiées par Corona (2005) comme pertinentes pour la sélection. La procédure d'agrégation / sélection identifie ces mêmes longueurs d'onde avec les même fréquences.

Les performances des différentes règles de classification sont résumées dans la table 5.10. Comme on le voit, la procédure d'agrégation n'améliore pas les prédictions, et les dégrade même légèrement. Une explication possible à cette dégradation est qu'ici l'information peut être réellement portée par une unique longueur d'onde, ce qui serait confirmé par la fréquence de sélection de la longueur d'onde 932. La procédure d'agrégation serait alors inutile, et résulterait en une dilution de l'information pertinente.

Borne inférieure pour l'erreur de prédiction kNN

Les résultats du paragraphe précédent montrent que les performances obtenues par la méthode des kNN sont moins bonnes que celles obtenues par décomposition en ondelettes. On peut alors se demander si cette baisse de performance est liée à la stratégie d'agrégation / sélection, qui ne serait pas optimale, ou si cette baisse de performance est inhérente à l'utilisation des kNN. Autrement dit, est-ce qu'une autre méthode de sélection de variables ou d'agrégation appliquée aux kNN permettrait d'égaler les performances de la décomposition en ondelettes? Nous proposons ici une méthode pour obtenir une borne inférieure de l'erreur

des prédiction des kNN mesurée par LOO ou par la méthode K-fold, valable quelle que soit les procédures appliquées aux kNN ne touchant que l'étape de tesselation (c'est le cas des procédures de sélection et d'agrégation).

La méthode consiste à appliquer la méthode kNN à l'échantillon d'entraînement en utilisant la variable Y à la fois dans l'étape de tesselation et dans l'étape de prédiction. Dans l'étape de tesselation, on utilise la variable Y à la place des variables X^1,...,X^p pour calculer les distances entre individus. Ceci est possible puisque dans le cadre de la régression la variable Y est continue. L'étape de prédiction reste inchangée : on prédit à partir des valeurs de Y des k plus proches voisins.

Toutes les procédures usuelles d'estimation du risque peuvent alors être employées, comme la procédure LOO ou la procédure K-fold. De manière triviale, il est impossible de trouver de meilleurs voisins qu'en calculant les distances entre observations à partir de la variable Y directement. Les taux d'erreur obtenus par LOO ou par K-fold seront donc des bornes inférieures déterministes des procédures LOO et K-fold réalisées sur les vraies données, i.e. en calculant les distances à partir de X^1,...,X^p. Remarquons que la procédure que nous décrivons permet d'obtenir des bornes inférieures pour une valeur de k fixée.

Le calcul de la borne inférieure LOO à partir des données Tecator montre qu'il n'est pas possible de descendre en dessous d'une erreur de prédiction de 0.16. Mais une telle borne est optimiste pour le plan de simulation [170,(5),45] choisi. Il faut ici calculer la borne pour la procédure de validation 5-fold, qui correspond au plan [170,(5),45]. On obtient alors une borne inférieure de 0.271.

Bien qu'il semble possible d'améliorer les résultats obtenus avec la méthode kNN, nous pouvons maintenant conclure qu'il est impossible de battre ou même d'égaler les performances obtenues par la méthode de décomposition en ondelettes en utilisant les kNN.

5.5.2 Sur le choix de la distance

Dans les différents exemples d'application présentés dans cette partie, nous avons systématiquement employé la distance euclidienne canonique. D'autres métriques sont employées dans la littérature, et de nombreux auteurs ont montré que le choix d'une bonne métrique est critique pour l'obtention de bonnes performances de prédiction. Nous reprenons maintenant les différentes étapes du raisonnement exposé en partie 5.2 pour déterminer à quel point la stratégie d'agrégation peut être adaptée à d'autres mesures de distance.

L'ensemble du raisonnement est fondé sur la définition 5.2.1 de la redondance entre variables pour les kNN, que nous avons basée sur l'identité des contributions aux distances entre individus. Ce point est définitivement acquis, puisque des contributions identiques entraînent une tesselation identique. La conservation des distances lors d'une agrégation est mesurée par la perte d'inertie. Cette perte d'inertie est calculée sur les n observations disponibles, nous faisons donc implicitement l'hypothèse que la perte mesurée sur d'autres points aurait mené aux mêmes conclusions sur l'agrégation. Cette hypothèse simple implique une certaine stabilité de la distance qui n'est pas toujours acquise. Plusieurs auteurs (Friedman (1994), Hastie et Tibshirani (1996)) ont suggéré l'utilisation de métriques adaptatives pour les kNN, i.e. de métriques telles que les poids des différentes variables dans le calcul de la distance entre deux points changent en fonction de la localisation de ces points dans l'espace. Pour de telles métriques, la mesure de la conservation des distances semble plus complexe, ce qui rend difficile la tâche d'agrégation. De ce fait, le programme de minimisation 5.1 n'est pertinent que lorsque la métrique Σ considérée est non adaptative.

Dès lors que le programme 5.1 fait sens, se pose le problème de sa résolution. En partie 5.2,

nous nous sommes servis de la propriété d'additivité de la distance euclidienne canonique pour obtenir la forme simplifiée du programme de minimisation 5.2. La propriété d'additivité, bien que théoriquement non nécessaire, est en pratique requise pour établir l'optimalité locale de la procédure d'agrégation hiérarchique. En effet, celle-ci suppose que la perte d'inertie mesurée localement sur les deux variables agrégées peut être identifiée à la perte d'inertie globale due à l'agrégation des deux variables. Ainsi, bien que l'algorithme CAH soit applicable à toute métrique Σ, la propriété d'optimalité locale ne s'applique qu'aux métriques diagonales. On pourra en particulier envisager d'utiliser des distances de type ℓ^q, $q \geq 1$ comme métriques alternatives.

Dans les procédures de kNN pondérées (Royal (1966)), les k voisins votent en fonction de leur distance au point à classer. Les variables sont alors utilisées dans la phase de tesselation et dans la phase de vote. Toutefois, on utilise la même information sur les distances dans les deux phases. La procédure d'agrégation visant à conserver cette information, les résultats présentés dans cette partie sont généralisables aux procédures de kNN pondérés.

5.6 Agrégation supervisée pour CART

Dans la partie précédente, nous avons présenté notre stratégie de traitement de la redondance entre variables en traitant le cas de l'algorithme kNN. Cette stratégie est basée sur une analyse approfondie de l'algorithme de classification choisi. Une fois la redondance définie, un critère est sélectionné pour permettre de quantifier la redondance d'un groupes de variables (l'inertie dans le cas de kNN). On est alors en mesure de définir proprement l'objectif de traitement de la redondance comme un programme de minimisation du critère choisi, et de définir la distance entre deux variables pour l'algorithme de classification hiérarchique. Cette stratégie n'est donc pas limitée au seul exemple des kNN, et peut être généralisée à d'autres méthodes de classification supervisée. Nous montrons ici comment la stratégie peut être appliquée au cas de l'algorithme CART.

La description complète de l'algorithme CART se trouve au chapitre 4, partie 4.4.1. Nous ne reprenons pas ici cette présentation, nous rappelons simplement que le classement d'une observation est basé sur une série de question de type "Est-ce que la variable X^j est supérieure au seuil t_j ?". La question optimale pour l'objectif de classification est établie sur la base d'un indicateur de pureté. Au chapitre 4, le critère de pureté considéré était le taux d'erreur de classification. Dans la partie présente, nous ne précisons pas le critère de pureté choisi. En effet, tous les critères mentionnés sont basés sur la proportion d'observations de label 0 et 1 dans l'échantillon. Les opérations portant sur les variables X^1,...,X^p ne modifient donc pas ces critères.

Il faut bien remarquer que pour l'algorithme CART, l'information apportée par les variables est entièrement contenue dans l'ordre qu'elles engendrent sur les observations. En effet, lors de l'élaboration d'une question à partir d'une variable, seuls les rangs des individus sont pris en compte. Ainsi, deux variables engendrant les mêmes rangs engendreront des questions identiques. Ceci nous permet de définir la redondance pour l'algorithme CART :

Définition 5.6.1. *Deux variables sont redondantes pour l'algorithme CART si elles engendrent les mêmes rangs sur les individus.*

Nous cherchons donc l'agrégation préservant au mieux les rangs engendrés par les différentes variables sur les individus. On commence par introduire les variables R^1,...,R^p,

où R^j est la variable de rang correspondant à la variable X^j. Pour quantifier l'homogénéité de différentes variables de rang, on utilise à nouveau l'inertie :

$$I(R^1,...R^p) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}||R^i - R^j||^2$$

où R^i correspond au vecteur des rangs attribués par les variables 1,...p à l'observation i. Notons que plusieurs statistiques classiques de la littérature utilisent implicitement cette notion d'inertie calculée sur des rangs. Par exemple, le coefficient W de Kendall, classiquement utilisé pour comparer p classements, est proportionnel à un coefficient près au taux d'inertie précédent. En effet, ce coefficient est strictement proportionnel à la quantité

$$W = \sum_{k=1}^{n}\left(\sum_{i=1}^{p} R_k^i - \frac{\sum_{k=1}^{n}\sum_{i=1}^{p} R_k^i}{n}\right)^2 .$$

On a :

$$\begin{aligned}\sum_{k=1}^{n}\left(\sum_{i=1}^{p} R_k^i - \frac{\sum_{k=1}^{n}\sum_{i=1}^{p} R_k^i}{n}\right)^2 &= \sum_{k=1}^{n}\left(\sum_{i=1}^{p}\left(R_k^i - \frac{1}{n}\sum_{k=1}^{n} R_k^i\right)\right)^2 \\ &= \sum_{k=1}^{n}\sum_{i=1}^{p}\sum_{j=1}^{p}(R_k^i - \bar{R}^i)(R_k^j - \bar{R}^j) \\ &= \sum_{i=1}^{p}\sum_{j=1}^{p}\left\{\sum_{k=1}^{n}(R_k^i - \bar{R}^i)(R_k^j - \bar{R}^j)\right\}\end{aligned}$$

où $\bar{R}^i$ est la moyenne sur les individus k des rangs R_k^j. Le terme entre accolades est identifiable à un facteur près au coefficient de Spearman. En appliquant des résultats classiques (v. Saporta (1990) par exemple), on obtient

$$\begin{aligned}\sum_{k=1}^{n}\left(\sum_{i=1}^{p} R_k^i - \frac{\sum_{k=1}^{n}\sum_{i=1}^{p} R_k^i}{n}\right)^2 &= \frac{n^2-1}{12}\sum_{i=1}^{p}\sum_{j=1}^{p}\left(1 - \frac{6}{n(n^2-1)}\sum_{k=1}^{n}(R_k^i - R_k^j)^2\right) \\ &= Cste - \frac{1}{2n}\sum_{i=1}^{p}\sum_{j=1}^{p}(R^i - R^j)^2\end{aligned}$$

Ainsi, divers critères sur les rangs d'utilisation courante peuvent être vu comme des critères d'inertie.

L'ensemble des résultats sur la conservation de l'inertie de la partie précédente s'appliquent alors directement. En particulier, on peut définir le programme de traitement de la redondance dans le cas de l'algorithme CART comme suit :

Proposition 5. *Soient X_1,...,X_n n observations, pour lesquelles p variables X^1,...,X^p sont mesurées. Soient R^1,...,R^p les variables de rang associées aux variables initiales, et N_C un entier fixé tel que $N_C < p$. Trouver le meilleur regroupement de variables pour l'algorithme CART consiste à résoudre le programme de minimisation*

$$\min_{\mathcal{C}\in\mathbb{A}} P_{CART}\left(\left\{R^1,...,R^p\right\},\ \left\{Z^{\mathcal{C}_1},...,Z^{\mathcal{C}_{N_C}}\right\}\right) \quad , \tag{5.3}$$

où P_{CART} désigne la perte d'inertie pour les rangs et $\mathbb{A}$ est l'ensemble des partitions de $\{R^1,...,R^p\}$ en N_C groupes.

Toute la procédure algorithmique de la partie 5.2 peut alors être appliquée pour la résolution de ce nouveau programme de minimisation.

Notons que pour être applicable, la transformation proposée sur les données d'entraînement doit aussi être applicable aux données des échantillons de test et de validation. Ceci demande en pratique d'attribuer à chaque nouvelle observation un rang, fonction des données d'entraînement. En pratique, pour chaque variable, on attribuera à la nouvelle observation le rang de l'observation qui lui est la plus proche dans l'échantillon d'entraînement.

Ainsi, la stratégie proposée s'applique efficacement à l'algorithme CART. Remarquons que comme dans le cas des kNN, l'algorithme CART peut être utilisé dans le cadre de la régression. La procédure d'agrégation reste alors entièrement valable.

5.7 Discussion

Sur l'interprétation

Dans un article édifiant, Breiman (2001) expose son point de vue concernant le conflit entre la facilité d'interprétation et la performance d'un prédicteur. L'auteur commence par présenter le problème de la "multiplicité de bons modèles". Plusieurs modèles, très différents les uns des autres, sont susceptibles de donner les mêmes performances de prédiction. La solution que préconise Breiman est alors d'agréger ces différents modèles, pour obtenir de meilleures performances de prédiction, en soulignant que ce gain en prédiction aura pour coût une perte en interprétation. L'auteur illustre son point de vue en comparant deux algorithmes, CART et les forêts aléatoires (random forests). A une extrémité, CART est une règle de décision très simple à interpréter, mais son pouvoir prédictif est parfois limité :

While trees rate an A+ on interpretability, they are good, but not great, predictors. Give them, say, a B on prediction.

A l'autre extrémité, les forêts aléatoires améliorent sensiblement les performances de CART, mais la règle construite est très complexe :

So forest are A+ predictors. But their mechanism for producing a prediction is difficult to understand. Trying to delve into the tangled web that generated a plurality vote from 100 trees is an Herculean task. So on interpretability, they rate an F.

Pour Breiman, ces deux exemples montrent qu'il y a souvent un compromis à faire entre performance et interprétation, et que l'agrégation est un moyen efficace d'améliorer les résultats si l'on privilégie les performances de prédiction.

Il est intéressant de comparer l'approche de Breiman à celle présentée ici. Breiman propose d'agréger les modèles, et non les variables, car il ne fait aucune hypothèse sur l'origine du problème de multiplicité de bons modèles. A l'inverse, nous supposons que le problème de multiplicité est dû à la redondance entre variables. Cette hypothèse semble justifiée pour un grand nombre de problèmes, mais n'est jamais faite par Breiman (le mot redondance n'apparaît jamais dans l'article). De cette hypothèse naît l'idée de réaliser l'agrégation au niveau des variables, et non au niveau des modèles. La conséquence de ce déplacement est

double. D'une part le gain que nous visons est un gain en interprétation, et non un gain en performance. D'autre part le gain observé peut être à la fois un gain en interprétation et en performance. Ceci montre qu'il n'y a pas toujours conflit entre interprétation et prédiction, et qu'une prise en compte pertinente de la structure des données permet de gagner sur les deux aspects.

Comment mesure-t-on un gain en interprétation ? La chose est parfois simple, comme dans le cas des données Goat-Boat : nous savons *a priori* que l'information n'est pas portée par une longueur d'onde, mais par une gamme de longueurs d'onde. Cette idée est confirmée par l'identification d'une gamme hautement discriminante. Le gain en interprétation est alors validé par notre connaissance des données, et par le gain en performance obtenu grâce à l'agrégation. Dans le cas des puces à ADN, on peut s'interroger sur le gain obtenu lorsque l'on remplace une règle de décision basée sur quelques gènes obtenus par sélection par une règle de classification basée sur quelques groupes de gènes, chacun composé de plusieurs dizaines de gènes, obtenus par agrégation-sélection. Le nombre de variables explicatives augmente sensiblement, semblant rendre l'interprétation plus compliquée. En réalité, l'agrégation-sélection identifie l'information portée par les gènes. Lorsque plusieurs gènes portent une même information, ils sont tous sélectionnés. En effet, ce n'est plus la méthode statistique qui doit choisir quel gène est le représentant privilégié de cette information, mais la connaissance biologique du problème traité. A l'inverse, la sélection de variables réalise à la fois l'identification de l'information et le choix du représentant, ce dernier choix étant principalement guidé par l'échantillonnage. La procédure d'agrégation-sélection est donc plus adaptée pour l'interprétation des résultats, au sens où le choix parmi les gènes portant une même information est rendu au spécialiste, c'est-à-dire ici au biologiste.

Généralisation de la stratégie

Nous avons choisi de traiter le problème de la redondance de manière dédiée, c'est-à-dire de faire dépendre la définition de la redondance entre variables de la méthode sélectionnée pour réaliser la classification. Bien que la notion de redondance dépende de l'algorithme choisi, nous avons essayé de proposer une stratégie générique pour le traitement de la redondance. Cette stratégie fut présentée pour l'algorithme kNN, puis adapté à l'algorithme CART. Toutefois, la méthode d'agrégation proposée ne semble pas universelle. Considérons l'exemple de l'analyse discriminante linéaire diagonale (DLDA). Dans ce modèle, on suppose que les lois conditionnelles des données suivent des lois normales

$$\begin{aligned} X|Y=0 &\hookrightarrow \mathcal{N}(\mu_0,\, D) \ , \\ X|Y=1 &\hookrightarrow \mathcal{N}(\mu_1,\, D) \ , \end{aligned}$$

de même matrice de variance-covariance diagonale D. La règle de classification est alors très simple et peut s'écrire

$$\phi_{DLDA}(x) = \begin{cases} 1 & \text{si } \displaystyle\sum_{i=1}^{p} \frac{(\mu_1^i - \mu_0^i)x^i}{\sigma_i^2} + K > 0 \\ 0 & \text{sinon.} \end{cases}$$

Agréger les variables X^1 et X^2 consiste alors à trouver la combinaison linéaire $\alpha X^1 + \beta X^2$ telle que la nouvelle combinaison linéaire

$$\frac{(\mu_1^{12} - \mu_0^{12})x^{12}}{\sigma_{12}^2} + \sum_{i>2}^{p} \frac{(\mu_1^i - \mu_0^i)x^i}{\sigma_i^2} + K'$$

donne les classements les plus proches possibles de la règle initiale, où μ_1^{12}, μ_0^{12} et σ_{12}^2 sont les paramètres de la nouvelle variables créée. Les valeurs des paramètres α et β sont obtenues en minimisant

$$\begin{aligned}
& \sum_{k=1}^{n}\left(\sum_{i=1}^{p}\frac{(\mu_1^i-\mu_0^i)x_k^i}{\sigma_i^2}-\frac{(\mu_1^{12}-\mu_0^{12})x^{12}}{\sigma_{12}^2}-\sum_{i>2}^{p}\frac{(\mu_1^i-\mu_0^i)x_k^i}{\sigma_i^2}\right)^2 \\
= & \sum_{k=1}^{n}\left(\frac{(\mu_1^1-\mu_0^1)x_k^1}{\sigma_1^2}+\frac{(\mu_1^2-\mu_0^2)x_k^2}{\sigma_2^2}-\frac{(\mu_1^{12}-\mu_0^{12})x_k^{12}}{\sigma_{12}^2}\right)^2 \\
= & \sum_{k=1}^{n}\left(\left(\frac{(\mu_1^1-\mu_0^1)}{\sigma_1^2}-\frac{(\mu_1^{12}-\mu_0^{12})}{\sigma_{12}^2}\alpha\right)x_k^1+\left(\frac{(\mu_1^2-\mu_0^2)}{\sigma_2^2}-\frac{(\mu_1^{12}-\mu_0^{12})}{\sigma_{12}^2}\beta\right)x_k^2\right)^2 \quad .
\end{aligned}$$

Un simple calcul montre qu'il suffit de prendre des valeurs de α et β telles que

$$\alpha=\frac{(\mu_1^1-\mu_0^1)\sigma_2^2}{(\mu_1^2-\mu_0^2)\sigma_1^2}\beta$$

pour que la perte engendrée soit nulle. Le fait que l'agrégation n'engendre aucune perte retire toute pertinence à la procédure puisque toutes les agrégations sont alors équivalentes.

Caractérisation des classificateurs

La stratégie proposée n'étant pas universelle, il serait intéressant de pouvoir déterminer *a priori* les algorithmes pour lesquels la stratégie peut s'appliquer. Si l'on considère les exemples précédents, nous pouvons remarquer que dans les procédures kNN et CART, l'information qu'apportent les variables est une information sur les relations entre individus. Pour ces deux algorithmes, l'information peut être résumée dans un tableau (t_{ij}), i,$j = 1...n$ sur les individus. Dans le cas de l'algorithme kNN, (t_{ij}) représente le vecteur des contributions à la distance entre les individus i et j, et dans le cas CART, (t_{ij}) est un vecteur composé de 1 et de 0, la coordonnée k valant 1 si pour la variable k l'individu i est supérieur à l'individu j. Dans une telle représentation, traiter la redondance consiste à identifier les coordonnées des vecteurs qui sont systématiquement identiques dans les cases du tableau. Il n'est pas possible d'obtenir une telle représentation pour l'analyse discriminante de Fisher. En effet, l'information portée par les variables est une information sur les paramètres des distributions conditionnelles, plutôt qu'une information sur les relations entre individus. Ceci expliquerait les difficultés rencontrées pour l'application de notre stratégie à cette famille de méthode. Cette distinction entre les méthodes paramétriques d'une part qui modélisent directement les deux sous-populations, et les méthodes fondées sur les relations entre observations d'autre part est confortée par nos premiers travaux sur d'autres méthodes de classification. Nos premiers travaux concernant le traitement de la redondance dédié à la régression logistique montrent que l'on retrouve les mêmes difficultés de formalisation du problème que dans le cas de l'analyse discriminante. A l'inverse, l'algorithme des SVM semble être adapté pour la stratégie que nous proposons.

Dans l'algorithme SVM, l'information pertinente apportée par les variables est résumée par la matrice de Gram. Les SVM linéaires font clairement partie des méthodes fondées sur les relations entre observations : chaque cellule (t_{ij}) est composée du vecteur des contributions au produit scalaire de chaque variable. Le traitement de la redondance consistera à fusionner les variables de manière à conserver au mieux les éléments de cette matrice, c'est-à-dire à conserver au mieux les produits scalaires. La perte due à l'agrégation des variables peut être

quantifiée par la distance entre l'ancienne matrice de Gram G et la nouvelle matrice G'. On pourra par exemple utiliser la distance ℓ_2 :

$$d^2(G,G') = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(< X_i,X_j > - < X_i^{new},X_j^{new} >\right)^2 \quad , \tag{5.4}$$

où X_i^{new} est la représentation de l'individu i dans l'espace des variables compressées. Trouver l'agrégation à N_C groupes minimisant la perte au sens de la distance 5.4 pose un problème d'optimisation difficile, mais une solution approchée peut être trouvée en utilisant l'algorithme CAH, comme dans les cas des algorithmes CART et kNN. Notons toutefois que contrairement aux problèmes d'optimisation 4 et 5, la perte SVM définie à partir de la distance 5.4 n'est pas additive. La solution approchée trouvée par l'algorithme CAH risque donc d'être plus éloignée de la solution optimale que dans le cas des kNN ou de CART.

Bibliographie

Tecator Infratec Food and Feed Analyzer. Données disponibles à l'URL http://lib.stat.cmu.edu/datasets/tecator.

Akaike, H. (1973). Information theory as an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, (B. Petrov et F. Csaki, ed.), 267–281. Akademiai Kiado, Budapest.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. et Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA.* **96 (12)** 6745–6750.

Amato, U., Antoniadis, A. et De Feiss, I. (2005). Dimension reduction in functional regression with applications. *Comp. Stat. and Data Anal.* **50 (9)** 2422–2446.

Anderberg, M. (1973). *Cluster Analysis for Applications.* New York: Academic Press, Inc.

Barron, A., Birge, L. et Massart, M. (1995). Risk bounds for model selection via penalization. *Probability Theory and Related Fields.* **113** 301–413.

Bartlett, P. L., Boucheron, S. et Lugosi, G. (Oct., 2000), Model selection and error estimation. Technical Report 508, Department of Economics and Business, Universitat Pompeu Fabra.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M. et Yakhini, Z. (2000). Tissue classification with gene expression profiles. *J. Comp. Biol.* **7 (3-4)** 559–583.

Bi, J., Bennett, M., Embrechts, C., Breneman, C. et Song, M. (2003). Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research.* **3** 1229–1243.

Biau, G., Bunea, F. et Wegkamp, M. (2005). Functional classification in Hilbert spaces. *IEEE Trans. Inform. Theory.* **51** 2163–2172.

Birgé, L. et Massart, P. (2001a). Gaussian model selection. *J. Eur. Math. Soc.* **3** 203–268.

Birgé, L. et Massart, P. (2001b), A generalized C_p criterion for Gaussian model selection. Technical report, Publication n°647, Universités de Paris 6 & Paris 7.

Birgé, L. et Rozenholc, Y. (2002), How many bins should be put in a regular histogram. Technical report, Publication n°721, Université Paris 6 & Paris 7.

Blanchard, G., Bousquet, O. et Massart, P. Statistical performance of support vector machines. Submitted.

Blum, A. et Langley, P. (1997). Selection of relevant features and exemples in machine learning. *Artificial Intelligence.* 245–271.

Borggaard, C. et Thodberg, H. (1992). Optimal minimal neural interpretation of spectra. *Anal. Chem.* **64** 545–551.

Boser, B., Guyon, I. et Vapnik, V. (1992). A training algorithm for optimal margin classifier. In *Fifth Annual Workshop on Computational Learning Theory*, 144–152. ACM.

Boucheron, S., Bousquet, O. et Lugosi, G. (2005). Theory of classification: some recent advances. *ESAIM: Probability and Statistics*. **9** 323–375.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*. **16 (3)** 199–231.

Breiman, L., Friedman, J., Olshen, R. et Stone, C. (1984). *Classification and regression trees*. Wadsworth International, Belmont, CA.

Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Jr, M. et Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*. **97** 262–267.

Castellan, G. (2000). *Sélection d'histogrammes ou de modèles exponentiels de polynomes par morceaux à l'aide d'un critère de type Akaike*. PhD thesis, Université Paris-Sud.

Cherkassky, V. et Ma, Y. (2003). Comparison of model selection for regression. *Neural Computation*. **15 (7)** 1691–1714.

Corona, L., F. Amaury. (2005). Input selection and function approximation using the som: an application to spectrometric modeling. In *WSOM'05, 5th Workshop on Self-Organizing Maps*.

Cristianini, N. et Shawe-Taylor, J. (1999). *An introduction to support vector machines*. Cambridge University Press.

Daudin, J. et Mary-Huard, T. (2005). Model selection in classification: the swapping method. In *International Symposium on Applied Stochastic Models and Data Analysis*. Brest (France).

Dettling, M. (2003). Revealing predictive gene clusters with supervised algorithms. In *DSC 2003 Working Papers*.

Dettling, M. et Bühlmann, P. (2002). Supervised clustering of genes. *Genome Biology*. **3 (12)** 1–15.

Devroye, L., Gyorfi, L. et Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Springer.

Diaz-Uriarte, R., (2004). Molecular signatures from gene expression data. Unpublished.

Ding, C. et Peng, H. (2003). Minimum redundancy feature selection from microarray gene expression data. Proceedings of the Computational Systems Bioinformatics.

Donoho, D. et Johnstone, I. (1994). Ideal spatial adaption by wavelet shrinkage. *Biometrika*. 425–455.

Dougherty, E. (2001). Small-sample issue for microarray-based classification. *Comparative and Fonctional Genomics*. 28–34.

Dudoit, S., Fridlyand, J. et Speed, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Amer. Statist. Assoc.* **97** 77–87.

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *J. Amer. Statist. Assoc.* 461–470.

Efron, B. (2004). The estimation of prediction error: covariance penalties and cross-validation. *J. Amer. Statist. Assoc.* **99** 619–642.

Fisher, R. (1936). The use of multiple measurments in taxinomic problems. *Annals of Eugenics*. **7** 179–188.

Fix, E. et Hodges, J. (1991a). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. (B. (Ed.), ed.), Chapitre Discriminatory analysis- nonparametric discrimination: Consistency principles. IEEE Computer Society Press, Los Alamitos, CA, Reprint of original work from 1952.

Fix, E. et Hodges, J. (1991b). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. (B. (Ed.), ed.), Chapitre Nonparametric Discrimination: small sample performance. IEEE Computer Society Press, Los Alamitos, CA, Reprint of original work from 1952.

Fort, G. et Lambert-Lacroix, S. (2004). Classification using partial least squares with penalized logistic regression. *Bioinformatics*. **21 (7)** 1104–1111.

Fraley, C. et Raftery, A. E. (1998). How many clusters ? which clustering method ? answer via model-based cluster analysis. *The Computer Journal*. **41** 578–588.

Friedman, J. (1994), Flexible metric nearest neighbour classification. Technical report, Stanford University.

Fukumizu, K., Bach, F. et Jordan, M. (2004). Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*. **5** 73–99.

Geman, D., d'Avignon, C., Naiman, D. et Winslow, R. (2004). Classifying gene expression profiles from pairwise mRNA comparisons. *Statist. Appl. in Genetics and Molecular Biology*. **3 (1)** –.

Gey, S. et Lebarbier, E. (2002), A cart based algorithm for detection of multiple change-points in the mean for large samples. Technical report, Publication Université Paris-VI 10.

Gey, S. et Nedelec, E. (2005). Model selection for cart regression trees. *IEEE Trans. Inform. Theory*. To appear.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. et Lander, E. (1999). Class prediction and discovery using gene expression data. *Science*. **286** 531–537.

Guyon, I. et Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*. **3** 1157–1182.

Guyon, I., Weston, J., Barnhill, S. et Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*. **46** 389–422.

Györfi, L. (2002). *Principles of nonparametric learning*. Springer Wien NY.

Hall, M. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th International Conf. on Machine Learning*.

Harman, H. (1973). *Modern Factor Analysis*. University of Chicago Press.

Hastie, T., Tibshirani, R., Eisen, M., Brown, P., Scherf, U., Weinstein, J., Alizadeh, A., Staudt, L. et Botstein, D. (1999), Gene shaving: a new class of clustering methods for expression arrays. Technical report, Stanford: Department of Statistics, Stanford University.

Hastie, T., Tibshirani, R. et Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York.

Hastie, T. et Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Anal. and Machine Intel.* **18 (6)** 607–616.

Hechenbichler, K. et Schliep, K. (2004), Weighted k-nearest-neighbor techniques and ordinal classification. Technical report, Discussion Paper 399, SFB 386, Ludwig-Maximilians University Munich.

Jornsten, R. et Yu, B. (2003). Simultaneous gene clustering and subset selection for sample classification via mdl. *Bioinformatics.* **19 (9)** 1100–1109.

Kearns, M., Mansour, Y., Ng, A. et Ron, D. (1997). An experimental and theoretical comparison of model selection methods. *Machine Learning.* **27 (1)** 7–50.

Klecka, W. (1980). *Discriminant Analysis.* (S. U. P. S. on Quantitative Applications in the Social Sciences, ed.). 07-019. Beverly Hills: Sage Publications.

Kohavi, R. et John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence.* **97** 273–324.

Koller, D. et Sahami, M. (1996). Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning.*

Krishnapuram, B., Hartemink, A., Carin, L. et Figueiredo, M. (2004a). A bayesian approach to joint feature selection and classifier design. *IEEE Trans. Pattern Anal. and Machine Intel.* **26** 1105–1111.

Krishnapuram, B., Carin, L. et Hartemink, A. (2004b). *Kernel methods in computational biology.* (B. Schölkopf, K. Tsuda, et J.-P. Vert, ed.), Chapitre Gene expression analysis: Joint feature selection and classifier design. MIT Press.

Lal, T., Chapelle, O., Weston, J. et Elisseeff, A. Embedded methods. Available at http://www.kyb.mpg.de/publications/pdfs/pdf3012.pdf.

Lavielle, M. (1999). Detection of multiple changes in a sequence of dependent variables. *Stoch. Proc. and Appl.* **83** 79–102.

Lebarbier, E. (2005). Detecting multiple change-points in the mean of gaussian process by model selection. *Signal processing.* **85** 717–736.

Lebarbier, E. et Mary-Huard, T. (2006). Une introduction au critère BIC : fondements théoriques et interprétation. *Journal de la SFdS.* To appear.

Lugosi, G. et Zeger, K. (1995). Concept learning using complexity regularization. *IEEE Trans. Inform. Theory.* **42 (1)** 48–54.

Mallows, C. (1973). Some comments on C_p. Technometrics. **12** *591–612.*

Mammen, E. *et* Tsybakov, A. *(1999). Smooth discrimination analysis.* Ann. Statist. **27 (6)** *1808–1829.*

Mary-Huard, T., Robin, S., Daudin, J., Bitton, F., Cabannes, E. *et* Hilson, P. *(2004). Spotting effect in microarray experiments.*

Mary-Huard, T., Picard, F. *et* Robin, S. *(2006).* Mathematical and Computational Methods in Biology. *Chapitre Introduction to Statistical Methods for Microarray Data Analysis. Hermann: Paris.*

Massart, P. *(2000). Some applications of concentration inequalities to statistics.* Annales de la Faculté des Sciences de Toulouse. **IX** *245–303.*

Massart, P. *et* Nédélec, E. *(2006), Risk bounds for statistical learning. Technical report, To appear in Ann. Statist.*

Michiels, S., Koscielny, S. *et* Hill, C. *(2005). Prediction of cancer outcome with microarrays: a multiple random validation strategy.* Lancet. **365** *488–492.*

Nobel, A. *(2002). Analysis of a complexity-based pruning scheme for classification trees.* IEEE Trans. Inform. Theory. **48 (8)** *2362–2368.*

Press, W. H., Flannery, B. P., Teukolsky, S. A. *et* Vetterling, W. T. *(1988).* Numerical recipes in C: the art of scientific computing. *Cambridge University Press New York, NY, USA.*

Rakotomamonjy, A. *(2003). Variable selection using SVM-based criteria.* Journal of Machine Learning Research. **3** *1357–1370.*

Ripley, B. *et* Hjort, N. *(1995).* Pattern Recognition and Neural Networks. *Cambridge University Press.*

Rissanen, J. *(1987). Stochastic complexity.* J. R. Statist. Soc. B. **49** *223–239.*

Rossi, F. *et* Villa, N. *(2006). Support vector machine for functional data classification.* Neural Computing. **69 (7–9)** *223–239.*

Royal, R. *(1966).* A class of non parametric estimators of a smooth regression function. *PhD thesis, Stanford University, Stanford, CA.*

Saporta, G. *(1990).* Probabilités, analyse des données et statistique. *Paris: Editions Technip.*

Schölkopf, B. *et* Smola, A. *(2002).* Learning with Kernels, *Support Vector Machines, Regularization, Optimization and Beyond. The MIT Press.*

Schwarz, G. *(1978). Estimating the dimension of a model.* Ann. Statist. **6** *461–464.*

Simon, H. *(1993). General lower bounds on the number of examples needed for learning probabilistic concepts. In* Proceedings of the Sixth Annuals ACM Conference on Computationnal Learning Theory*, 402–412. Association for Computing Machinery.*

Smola, A. *et* Schölkopf, B. *(1998). From regularization operators to support vector kernels. In* Advances in Neural Information Processing Systems*, volume 10, 343–349.*

Su, Y., Murali, T., Pavlovic, V., Schaffer, M. *et* Kasif, S. *(2003). Rankgene: identification of diagnostic genes based on expression data.* Bioinformatics. **19 (12)** *1578–1579.*

Tibshirani, R. *et* Knight, K. *(1999). The covariance inflation criterion for adaptive model selection.* J. R. Statist. Soc. B. **61 (3)** *529–546.*

Tsamardinos, I. *et* Aliferis, C. *(2003). Toward principled feature selection: relevancy, filters and wrappers. In* Ninth International Workshop on Artificial Intelligence and Statistics*. Key West, Florida, USA.*

Tsybakov, A. *(2004). Optimal aggregation of classifiers in statistical learning.* Ann. Statist. **32 (1)** *135–166.*

Tuleau, C. *(2005).* Sélection de variables pour la discrimination en grande dimension et classification de données fonctionnelles. *PhD thesis, Université Paris-Sud XI.*

Vapnik, V. *(1998).* Statistical learning theory. *Wiley, NY.*

Vapnik, V. *et* Chervonenkis, A. *(1981). Necessary and sufficient conditions for the uniform convergence of means to their expectations.* Theory of probability and its applications. **26** *821–832.*

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. *et* Vapnik, V. *(2000).* Advances in neural information processing systems*. Chapitre Feature selection for SVMs. MIT Press.*

Xing, E., Jordan, M. *et* Karp, R. *(2001). Feature selection for high-dimensional genomic microarray data. In* Proceedings of the Eighteenth International Conference on Machine Learning*.*

Xiong, M., Li, W., Zhao, J., Jin, L. *et* Boerwinkle, E. *(2001). Feature (gene) selection in gene expression-based tumor classification.* Mol Genet Metab. **73 (3)** *239–247.*

Xu, M. *et* Zhang, L. *A robust method for generating discriminative gene clusters. Unpublished.*

Ye, J. *(1998). On measuring and correcting the effects of data mining and model selection.* J. Amer. Statist. Assoc. **93** *120–131.*

Yu, L. *et* Liu, H. *(2004a). Efficient feature selection via analysis of relevance and redundancy.* Journal of Machine Learning Research. **5** *1205–1224.*

Yu, L. *et* Liu, H. *(2004b). Redundancy based feature selection for microarray data. In* Proceedings of the Tenth ACM SIGKDD Conference. *Seattle, Washington, USA.*

Zhu, J. *et* Hastie, T. *(2003). Classification of gene microarrays by penalyzed logistic regression.* Biostatistics. **3** *427–43.*

Annexe A

Une introduction au critère BIC : fondements théoriques et interprétation

L'analyse des données de grande dimension pose inévitablement la question de la sélection de modèles. Dans le cadre de la classification supervisée, nous avons vu au chapitre 1 que la construction d'un classificateur Φ_n^* performant passe par le choix raisonné d'une classe de classificateurs. La stratégie SRM propose un cadre rigoureux pour le choix d'une classe dans une liste fixée. Plus généralement, la sélection de modèles est un problème récurrent en classification supervisée, mais aussi dans le cadre de la sélection de variables (cf. chapitre 4), de la détection de ruptures (Lebarbier (2005)) ou du choix du nombre de composantes d'un modèle de mélange (Fraley et Raftery (1998)). La littérature sur la sélection de modèles, et plus particulièrement sur la sélection de modèles par critère pénalisé, est donc vaste.

L'article qui suit présente le critère BIC (Schwarz (1978)), qui repose sur une approche bayésienne de la sélection de modèles. L'ensemble des étapes de la construction du critère ainsi que les hypothèses posées à chacune de ces étapes y sont détaillées. Cette description du critère BIC est aussi l'occasion de présenter une conception de la sélection de modèles différente de celle présentée au chapitre 1. Dans la stratégie SRM, la définition du meilleur modèle dépend de la quantité d'information dont on dispose, i.e. du nombre d'observations. En effet, dans le cadre de la SRM, et plus généralement dans le cadre de la sélection de modèles adaptative, le modèle optimal est par définition celui qui réalise le meilleur compromis biais-variance. Le terme de variance mesure la difficulté d'estimation pour le nombre d'observations disponibles. La stratégie consiste alors à sélectionner un estimateur dont les performances soient comparables à celles que l'on aurait obtenues en sélectionnant directement l'estimateur dans la classe optimale.

La définition bayésienne du modèle optimale est différente. Le modèle optimal est le modèle contenant l'estimateur optimal. S'il existe plusieurs modèles contenant l'estimateur optimal, le modèle optimal est le plus petit d'entre eux au sens de la dimension (si l'on considère des classes paramétriques). L'objectif est alors de proposer une stratégie de sélection qui garantisse asymptotiquement que le modèle choisi est le modèle optimal. On parle alors de critère "consistant".

Les deux stratégies (sélection par critère adaptatif et sélection par critère consistant) sont donc très différentes. La différence majeure vient de la définition du modèle optimal, qui dépend de la quantité d'information pour la stratégie adaptative, et n'en dépend pas pour la

stratégie consistante. Dans ce manuscrit, nous étudions les données de grande dimension. Dans un tel contexte, il est nécessaire de proposer des méthodes adaptées pour la faible quantité de données dont on dispose. C'est pourquoi les approches présentées aux chapitres 3 et 4 sont toutes deux des approches adaptatives.

Cette étude a fait l'objet d'un article (Lebarbier et Mary-Huard (2006)) qui sera prochainement publié au Journal de la Société Française de Statistique.

Une introduction au critère BIC : fondements théoriques et interprétation

Emilie Lebarbier, Tristan Mary-Huard*

Résumé

Dans cet article, nous proposons une discussion sur le critère de sélection de modèles BIC (Bayesian Information Criterion). Afin de comprendre son comportement, nous décrivons les étapes de sa construction et les hypothèses nécessaires à son application en détaillant les approximations dont il découle. En s'appuyant sur la notion de quasi-vrai modèle, nous reprécisons la propriété de "consistance pour la dimension" définie pour BIC. Enfin, nous mettons en évidence les différences de fond entre le critère BIC et le critère AIC d'Akaike en comparant leurs propriétés.

Mots clés : Critère de sélection de modèles ; Critère bayésien ; Approximation de Laplace ; Consistance pour la dimension.

Abstract

In this article we propose a discussion on the Bayesian model selection criterion BIC (Bayesian Information Criterion). In order to understand its behaviour, we describe the steps of its construction as well as the hypotheses required for its application and the approximations needed. Relying on the notion of quasi-true model, we explain the "dimension-consistency" property of BIC. Finally we show the basic differences between BIC and AIC via the comparison of their respective properties.

Keywords : Model selection criterion; Bayesian criterion; Laplace approximation; Dimension-consistency.

* INA-PG (dépt OMIP) / INRA (dépt MIA), 16 rue Claude Bernard, Paris Cedex 05. Emilie.Lebarbier@inapg.fr. Tristan.Maryhuard@inapg.fr

1 Introduction

La sélection de modèles est un problème bien connu en statistique. Lorsque le modèle est fixé, la théorie de l'information fournit un cadre rigoureux pour l'élaboration d'estimateurs performants. Mais dans un grand nombre de situations, les connaissances *a priori* sur les données ne permettent pas de déterminer un unique modèle dans lequel se placer pour réaliser une inférence. C'est pourquoi, depuis la fin des années 70, les méthodes pour la sélection de modèles à partir des données ont été développées. Les exemples classiques d'application de ces méthodes sont la sélection de variables, ou le choix du nombre de composantes d'un mélange de lois, d'un ordre d'auto-régression, ou de l'ordre d'une chaîne de Markov.

L'une des réponses apportées par les statisticiens au problème de la sélection de modèles est la minimisation d'un critère pénalisé. Les premiers critères apparaissant dans la littérature sont l'Akaike Information Criterion (AIC, Akaike 1973), le Bayesian Information Criterion (BIC, Schwarz 1978), le Minimum Description Length (MDL, Rissanen 1978) et le C_p de Mallows (Mallows 1974). Parmi ces critères, AIC et BIC ont été largement diffusés et appliqués. D'un point de vue théorique, beaucoup de travaux ont été réalisés concernant leurs propriétés statistiques et leur adaptation à des modèles spécifiques. En particulier, plusieurs versions corrigées du critère AIC ont été proposées : AICC (Hurvich et Tsai 1989) et c-AIC (Sugiura 1978) pour de petites tailles d'échantillons par rapport au nombre de paramètres à estimer ; AICR (Ronchetti 1985) pour une régression avec erreurs non-gaussiennes ; QAIC (Burnham et Anderson 2002) et c-QAIC (Shi et Tsai 1998) pour des données sur-dispersées. Il existe ainsi une littérature très fournie sur la sélection de modèles par critère pénalisé, qui se développe encore actuellement avec l'apparition d'outils sophistiqués de probabilité, comme par exemple les inégalités de concentration et de déviation, permettant à la fois la construction de critères et leur étude.

Nous nous intéressons ici au critère BIC qui se place dans un contexte bayésien de sélection de modèles. Bien que couramment utilisé par les statisticiens et largement décrit, certains points de sa construction et de son interprétation sont régulièrement omis dans les démonstrations proposées dans la littérature. Il est bien connu que le critère BIC est une approximation du calcul de la vraisemblance des données conditionnellement au modèle fixé. Cependant les résultats théoriques utilisés sont souvent peu explicités, tout comme les hypothèses nécessaires à leurs

applications. Par ailleurs, l'interprétation de BIC et la notion de "consistance pour la dimension" ne sont pas toujours très claires pour les utilisateurs.
L'objectif de cet article est d'expliciter ces différents points. Dans un premier temps, nous reprenons de manière détaillée la démonstration de l'ensemble des approximations asymptotiques sur lesquelles repose la construction du critère BIC, en précisant les hypothèses et le rôle des distributions *a priori* posées sur les modèles et les paramètres des modèles (Partie 2). Dans un deuxième temps, nous explicitons le sens des notions de probabilité *a priori* et *a posteriori*. Cela permettra de préciser l'objectif du critère BIC qui est loin d'être explicite au regard de sa définition, et de discuter de l'hypothèse que le "vrai" modèle appartient aux modèles en compétition, hypothèse généralement posée par les auteurs (Partie 3). Enfin, nous présentons et commentons les méthodes de comparaison entre BIC et AIC usuellement proposées, ces deux critères étant souvent mis en concurrence dans la pratique (Partie 4).

2 Construction du critère BIC

Dans cette partie, nous présentons la construction du critère BIC. Pour cela, nous nous appuyons sur les propositions de Raftery (1995).

On dispose d'un n-échantillon $X = (X_1, \ldots, X_n)$ de variables aléatoires indépendantes de densité inconnue f et l'objectif est de l'estimer. Pour cela, on se donne une collection finie de modèles paramétrés $\{M_1, \ldots, M_m\}$. Un modèle M_i est l'ensemble des densités g_{M_i} de paramètre θ_i appartenant à l'espace vectoriel Θ_i de dimension K_i :

$$M_i = \{g_{M_i,\theta_i} \ ; \theta_i \in \Theta_i\}$$

Il s'agit de choisir un modèle parmi cette collection de modèles.

Le critère BIC se place dans un contexte bayésien : θ_i et M_i sont vus comme des variables aléatoires et sont munis d'une distribution *a priori*. Notons

– $P(M_i)$ la distribution *a priori* du modèle M_i. Elle représente le poids que l'on souhaite attribuer à ce modèle. Par exemple, à partir d'informations que peut détenir l'utilisateur, on peut suspecter que la vraie f est proche de certains modèles particuliers et on peut alors

donner à ces modèles un poids plus important. En général cependant cette distribution *a priori* est supposée non-informative (uniforme), ne privilégiant aucun modèle :

$$P(M_1) = P(M_2) = \ldots = P(M_m) = 1/m.$$

– $P(\theta_i|M_i)$ la distribution *a priori* de θ_i sachant le modèle M_i. L'ensemble des paramètres θ_i est en effet défini pour le modèle M_i considéré. Nous verrons que cette distribution n'intervient pas dans la forme du critère BIC mais la qualité des approximations faites peut en dépendre.

BIC cherche à sélectionner le modèle M_i qui maximise la probabilité *a posteriori* $P(M_i|X)$:

$$M_{BIC} = \underset{M_i}{\operatorname{argmax}}\, P(M_i|X). \tag{1}$$

En ce sens BIC cherche à sélectionner le modèle le plus vraisemblable au vu des données. La partie 3 est plus particulièrement consacrée à l'interprétation de la probabilité *a posteriori* de M_i. D'après la formule de Bayes, $P(M_i|X)$ s'écrit

$$P(M_i|X) = \frac{P(X|M_i)P(M_i)}{P(X)}. \tag{2}$$

Nous supposons dans toute la suite que la loi *a priori* des modèles M_i est non informative. Sous cette hypothèse et d'après (1) et (2), la recherche du meilleur modèle ne nécessite que le calcul de la distribution $P(X|M_i)$. Ce calcul s'obtient par l'intégration de la distribution jointe du vecteur des paramètres θ_i et des données X conditionnellement à M_i, $P(X,\theta_i|M_i)$, sur toutes les valeurs de θ_i :

$$P(X|M_i) \;=\; \int_{\Theta_i} P(X,\theta_i|M_i)d\theta_i = \int_{\Theta_i} g_{M_i}(X,\theta_i)P(\theta_i|M_i)d\theta_i,$$

où $g_{M_i}(X,\theta_i)$ est la vraisemblance correspondant au modèle M_i de paramètre θ_i :

$$g_{M_i}(X,\theta_i) = \prod_{k=1}^{n} g_{M_i}(X_k,\theta_i) = P(X|\theta_i,M_i).$$

On réécrit cette intégrale sous la forme

$$P(X|M_i) \;=\; \int_{\Theta_i} \mathrm{e}^{g(\theta_i)}d\theta_i, \text{ où } \; g(\theta_i) = \log\left(g_{M_i}(X,\theta_i)P(\theta_i|M_i)\right).$$

La probabilité $P(X|M_i)$ est appelée *vraisemblance intégrée pour le modèle* M_i. Le calcul exact de cette probabilité est rarement possible, on l'approche alors en utilisant la méthode d'approximation de Laplace :

Proposition 2.1 Approximation de Laplace. *Soit une fonction* $L : \mathbb{R}^d \to \mathbb{R}$ *telle que* L *est trois fois différentiable sur* $\mathbb{R}^d$ *et atteint un unique maximum sur* $\mathbb{R}^d$ *en* $u^\star$*. Alors*

$$\int_{\mathbb{R}^d} e^{nL(u)} du = e^{nL(u^\star)} \left(\frac{2\pi}{n}\right)^{\frac{d}{2}} |-L''(u^\star)|^{-\frac{1}{2}} + O(n^{-1}).$$

Nous détaillons en Annexe A la démonstration de cette approximation proposée par Tierney et Kadane (1986) et discutons de plus les hypothèses pour l'application de ce résultat à des fonctions L qui dépendent de n comme c'est le cas ici puisque nous l'appliquons à la fonction :

$$L_n(\theta_i) = \frac{g(\theta_i)}{n} = \frac{1}{n}\sum_{k=1}^{n} \log\left(g_{M_i}(X_k,\theta_i)\right) + \frac{\log\left(P(\theta_i|M_i)\right)}{n}. \tag{3}$$

Nous notons

- $\theta_i^\star = \text{argmax}_{\theta_i \in \Theta_i} L_n(\theta_i)$,
- $A_{\theta_i^\star}$ l'opposé de la matrice hessienne des dérivées secondes partielles de la fonction $L_n(\theta_i)$ en θ_i :

$$A_{\theta_i^\star} = -\left[\frac{\partial^2 L_n(\theta_i)}{\partial\theta_i^j \partial\theta_i^l}\right]_{j,l} |_{\theta_i=\theta_i^\star}\ , \tag{4}$$

  où θ_i^j est la $j$ème composante du vecteur des paramètres θ_i.

Nous obtenons

$$P(X|M_i) \quad = \quad \mathrm{e}^{g(\theta_i^\star)} \left(\frac{2\pi}{n}\right)^{K_i/2} |A_{\theta_i^\star}|^{-1/2} + O_P(n^{-1}),$$

ou encore

$$\log\left(P(X|M_i)\right) = \log\left(g_{M_i}(X,\theta_i^\star)\right) + \log\left(P(\theta_i^\star|M_i)\right) - \frac{K_i}{2}\log(n) + \frac{K_i}{2}\log(2\pi) - \frac{1}{2}\log\left(|A_{\theta_i^\star}|\right) + O_P(n^{-1}). \tag{5}$$

La difficulté maintenant est l'évaluation de $\theta_i^\star$ et de $A_{\theta_i^\star}$. Asymptotiquement, $\theta_i^\star$ peut être remplacé par l'estimateur du maximum de vraisemblance $\hat{\theta}_i$:

$$\hat{\theta}_i = \underset{\theta_i \in \Theta_i}{\text{argmax}} \frac{1}{n} g_{M_i}(X,\theta_i),$$

et $A_{\theta_i^\star}$ remplacé par $I_{\hat{\theta}_i}$, où $I_{\hat{\theta}_i}$ est la matrice d'information de Fisher pour une observation définie par :

$$I_{\hat{\theta}_i} = -\mathbb{E}\left(\left[\frac{\partial^2 \log\left(g_{M_i}(X_1,\theta_i)\right)}{\partial\theta_i^j \partial\theta_i^l}\right]_{j,l} |_{\theta_i=\hat{\theta}_i}\right).$$

En effet, lorsque n est grand, $\log(g_{M_i}(X,\theta_i)P(\theta_i|M_i))$ se comporte comme $\log(g_{M_i}(X,\theta_i))$, qui croît avec n tandis que $\log(P(\theta_i|M_i))$ reste constant. Remplacer $\theta_i^\star$ par $\hat{\theta}_i$ et $A_{\theta_i^\star}$ par $I_{\hat{\theta}_i}$ dans (5) introduit un terme d'erreur en $n^{-1/2}$ (cf Annexe B). Nous obtenons :

$$\begin{aligned} \log\left(P(X|M_i)\right) &= \log\left(g_{M_i}(X,\hat{\theta}_i)\right) - \frac{K_i}{2}\log(n) \\ &+ \log\left(P(\hat{\theta}_i|M_i)\right) + \frac{K_i}{2}\log(2\pi) - \frac{1}{2}\log\left(|I_{\hat{\theta}_i}|\right) + O_P(n^{-1/2}). \end{aligned} \tag{6}$$

Par continuité et par la convergence en probabilité de l'estimateur du maximum de vraisemblance $\hat{\theta}_i$, nous obtenons que le premier terme est de l'ordre de $O_P(n)$, le second de l'ordre de $O(\log(n))$ et tous les derniers termes de l'ordre de $O_P(1)$. En négligeant les termes d'erreurs $O_P(1)$ et $O_P(n^{-1/2})$, nous obtenons

$$\log\left(P(X|M_i)\right) \approx \log\left(g_{M_i}(X,\hat{\theta}_i)\right) - \frac{K_i}{2}\log(n).$$

C'est de cette approximation que le critère BIC est issu. Plus précisément, pour le modèle M_i il correspond à l'approximation de $-2\log P(X|M_i)$. BIC est donc défini par :

$$BIC_i = -2\log\left(g_{M_i}(X,\hat{\theta}_i)\right) + K_i\log(n). \tag{7}$$

Le modèle sélectionné par ce critère est

$$M_{BIC} = \underset{M_i}{\operatorname{argmin}}\, BIC_i.$$

Remarque 1. Nous avons fait l'hypothèse que la loi des modèles $P(M_i)$ est uniforme. La prise en compte d'une information *a priori* sur les modèles est toutefois possible, on utilise alors le critère modifié :

$$-2\log\left(g_{M_i}(X,\hat{\theta}_i)\right) + K_i\log(n) - 2\log\left(P(M_i)\right)$$

On se reportera à l'article de Kass et Wasserman 1996 pour une discussion sur la spécification d'une loi *a priori* informative sur les modèles.

Remarque 2. L'erreur en $O_P(n^{-1/2})$ dans l'égalité (6) est négligeable lorsque n tend vers l'infini. Par contre l'erreur d'approximation en $O_P(1)$ peut perturber le choix du modèle final même si les deux premiers termes sont prépondérants quand n est grand puisqu'elle est systématique. Néanmoins pour certaines distributions *a priori* sur les paramètres θ_i, le terme d'erreur peut être plus petit que $O_P(1)$ (Raftery 1995 ; Kass et Wasserman 1995).

3 Interprétation du critère BIC

L'une des difficultés du critère BIC est son interprétation. La question est la suivante : quel est le modèle que l'on cherche à sélectionner par le critère BIC ? A ce niveau, les notions de probabilité *a priori* ou *a posteriori* d'un modèle sont peu explicites et ne donnent pas une idée intuitive de ce que BIC considère être un "bon" modèle. Les considérations asymptotiques présentées ici vont nous permettre d'interpréter cette notion de meilleur modèle, de déterminer ce que l'on entend par probabilité *a posteriori* d'un modèle, et de préciser en quel sens BIC est un critère "consistant pour la dimension". Cette interprétation nous permettra aussi de discuter la nécessité de l'hypothèse d'appartenance du vrai modèle à la liste des modèles considérés.

Concernant la consistance du critère BIC, les premiers travaux remontent au milieu des années 80, et visaient à établir la consistance du critère dans des cas simples, par exemple dans le cadre de familles exponentielles (Hartigan 1985 ; Haughton 1988 ; Poskitt 1987). D'autres cas ont été ensuite étudiés, comme la consistance de BIC pour la détermination du nombre de composantes d'un modèle de mélange (Chernoff et Lander 1995 ; Dacunha-Castelle et Gassiat 1997 ; Keribin 1998) ou pour la détermination de l'ordre d'une chaîne de Markov (Csiszar et Shields 2000), et font encore actuellement l'objet de publications (Azaïs, Gassiat, et Mercadier 2003). Le lecteur ne trouvera ici qu'une présentation simplifiée de la démonstration de la consistance de BIC dans un cas simple, visant à établir proprement les notions de probabilités *a priori* et *a posteriori*. Une démonstration plus rigoureuse de cette consistance pourra par exemple être trouvée dans (Dudley et Haughton 1997) ou (Dudley et Haughton 2002).

3.1 Le "quasi-vrai" modèle

Nous reprenons ici la remarquable présentation de cette notion proposée par Burnham et Anderson (2002).

Rappelons que la densité à estimer est f. Par simplicité, on se place dans le cas simple où les m modèles M_1,...,M_m sont supposés emboîtés, i.e. $\Theta_1 \subset \Theta_2 ... \subset \Theta_m$. La pseudo-distance de Kullback-Leibler (appelée dans la suite distance KL) entre deux densités f et g est définie par :

$$d_{KL}(f,g) = \int_\Omega \log\left(\frac{f(x)}{g(x)}\right) f(x)dx.$$

Par abus de notation, on définit la distance KL de f au modèle M_i par :

$$d_{KL}(f,M_i) \quad = \quad \inf_{\theta_i} \; d_{KL}(f,g_{M_i}(.,\theta_i)). \tag{8}$$

Puisque les modèles sont emboîtés, la distance KL est une fonction décroissante de la dimension K_i. On note M_t le modèle à partir duquel cette distance ne diminue plus. Du point de vue de la distance KL, M_t doit être préféré à tous les sous-modèles M_i, $i = 1,...,t-1$ puisqu'il est plus proche de f. Par ailleurs, M_t doit aussi être préféré à tous les modèles d'ordre supérieurs M_i, $i = t+1,...,m$, puisqu'ils sont plus compliqués que M_t sans pour autant être plus proches de f : ces modèles sont donc surajustés. Nous allons montrer que le critère BIC est consistant pour ce modèle particulier, désigné par Burnham et Anderson comme le modèle "quasi-vrai". Pour n supposé grand, on s'intéresse à la différence :

$$BIC_i - BIC_t, \;\; i \neq t.$$

Premier cas : $i < t$

D'après (7), on a :

$$\begin{aligned} BIC_i - BIC_t &= -2\log(g_{M_i}(X,\hat{\theta}_i)) + 2\log(g_{M_t}(X,\hat{\theta}_t) + (K_i - K_t)\log(n) \\ &= 2n\left[-\frac{1}{n}\sum_{k=1}^{n}\log(g_{M_i}(x_k,\hat{\theta}_i)) + \frac{1}{n}\sum_{k=1}^{n}\log(g_{M_t}(x_k,\hat{\theta}_t))\right] + (K_i - K_t)\log(n) \\ &= 2n\left[\frac{1}{n}\sum_{k=1}^{n}\log\left(\frac{f(x_k)}{g_{M_i}(x_k,\hat{\theta}_i)}\right) - \frac{1}{n}\sum_{k=1}^{n}\log\left(\frac{f(x_k)}{g_{M_t}(x_k,\hat{\theta}_t)}\right)\right] + (K_i - K_t)\log(n). \end{aligned}$$

Les deux dernières sommes sont des estimateurs convergeant en probabilité des quantités $d_{KL}(f,M_i)$ et $d_{KL}(f,M_t)$, respectivement (cf Ripley 1995). Pour n grand, on a donc :

$$BIC_i - BIC_t \approx 2n[d_{KL}(f,M_i) - d_{KL}(f,M_t)] + (K_i - K_t)\log(n).$$

Cette approximation, bien que déterministe, suffit à expliciter le comportement asymptotique de $BIC_i - BIC_t$: le premier terme domine et tend vers $+\infty$ avec n. On en déduit donc qu'asymptotiquement les modèles M_i, $i = 1,...,t-1$ sont disqualifiés par le critère BIC.

Deuxième cas : $i > t$

Dans ce cas là, on reconnaît dans le terme $2\log(g_{M_i}(X,\hat{\theta}_i)) - 2\log(g_{M_t}(X,\hat{\theta}_t))$ la statistique du

test du rapport de vraisemblance pour deux modèles emboîtés, qui sous l'hypothèse Ho suit asymptotiquement une loi du Chi-2 à $(K_i - K_t)$ degrés de liberté. On a donc :

$$BIC_i - BIC_t \approx -\chi^2_{(K_i - K_t)} + (K_i - K_t)\log(n).$$

C'est ici le second terme qui domine et tend vers $+\infty$ avec n, les modèles M_i, $i = t+1$,...,m sont eux aussi disqualifiés. Le terme en $\log(n)$ joue donc un rôle fondamental : il assure que le critère BIC permet de converger vers le quasi-vrai modèle. Cette convergence vers le quasi-vrai modèle, même s'il est emboîté dans un modèle plus complexe, est appelée consistance pour la dimension.

Il nous est maintenant possible d'interpréter clairement ce que l'on entend par probabilité *a posteriori* du modèle M_i. Elle s'estime à partir des différences $\Delta BIC_i = BIC_i - BIC_{min}$, où BIC_{min} désigne la plus petite valeur observée de BIC sur les m modèles. On a :

$$P(M_i|X) \approx \frac{exp(-\frac{1}{2}\Delta BIC_i)}{\sum_{l=1}^{m} exp(-\frac{1}{2}\Delta BIC_l)}.$$

Cette probabilité tend vers 1 pour le modèle quasi-vrai lorsque n tend vers l'infini, et vers 0 pour tous les autres. Au vu des considérations précédentes, nous pouvons définir cette probabilité comme la probabilité que M_i soit le modèle quasi-vrai de la liste considérée, sachant les données.

3.2 Le vrai modèle fait-il partie de la liste ?

La question de savoir si le vrai modèle ayant engendré les données doit apparaître ou non dans la liste des modèles considérés est longtemps demeurée en suspens dans la littérature consacrée au critère BIC. Bien que nulle part cette hypothèse apparaisse comme nécessaire dans la construction du critère BIC, les auteurs (Schwarz 1978 ; Raftery 1995) posent souvent cette hypothèse, sans toutefois préciser à quelle étape du raisonnement elle intervient. On peut alors se demander si l'hypothèse est nécessaire du point de vue théorique d'une part, pour démontrer la consistance du critère BIC, et du point de vue pratique d'autre part, pour appliquer le critère BIC.

Du point de vue théorique, nous avons vu dans la partie précédente que l'hypothèse n'est pas nécessaire pour établir la consistance vers le quasi-vrai modèle. En réalité, l'hypothèse n'est

nécessaire que si l'on s'intéresse à la convergence de BIC vers le vrai modèle : elle sert alors à identifier le quasi-vrai modèle, vers lequel la convergence est toujours assurée, au vrai modèle.

Du point de vue pratique, supposer que le vrai modèle fait partie des modèles en compétition semble peu réaliste, excepté dans de rares cas où le phénomène étudié est simple et bien décrit. Cette constatation n'a aucune conséquence lorsque l'on souhaite comparer des modèles entre eux puisque l'hypothèse n'est pas nécessaire pour la dérivation du critère BIC. Remarquons toutefois que le quasi-vrai modèle de la collection peut être arbitrairement loin (au sens de la distance KL) du vrai modèle. La consistance du critère BIC ne garantit donc pas la qualité du modèle sélectionné, qui dépend fondamentalement du soin apporté par l'expérimentateur pour construire la collection des modèles envisagés.

4 Comparaison des critères AIC et BIC

Les critères AIC (Akaike 1973) et BIC ont souvent fait l'objet de comparaisons empiriques (Burnham et Anderson 2002 ; Bozdogan 1987). Dans la pratique, il a été observé que le critère BIC sélectionne des modèles de dimension plus petite que le critère AIC, ce qui n'est pas surprenant puisque BIC pénalise plus qu'AIC (dès que $n > 7$). La question qui nous intéresse ici est de savoir si l'on peut réellement comparer les performances de ces deux critères, et si oui sur quelles bases. Cette question se justifie pleinement au vu de la littérature. Bien souvent les conclusions des auteurs sur les performances d'AIC et de BIC sont plus guidées par l'idée que se font les auteurs d'un "bon critère" que par la démonstration objective de la supériorité d'un critère sur l'autre, comme l'illustre la présentation des deux critères par Burnham et Anderson (Burnham et Anderson 2002).

Nous commencerons donc par rappeler les propriétés respectives d'AIC et de BIC, avant de considérer les méthodes proposées pour leur comparaison.

4.1 Propriétés des critères

Nous avons vu que le critère BIC est consistant pour le modèle quasi-vrai. Montrons maintenant qu'AIC ne partage pas cette propriété. En effet l'objectif du critère AIC est de choisir le

modèle M_i vérifiant :

$$M_{AIC} = \underset{M_i}{\operatorname{argmin}} \mathbb{E}\left[\int \log\left(\frac{f(x)}{g_{M_i}(x,\hat{\theta}_i)}\right) f(x)dx\right], \tag{9}$$

en minimisant le critère suivant :

$$M_{AIC} = \underset{M_i}{\operatorname{argmin}} -2\log(g_{M_i}(X,\hat{\theta}_i)) + 2K_i.$$

En reprenant le raisonnement asymptotique détaillé pour BIC sur l'exemple de la partie 3.1, on a :

$$\begin{aligned} AIC_i - AIC_t &\approx 2n[d_{KL}(f,M_i) - d_{KL}(f,M_t)] + 2(K_i - K_t) \qquad i < t \\ AIC_i - AIC_t &\approx -\chi^2_{(K_i - K_t)} + 2(K_i - K_t) \qquad\qquad\qquad\quad i > t. \end{aligned}$$

Les modèles M_i, $i < t$ sont asymptotiquement disqualifiés. En revanche, la probabilité de disqualifier les modèles M_i, $i > t$ ne tend pas vers 0, puisque le terme issu des pénalités $2(K_i - K_t)$ ne diverge pas quand n tend vers l'infini. AIC n'est donc pas consistant pour le quasi-vrai modèle (une démonstration complète de ce résultat peut être trouvée dans (Hannan 1980)).

Ce résultat ne démontre en rien la supériorité de BIC sur AIC, car ce dernier n'a pas été conçu pour être consistant, mais dans une optique d'efficacité prévisionnelle. En effet, l'objectif d'AIC est de choisir parmi les m modèles considérés le modèle vérifiant (9), ou de manière équivalente :

$$M_{AIC} = \underset{M_i}{\operatorname{argmin}} \left(d_{KL}(f,M_i) + \mathbb{E}\left[\int_\Omega \log\left(\frac{g_{M_i}(x,\bar{\theta}_i)}{g_{M_i}(x,\hat{\theta}_i)}\right) f(x)dx\right]\right),$$

où $\bar{\theta}_i$ est la valeur de θ_i vérifiant (8). Le premier terme mesure la distance de f au modèle M_i (biais) et le deuxième la difficulté d'estimer $g_{M_i}(.,\bar{\theta}_i)$ (variance). Sélectionner un modèle par AIC revient donc à chercher le modèle qui fait le meilleur compromis biais - variance pour le nombre de données n dont on dispose. La prise en compte de la taille de l'échantillon vient de ce que l'on somme sur tous les échantillons possibles la distance KL entre f et $g_{M_i}(.,\bar{\theta}_i)$. Le meilleur modèle au sens AIC dépend donc de n.

L'objet de cet article n'étant pas de démontrer les propriétés d'AIC, nous nous contenterons ici de dire que dans le cadre gaussien et à nombre de modèles candidats M_i fini, AIC est efficace alors que BIC ne l'est pas (cf Birgé et Massart 2001).

4.2 Méthodes de comparaison

Maintenant qu'il est clair que la notion de meilleur modèle est différente pour AIC et BIC, nous pouvons examiner les méthodes proposées dans la littérature pour les comparer. Nous présentons ici deux points de vue usuels, basés sur des simulations, qui nous permettront de conclure plus généralement sur l'ensemble des méthodes de comparaison proposées. Chaque méthode est discutée d'un point de vue théorique, puis d'un point de vue pratique.

4.2.1 Sélection du vrai modèle

La première méthode est basée sur la simulation de données à partir d'un modèle M_t, qui fait partie de la liste des modèles $M_1 \subset ... \subset M_t \subset ... \subset M_m$ considérés par la suite. Puisque l'on connaît le vrai modèle, on regarde sur un grand nombre de simulations lequel des deux critères le retrouve le plus souvent (Bozdogan 1987). Théoriquement, on peut considérer deux situations, suivant la taille de l'échantillon :

- Lorsque n est petit, le choix optimal pour AIC n'est pas forcément M_t. Ce dernier peut être trop complexe pour la quantité de données n disponible, et il peut exister un modèle M_i de dimension plus petite réalisant un meilleur compromis biais-variance.
- Lorsque n est (très) grand, M_t est meilleur que tous ses sous-modèles, puisque la variance est négligeable devant le biais. Toutefois, un modèle légèrement sur-ajusté aura le même biais que M_t et sa variance, bien que plus grande, sera de toute façon elle aussi négligeable devant le biais. Du point de l'efficacité, les deux modèles sont donc admissibles.

Dans les deux cas, AIC choisit donc un modèle optimal (au sens biais-variance) sans pour autant choisir M_t. Du point de vue théorique, ce type de comparaison favorise donc le critère BIC, puisque lui seul a pour objectif de sélectionner le vrai modèle[1].

En pratique, les résultats obtenus sur des simulations donnent des conclusions très différentes suivant la taille de l'échantillon et la complexité du vrai modèle. Généralement les modèles simulés sont très simples. BIC sélectionne alors le vrai modèle, et AIC le vrai modèle ou un modèle plus grand, ce qui amène les auteurs à conclure que BIC est plus performant pour le choix du vrai modèle. Toutefois, lorsque le modèle est plus complexe, par exemple composé d'une multitude de "petits effets", on constate que BIC devient moins performant qu'AIC car même

1. Dans le cas de simulations, le vrai modèle fait bien partie de la liste.

pour de grandes tailles d'échantillon BIC sélectionne des modèles sous-ajustés.

4.2.2 Sélection d'un modèle prédictif

La deuxième méthode est basée sur la qualité de la prédiction (Burnham et Anderson 2002). On simule des données (x_i,y_i) et l'objectif est de sélectionner un modèle de régression pour faire de la prédiction. Les données simulées sont divisées en deux échantillons X^1 et X^2, de taille respective n_1 et n_2. X^1 est utilisé pour choisir un modèle de prédiction dans la liste $M_1 \subset ... \subset M_m$, et X^2 sert pour le calcul de la performance de prédiction du modèle choisi, mesurée par :

$$MSE = \frac{1}{n_2} \sum_{i \in X^2} (\hat{y}_i - y_i)^2.$$

D'un point de vue théorique, le critère AIC est favorisé puisqu'il prend explicitement en compte la difficulté d'estimation des paramètres dans le terme de variance. Par ailleurs, dans la plupart des cas les données simulées sont gaussiennes. Les critères MSE et AIC sont alors équivalents. Ainsi, le critère gagnant est celui qui a été créé pour répondre à la question posée.

En pratique, on observe généralement de meilleures performances pour AIC que pour BIC, mais pour une raison toute autre que celle invoquée ci-dessus. La pénalité en $\log(n)$ de BIC fait que les modèles sélectionnés par ce critère sont souvent sous-ajustés. En conséquence, le biais de ces modèles est grand, les performances de prédiction ne sont pas satisfaisantes. Toutefois, ici encore les résultats dépendent du modèle simulé et de la taille de l'échantillon. En particulier lorsque le modèle M_t est simple et n_1 grand, BIC peut montrer de meilleures performances qu'AIC.

Lorsque l'objectif de l'analyse statistique est la prédiction, une alternative possible à la sélection d'un "meilleur" modèle pour la prédiction est de considérer l'ensemble des prédictions faites à partir des différents modèles et d'en faire la synthèse, en considérant une moyenne pondérée des prédictions de chaque modèle. Cette alternative a été explorée et peut se justifier théoriquement du point de vue bayésien (Hoeting, Madigan, Raftery, et Volinsky 1999 ; Madigan et Raftery 1994) comme du point de vue de la théorie de l'information (Burnham et Anderson 2002). Dans le cadre bayésien, on parle alors de Bayesian Model Averaging (BMI), et le critère BIC peut être utilisé pour établir la pondération de chaque modèle. Les résultats obtenus en pratique sont souvent meilleurs que ceux obtenus en réalisant la prédiction à l'aide d'un unique

modèle. Le lecteur intéressé se reportera avec profit à l'article de Hoeting *et al.* (1999).

4.2.3 Quel critère choisir ?

La conclusion est que le choix d'un critère de sélection de modèles doit être conditionné par l'objectif de l'analyse et la connaissance des données. De nombreux auteurs ont remarqué que BIC et AIC sont utilisés indifféremment quel que soit le problème posé, bien que n'ayant pas le même objectif (Reschenhoffer 1996). Pourtant, choisir entre ces deux critères revient à choisir entre un modèle explicatif et un modèle prédictif. Ce choix devrait donc être argumenté en fonction de l'objectif des utilisateurs. Par ailleurs, les résultats sur données simulées montrent à quel point les performances pratiques sont fonction des situations, en particulier de la complexité du vrai modèle et des modèles candidats, et de la taille de l'échantillon. Ces considérations pratiques et théoriques montrent qu'il n'existe pas de critère universellement meilleur. Seuls l'objectif de l'expérimentateur et sa connaissance des données à analyser peuvent donner un sens à la notion de supériorité d'un critère sur l'autre.

5 Conclusions

Nous avons éclairci les hypothèses, les objectifs et les propriétés du critère BIC. Les considérations de cet article ne prétendent pas être exhaustives : en particulier, nous n'avons pas présenté ici les liens entre BIC et les facteurs de Bayes (Kass et Raftery 1995), ni la place de BIC dans la théorie de la complexité stochastique développée par Rissanen (1987). Nous terminerons par quelques remarques sur les différents points abordés dans cet article.

Il est important de souligner que la construction du critère BIC réalisée en partie 2 a été obtenue dans un cadre asymptotique. En pratique, les tailles d'échantillons sont souvent trop petites pour rentrer dans ce cadre, ce qui peut poser différents problèmes. D'une part les approximations réalisées, comme la méthode de Laplace, peuvent se révéler très inexactes. D'autre part, on constate que la loi *a priori* sur les paramètres $P(\theta_i|M_i)$ n'apparaît pas dans le critère (7). Cette absence est rassurante, puisqu'elle signifie qu'une mauvaise spécification de $P(\theta_i|M_i)$ n'aura aucun poids sur la sélection de modèles. Toutefois, cette absence ne se justifie qu'asymptotiquement, lorsque l'on remplace θ^* par $\hat{\theta}$ dans l'équation (6), autrement dit lorsque l'on peut

faire l'hypothèse que l'information apportée par $P(\theta_i|M_i)$ est négligeable comparée à l'information apportée par l'échantillon. Cette hypothèse n'est pas convenable lorsque n est petit, sauf si $P(\theta_i|M_i)$ est supposée uniforme, ce qui n'est pas toujours possible. On retrouve ici la difficulté propre à l'application de critères asymptotiques à des cas concrets.

Malgré ces considérations, dans un grand nombre de cas l'application du critère BIC fournit des résultats très satisfaisants. Tout d'abord, il existe des cas pour lesquels on souhaite explicitement décrire la structure de la population étudiée. La sélection de modèles dans le cadre du modèle de mélange (Fraley et Raftery 1998) est un bon exemple : l'objectif est de trouver le nombre de composantes du mélange, qui sera ensuite interprété pour distinguer autant de sous-populations distinctes. C'est pourquoi les auteurs (Celeux et Soromenho 1996 ; McLachlan et Peel 2000) s'accordent pour dire que BIC donne de meilleurs résultats qu'AIC : AIC est logiquement disqualifié puisqu'il n'est pas consistant. Notons toutefois que pour cet objectif, d'autres critères plus performants que BIC ont été proposés pour la sélection du nombre de composantes dans un mélange (Biernacki, Celeux, et Govaert 2000).

Par ailleurs, les comparaisons entre AIC et BIC sont généralement réalisées avec une collection finie de modèles. Mais il existe des situations où le nombre de modèles à considérer augmente avec le nombre de données. On peut citer les exemples de la détection de ruptures ou de l'estimation de vraisemblance par histogramme. Pour ces situations, il a été observé que la dimension des modèles choisis avec AIC explose alors que BIC semble proche de l'efficacité (Lebarbier 2005 ; Castellan 1999). Dans ces cas précis, AIC est donc battu sur son propre terrain ! Le paradoxe n'est ici qu'apparent. Lorsque le nombre de modèles à considérer augmente plus vite que la taille de l'échantillon, Birgé et Massart (2001) ont démontré que seuls des critères ayant un terme en $\log(n)$ peuvent être efficaces. Bien que possédant un terme en $\log(n)$ pour d'autres raisons, le critère BIC est alors plus efficace qu'AIC.

Annexes

Pour simplifier l'écriture, nous notons $\theta = \theta_i$, $M_i = M$ et $P(\theta|M) = P(\theta)$. Tous les résultats démontrés dans cette partie requièrent que $P(\theta) \neq 0$ et plus particulièrement que $\log(P(\theta))$ reste borné pour tout θ. On suppose aussi que la vraisemblance et ses dérivées ne dégénèrent pas en $\theta = \hat{\theta}$.

Annexe A

Démontrons la proposition 2.1. Pour plus de simplicité, nous supposons que la fonction L est définie sur $\mathbb{R}$ mais le résultat s'étend facilement à des fonctions de $\mathbb{R}^d$. L'idée principale derrière le résultat de la proposition 2.1 est que l'intégrale

$$\int_{\mathbb{R}} \mathrm{e}^{nL(u)} du \tag{10}$$

est concentrée autour son maximum (unique) quand n est grand. Pour obtenir l'ordre de l'erreur d'approximation, nous effectuons tout d'abord le développement de Taylor de la fonction $L(u)$ autour de son maximum $L(u^\star)$ à l'ordre 3 :

$$L(u) = L(u^\star) + (u-u^\star)L'(u^\star) + \frac{(u-u^\star)^2}{2}L''(u^\star) + \frac{(u-u^\star)^3}{6}L'''(u^\star) + O((u-u^\star)^4).$$

L'intégrale (10) devient :

$$\int_{\mathbb{R}} \mathrm{e}^{nL(u)} du = \mathrm{e}^{nL(u^\star)} \int_{\mathbb{R}} \mathrm{e}^{\frac{n(u-u^\star)^2}{2}L''(u^\star)} \mathrm{e}^{\frac{n(u-u^\star)^3}{6}L'''(u^\star)} \mathrm{e}^{O(n(u-u^\star)^4)} du \tag{11}$$

puisque par hypothèse, $L'(u^\star) = 0$. Nous cherchons à faire apparaître les moments d'une loi gaussienne. Pour cela, nous développons le second terme exponentiel sous l'intégrale en utilisant le développement de la fonction exponentielle à l'ordre 2 autour de 0 :

$$\mathrm{e}^x = 1 + x + \frac{x^2}{2} + O(x^3).$$

L'intégrale dans l'expression (11) vaut

$$\begin{aligned}
&\int_{\mathbb{R}} \mathrm{e}^{n\frac{(u-u^\star)^2}{2}L''(u^\star)} du \\
+ &\int_{\mathbb{R}} \mathrm{e}^{n\frac{(u-u^\star)^2}{2}L''(u^\star)} \left[\frac{n(u-u^\star)^3}{6}L'''(u^\star) + \frac{n^2(u-u^\star)^6}{72}L'''(u^\star)^2\right] du \\
+ &\int_{\mathbb{R}} \mathrm{e}^{n\frac{(u-u^\star)^2}{2}L''(u^\star)} \left[O(n(u-u^\star)^4) + O(n^2(u-u^\star)^7) + O(n^2(u-u^\star)^8)\right] du.
\end{aligned}$$

En posant

$$\sigma = \frac{1}{\sqrt{-nL''(u^\star)}} \quad \text{et} \quad v = \frac{(u-u^\star)}{\sigma}, \tag{12}$$

nous avons pour $i \geq 0$,

$$\int_{\mathbb{R}} (u-u^\star)^i \mathrm{e}^{n\frac{(u-u^\star)^2}{2}L''(u^\star)} du = \sqrt{2\pi}\sigma\sigma^i \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} v^i \mathrm{e}^{-\frac{v^2}{2}} dv.$$

On reconnaît le moment d'ordre i d'une variable aléatoire V de loi gaussienne centrée réduite à une constante près. Les moments d'ordre impair étant nuls, nous obtenons

$$\begin{aligned}\int_{\mathbb{R}} \mathrm{e}^{nL(u)}du &= \mathrm{e}^{nL(u^\star)}\sqrt{2\pi}\sigma\left[\mathbb{E}[V^0]+\frac{n^2\sigma^6}{72}\mathbb{E}[V^6]+\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} O\left(v^4 n\sigma^4\right)\mathrm{e}^{-\frac{v^2}{2}}dv\right]\\ &= \mathrm{e}^{nL(u^\star)}\sqrt{2\pi}\sigma\left[1+\frac{5}{24}\frac{1}{nL''(u^\star)^3}+\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} O\left(\frac{v^4}{n}\right)\mathrm{e}^{-\frac{v^2}{2}}dv\right].\end{aligned}$$

Le terme d'erreur est d'ordre en $1/n$ et il est facile de voir que les termes d'erreurs supérieurs sont d'ordre inférieur ou égal à $1/n$. L'intégrale (10) devient

$$\int_{\mathbb{R}} \mathrm{e}^{nL(u)}du = \mathrm{e}^{nL(u^\star)}\sqrt{-\frac{2\pi}{nL''(u^\star)}}\left[1+O(n^{-1})\right]. \tag{13}$$

Ce qui conclut la preuve de la proposition 2.1.

Ce résultat est obtenu pour une fonction L qui ne dépend pas de n. Si ce n'est pas le cas, le résultat n'est plus si évident. En effet, en effectuant un développement de Taylor de $L(u)$ autour de $L(u^\star)$ à l'ordre 4, on obtient l'expression explicite du terme d'erreur en $O(n^{-1})$ (dans (13)) qui est :

$$\frac{1}{n}\left[\frac{5}{24}\frac{L'''(u^\star)^2}{L''(u^\star)^3}-\frac{1}{8}\frac{L''''(u^\star)}{L''(u^\star)^2}\right]+O(n^{-2}).$$

Ainsi pour que l'égalité (13) reste valable, il faut que les deux coefficients précédant $1/n$ restent bornés en n (Tierney et Kadane 1986). Il sera donc nécessaire de poser des conditions de régularité sur la fonction L qui assurent les conditions précédentes.

Ici nous cherchons à appliquer la proposition à la fonction L_n (définie par l'équation 3) qui dépend de n. Nous pouvons supposer que la convergence des fonctions d'intérêts vers des quantités possédant des bonnes propriétés suffit. Dans notre cas, on dispose de la convergence en probabilité de L_n vers une quantité qui ne dépend pas de n. En effet, on peut décomposer $L_n(\theta)$ de la manière suivante :

$$L_n(\theta) = LGn(\theta) + B_n(\theta) \quad ,$$

où

$$LG_n(\theta) = \frac{1}{n}\sum_{k=1}^{n}\log\left(g_M(X_k,\theta)\right) \quad et \quad B_n(\theta) = \frac{1}{n}\log\left(P(\theta)\right). \tag{14}$$

Sous la condition $\mathbb{E}[\log(g_M(X_1,\theta))] < \infty$, la loi faible des grands nombres assure la convergence en probabilité de $LG_n(\theta)$ vers $\mathbb{E}[\log(g_M(X_1,\theta))]$. De plus, $B_n(\theta) \xrightarrow{p.s} 0$. Ce qui conclut sur la convergence en probabilité de $L_n(\theta)$ vers $\mathbb{E}[\log(g_M(X_1,\theta)]$. Par le même raisonnement, on obtient facilement la convergence des dérivées de la fonction L_n.

Annexe B

L'objectif est ici de donner l'ordre des erreurs d'approximation de $\theta^\star$ par $\hat{\theta}$ et de $A_{\theta^\star}$ par $I_{\hat{\theta}}$.

B1 - Approximation de $\theta^\star$ par $\hat{\theta}$

On cherche à montrer

$$\sqrt{n}(\theta^\star - \hat{\theta}) = O_P(1).$$

On décompose ce terme en la somme de deux termes

$$\sqrt{n}(\hat{\theta} - \theta_0) + \sqrt{n}(\theta_0 - \theta^\star),$$

où θ_0 est l'unique maximum de $\mathbb{E}[\log(g_M(X_1,\theta))]$ (l'unicité existe sous la condition d'identifiabilité du modèle). Il suffit alors de montrer que ces deux termes sont bornés en probabilité.
Il est bien connu que sous des conditions de régularité, l'estimateur du maximum de vraisemblance $\hat{\theta}$ satisfait $\sqrt{n}(\hat{\theta} - \theta_0) \overset{\mathcal{L}, n\to\infty}{\longrightarrow} \mathcal{N}(0, I_{\theta_0}^{-1})$. Ce qui assure

$$\sqrt{n}(\hat{\theta} - \theta_0) = O_P(1).$$

Pour le second terme, le résultat est assuré par la convergence en probabilité de $L_n'(\theta)$ vers la même quantité que $LG_n'(\theta)$ qui est $E\left[\frac{\partial \log(g_M(X_1,\theta))}{\partial\theta}\right]$ (démonstration similaire à celle présentée en fin de l'Annexe A). Puisque les hypothèses sur cette limite ont déjà été posées pour obtenir le résultat sur $\hat{\theta}$, on obtient la convergence en probabilité de $\theta^\star$ vers θ_0 (cf Lemme 5.10 dans Van Der Vaart 1998) et la normalité asymptotique (cf Théorème 5.21 dans Van Der Vaart 1998).

B2 - Approximation de $A_{\theta^\star}$ par $I_{\hat{\theta}}$

Avant de commencer la démonstration, nous démontrons que $\sqrt{n}(A_\theta - I_\theta)$ est borné en probabilité. D'après les définitions de A_θ (4), $LG_n(\theta)$ et $B_n(\theta)$ (14), on écrit

$$A_\theta - I_\theta = LG_n''(\theta) - I_\theta + \frac{1}{n}[\log(P(\theta))]'',$$

où la dérivée signifie dérivée par rapport à θ. En notant que $\mathbb{E}[LG_n''(\theta)] = I_\theta$, sous la condition que $\mathbb{E}\left(\left[\frac{\partial^2 \log(g_M(X_1,\theta))}{\partial\theta^j \partial\theta^l}\right]^2_{j,l}\right) < \infty$, par le théorème central limite, on a la convergence en loi de $\sqrt{n}\,(LG_n''(\theta) - I_\theta)$, ce qui implique

$$LG_n''(\theta) - I_\theta = O_P(n^{-1/2}).$$

Comme $\frac{1}{\sqrt{n}}[\log(P(\theta))]''$ converge presque sûrement vers 0, on obtient pour tout θ

$$\sqrt{n}\,(A_\theta - I_\theta) = O_P(1) \quad . \tag{15}$$

Le second résultat qui va nous servir est celui démontré dans la partie précédente qui est

$$\theta^\star = \hat{\theta} + O_P(n^{-1/2}). \tag{16}$$

En effectuant un développement de Taylor de $A_{\theta^\star}$ autour de $A_{\hat{\theta}}$ à l'ordre 1, puisque $\theta^\star$ et $\hat{\theta}$ sont proches quand n est grand, on peut écrire

$$\sqrt{n}(A_{\theta^\star} - I_{\hat{\theta}}) = \sqrt{n}(A_{\hat{\theta}} - I_{\hat{\theta}}) + \sqrt{n}(\theta^\star - \hat{\theta})A'_{\hat{\theta}} + o(\sqrt{n}(\theta^\star - \hat{\theta})^2).$$

On remarque que $A'_{\hat{\theta}} = L'''_n(\hat{\theta})$ et on rappelle que la condition que cette quantité soit bornée en n est demandée pour obtenir l'ordre en $1/n$ dans l'approximation de Laplace. Le résultat (15) reste vrai pour $\theta = \hat{\theta}$. Il en vient que le premier terme est de l'ordre de $O_P(1)$. Par (16), on a que le second terme est de l'ordre de $O_P(1)$ et que le dernier (en $o_P(n^{1/2})$) est négligeable. On obtient alors

$$\sqrt{n}(A_{\theta^\star} - I_{\hat{\theta}}) = O_P(1).$$

Références

Akaike, H. (1973). Information theory as an extension of the maximum likelihood principle. In B.N. Petrov et F. Csaki (Eds.), *Second International Symposium on Information Theory*, pp. 267–281. Akademiai Kiado, Budapest.

Azaïs, J.-M., E. Gassiat, et C. Mercadier (2003). Asymptotic distribution and power of the likelihood ratio test for mixtures: bounded and unbounded case. Technical report, Preprint, Orsay.

Biernacki, C., G. Celeux, et G. Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on pattern analysis and machine intelligence 22*(7), 719–725.

Birgé, L. et P. Massart (2001). Gaussian model selection. *J. Eur. Math. Soc.* **3**, 203–268.

Bozdogan, H. (1987). Model selection and Akaike's information criterion (AIC): the general theory and its analytical extensions. *Psychometrika* **52**, 345–370.

Burnham, K.P. et D. Anderson (2002). *Model selection and multi-model inference*. Springer-Verlag New York.

Castellan, G. (1999). Modified Akaike's criterion for histogram density estimation. Technical Report 61, Université Paris XI.

Celeux, G. et G. Soromenho (1996). An entropy criterion for assessing the number of clusters in a mixture model. *Classification Journal* **13**, 195–212.

Chernoff, H. et E. Lander (1995). Asymptotic distribution of the likelihood ratio test that a mixture of two binomials is a single binomial. *Journal of Statistical Planning and Inference 43*(1), 19–40.

Csiszar, I. et P.C. Shields (2000). The consistency of the bic markov order estimator. *Ann. Statist. 28*(6), 1601–1619.

Dacunha-Castelle, D. et E. Gassiat (1997). The estimation of the order of a mixture model. *Bernoulli Journal of Mathematical Statistics and Probability 3*(3), 279–299.

Dudley, R.M. et D. Haughton (1997). Information criteria for multiple data sets and restricted parameters. *Statistica Sinica*, 265–284.

Dudley, R.M. et D. Haughton (2002). Asymptotic normality with small relative errors of posterior probabilities of half-spaces. *Ann. Statist. 30*(5), 1311–1344.

Fraley, C. et A. E. Raftery (1998). How many clusters? which clustering method? answer via model-based cluster analysis. *The Computer Journal* **41**, 578–588.

Hannan, E.J. (1980). The estimation of the order of an arma process. *Ann. Statist.* **8**, 1071–1081.

Hartigan (1985). A failure of likelihood ratio asymptotics for normal mixtures. In *Proceedings of the Berkeley Conference in Honor of Jerzy Neyman and Jack Kiefer*.

Haughton, D.M.A. (1988). On the choice of a model to fit data from an exponentiel family.

Ann. Statist. 16(1), 342–355.

Hoeting, J.A., D. Madigan, A.E. Raftery, et C.T. Volinsky (1999). Bayesian model averaging: A tutorial. *Statist. Science 14*(4), 382–417.

Hurvich, C.M. et C.L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika* **76**, 297–307.

Kass, R.E. et L.A. Wasserman (1996). The selection of prior distributions by formal rules. *J. Amer. Statist. Assoc.* **90**, 1343–1370.

Kass, R. E. et A. E. Raftery (1995). Bayes factors. *J. Amer. Statist. Assoc.* **90**, 773–795.

Kass, R. E. et L. Wasserman (1995). A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion. *J. Amer. Statist. Assoc.* **90(2)**, 928–934.

Keribin, C. (1998). Consistent estimate of the order of mixture models. *Comptes Rendus de l'Academie des Sciences* **326**, 243–248.

Lebarbier, E. (2005). Detecting multiple change-points in the mean of gaussian process by model selection. *Signal processing* **85**, 717–736.

Madigan, D. et A.E. Raftery (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *J. Amer. Statist. Assoc.* **89**, 1535–1546.

Mallows, C.L. (1974). Some comments on Cp. *Technometrics* **15**, 661–675.

McLachlan, G. et D. Peel (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics.

Poskitt, D. S. (1987). Precision, complexity and bayesian model determination. *J. R. Statist. Soc. B 49*(2), 199–208.

Raftery, A. E. (1995). Bayesian model selection in social research (with discussion). *Sociological Methodology*, 111–196.

Reschenhoffer, E. (1996). Prediction with vague prior knowledge. *Comm. Statist.* **25**, 601–608.

Ripley, B.D. (1995). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Rissanen, J (1978). Modelling by the shortest data description. *Automatica* **14**, 465–471.

Rissanen, J. (1987). Stochastic complexity. *J. R. Statist. Soc. B* **49**, 223–239.

Ronchetti (1985). Robust model selection in regression. *Statis. Probab. Lett.* **3**, 21–23.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.* **6**, 461–464.

Shi, P. et C.L. Tsai (1998). A note on the unification of the Akaike information criterion. *J. R. Statist. Soc. B* **60**, 551–558.

Sugiura (1978). Further analysis of the data by akaike's information criterion and the finite corrections. *Comm. Statist.* **A7**, 13–26.

Tierney, L. et J.B. Kadane (1986). Accurate approximations for posterior moments and marginal densities. *J. Amer. Statist. Assoc.* **81**, 33–59.

Van Der Vaart, A.W. (1998). *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics.

Annexe B

Etude de l'hétérogénéité des plaques dans les expériences de biopuces

Depuis la fin des années 90, la technologie des puces à ADN est devenue un outil incontournable de la génomique fonctionnelle. Cette technologie permet de mesurer l'expression de dizaines de milliers de gènes simultanément, à travers la quantification des ARN messagers présents dans les tissus et correspondant à ces gènes. Cette grande quantité de mesures disponibles représente une information précieuse pour le biologiste. Elle représente aussi un défi pour le statisticien, qui doit proposer des outils adaptés pour le traitement des données de grande dimension. Les données de biopuces ont ainsi amené les statisticiens à reformuler certains problèmes statistiques classiques, comme les méthodes de correction pour tests multiples ou la sélection de variables en classification supervisée.

Proposer de nouvelles méthodes d'analyse statistique nécessite souvent une bonne connaissance des données à traiter. Cette connaissance se révèle indispensable dans le cas des données de biopuces : le statisticien doit prendre en compte les limites de la technologie, ainsi que la nature même des données dont il dispose pour élaborer une stratégie d'analyse adaptée. Ce travail d'appropriation des données de biopuces a donné lieu à deux publications (Mary-Huard *et al.* (2004), Mary-Huard *et al.* (2006)). Nous présentons ici la première, qui porte sur les problèmes de normalisation.

Le processus expérimental partant de l'extraction de l'ARN d'une cellule pour aboutir à l'obtention d'une mesure d'expression d'un gène est complexe. Ce processus se décompose en plusieurs étapes techniques, chacune de ces étapes étant susceptible d'entacher la mesure finale de bruit expérimental. De ce fait, les données de biopuces doivent être normalisées avant d'être analysées. La normalisation a pour objet l'identification des différentes sources de variabilité technique, puis la réduction de cette variabilité afin d'obtenir une mesure biologique fiable de l'expression d'un gène. L'article qui suit étudie l'effet de l'étape de dépôt (spotting effect), étape où les séquences d'ADN sont déposées sur la puce. Le dépôt est réalisé par un robot automatisé, qui récupère les séquences dans des plaques de séquences. Chaque plaque fait l'objet d'une préparation spécifique préalable, qui peut entraîner des différences entre séquences provenant de plaques différentes. L'article présente les conséquences de l'hétérogénéité des plaques sur les niveaux d'expression mesurés, et précise les conditions nécessaires pour pouvoir corriger la variabilité introduite par ce facteur.

BMC Bioinformatics

BioMed Central

Methodology article

Open Access

Spotting effect in microarray experiments

Tristan Mary-Huard*[1], Jean-Jacques Daudin[1], Stéphane Robin[1], Frédérique Bitton[2], Eric Cabannes[3] and Pierre Hilson[2,4]

Address: [1]Institut National Agronomique Paris-Grignon, 16 rue Claude Bernard, 75231 Paris, France, [2]UMR Génomique Végétale, INRA-CNRS-Université d'Evry, CP 5708, F-91057 Evry, France, [3]Laboratoire d'Immunologie Virale, Institut Pasteur, 28 rue du Docteur Roux, 75724 Paris, France and [4]Department of Plant Systems Biology, Flanders Interuniversity Institute for Biotechnology (VIB), Ghent University, Technologiepark 927, B-9052 Gent, Belgium

Email: Tristan Mary-Huard* - maryhuar@inapg.fr; Jean-Jacques Daudin - daudin@inapg.fr; Stéphane Robin - robin@inapg.fr; Frédérique Bitton - bitton@urgv.fr; Eric Cabannes - cabannes@pasteur.fr; Pierre Hilson - pihil@gengenp.rug.ac.be

* Corresponding author

Published: 19 May 2004

BMC Bioinformatics 2004, **5**:63

Received: 09 January 2004
Accepted: 19 May 2004

This article is available from: http://www.biomedcentral.com/1471-2105/5/63

Abstract

Background: Microarray data must be normalized because they suffer from multiple biases. We have identified a source of spatial experimental variability that significantly affects data obtained with Cy3/Cy5 spotted glass arrays. It yields a periodic pattern altering both signal (Cy3/Cy5 ratio) and intensity across the array.

Results: Using the variogram, a geostatistical tool, we characterized the observed variability, called here the spotting effect because it most probably arises during steps in the array printing procedure.

Conclusions: The spotting effect is not appropriately corrected by current normalization methods, even by those addressing spatial variability. Importantly, the spotting effect may alter differential and clustering analysis.

Background

Microarray technology is probably the most successful in the area of functional genomics. Biologists use it to analyze gene expression at the genome scale by comparing the levels of messenger RNAs present in matched biological samples, for example grown under contrasted conditions or with different genetic configurations. Microarray data can be used for differential analysis, to identify genes whose expression strongly depends on the nature of the samples, as well as for clustering analysis, to identify coexpressed genes. Microarray data show a high level of variability. Some of this variability is relevant because it corresponds to the differential expression of genes. But, unfortunately, a large portion results from undesirable biases introduced during the many technical steps of the experimental procedure. Several sources of experimental noise have already been addressed, such as dye or fluorophore, fluorescence level or print-tips and statistical methods have been proposed to normalize data according to the related effects ([1,2]).

In this paper, we describe an experimental bias and use statistical methods to investigate the distribution of the signal across the microarray area. We use the variogram to analyze spatial dissimilarities between spots on the slide. Although spatial signal distribution across the slide has already been studied ([3,4]), the bias we report here has never before been explicitly characterized. We also present two experiments that give clues about the nature of the spotting effect, and finally we investigate the possibility to

correct the effect and the efficiency of usual normalization procedures to do it.

Analyzed datasets are produced by using glass arrays and the two-color labeling strategy by which two conditions are compared directly. In these experiments, mRNA samples are collected from case and reference cells. The two corresponding cDNA samples are synthesized and labeled with either the Cy3 (green) or Cy5 (red) fluorophore and are mixed and hybridized simultaneously to a single array. For each DNA feature (representing a gene) printed and bound on the array, the fluorescence emitted by the hybridized labeled cDNA is measured in the Cy3 and Cy5 channels. Both fluorescence measurements are compared to define the relative gene expression in case versus reference cells. We designate these two values G and R and define the signal and intensity associated with a given gene as follows:

• the signal associated with a gene is the logarithm of the ratio R/G. This quantity is used to identify differentially expressed genes;

• the intensity is defined as the logarithm of the product $R \times G$ (or $log(R \times G)/2$).

The goal of normalization is to correct the signal for experimental bias. Most existing normalization procedures do not specifically correct for potential spatial effects. The few that do only consider sources of variation that are restricted locally. For instance, the print-tip effect acts as a block effect, where the blocks are defined by the cluster of spots printed on the array with the same print-tip ([1]). The goal of this study is to determine whether normalization that corrects for additional spatial effects is necessary or whether current normalization models are sufficient.

Results

Our first test case is a self-hybridized microarray printed with *Arabidopsis thaliana* PCR-amplified cDNA sequences. In a self-hybridization microarray experiment, no gene should appear to be differentially expressed (R/G = 1) and the observed variability results from experimental effects. Also, no particular spatial pattern of intensity or signal is expected unless the DNA features are arranged on the array with respect to their type (for instance, transcribed versus intergenic regions as in [5]), which is not the case for this *Arabidopsis* microarray. Self-hybridization experiments have already proved to be efficient in detecting systematic biases ([1]). The *Arabidopsis* slide self-hybridization results show two spatial effects (Figure 1). First, the overall signal decreases from left to right. Second, the signal is arranged in a periodic pattern: sets of high signal vertical lines alternate with sets of low signal vertical lines. For practical reasons, rows in blocks are represented as vertical lines in Figures 1 and 5. Intensity values are structured according to a similar periodic pattern (data not shown).

To gain further insight in the data structure, we plotted the signal and intensity variograms for the *Arabidopsis* slide (figures 2 and 3), respectively. In both, the abscissa is the distance d between two points expressed as the number of rows that separates them, and the ordinate is the value of the variogram for a given distance calculated with formula (2) given in section Methods. When the observed value for each spot is independent from the value of all other spots at any given distance d, the variogram is a straight horizontal line. When a correlation exists only between closely neighboring spots (for example, because of local distortions of the slide), the curve will start at low $V(d)$ values for small distances, and reach a higher horizontal plateau because the correlation disappears as d increases. Figures 2 and 3 highlight a different pattern: a given spot in a given row is similar to spots that are N = 10, 20, 30... rows apart. Spots that are 10 rows apart are particularly similar, which can be explained because spots in row N+10 are the duplicates of spots in row N within each block. Yet, this duplication does not explain the resemblance between spots distant by multiples of 10 rows higher than 10. This is probably due to the fact that all similar DNA features N × 10 rows apart from each other are printed in the same step (see Methods). The same pattern can also be observed in columns (data not shown). For convenience only, the observed periodic bias will be called the "spotting effect".

Detection of the spotting effect in multiple microarray datasets

To determine whether the spotting effect is particular to the presented *Arabidopsis* slide or a common experimental bias in spotted microarrays, we studied twelve slides provided by three European or Canadian Laboratories and five slides available in the public Stanford MicroArray Database http://genome-www5.stanford.edu/MicroArray/SMD/. Results are described for one slide from the first set (Tor270, see Table 1) and two from the second (Lieb3727, [5] and Zhu473, [6]) as representative samples of our analysis. All slides were used for transcription profile comparisons or clustering analysis, except for the *Arabidopsis* slide that was a self-hybridization experiment and the Lieb3727 slide that was a chromatin immunoprecipitation microarray experiment (ChIP-chip). All were printed with PCR amplicons using two different robots (Microgrid II and ChipWriters) and according to various spotting designs: 16, 32 or 48 print-tip heads, duplicate prints in rows or columns, side by side, or far apart. Table 1 provides a summarized description of the microarrays for which results are presented below.

BMC Bioinformatics 2004, **5** http://www.biomedcentral.com/1471-2105/5/63

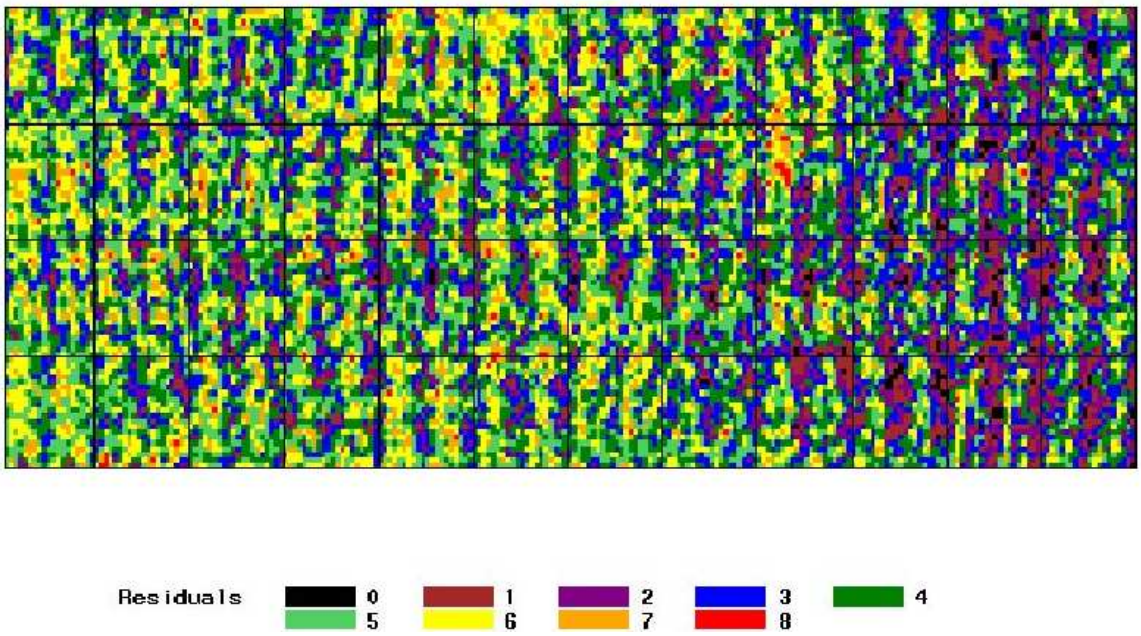


Figure 1
Spatial distribution of the signal for the self-hybridized Arabidopsis slide Each pixel represents the uncorrected log-ratio of the median Cy5 (635 nm) and Cy3 (532 nm) channel fluorescence measurements, associated to a printed DNA feature. Background is not represented. The picture is not a re-plot of the original image captured during the scanning process. Labels correspond to the 9-quantiles of the signal distribution.

The variograms in Figure 4 show that the raw signal is periodically structured according to rows in the different microarray datasets (25, 24 and 21 rows per block), stressing the prevalence of spotting effect biases and the need for correction through normalization procedures. We also computed the Type III Mean Squares (MSq) values associated with the print-tips, spotting and intensity-per-block effects, according to the following model:

$$Y_s = \mu + \alpha_r^{sp} + \beta_{bl} + \beta_{bl} \times Int_s + \varepsilon_s \qquad (1)$$

$$\mathbb{V}(\varepsilon_s) = \sigma^2$$

where Y_s is the signal measured at spot s, and α_r^{sp}, β_{bl} and Int_s the mean spotting row, block and intensity effects, respectively. Spotting row effect means that only one parameter is estimated for all the rows spotted at the same time. For example, in the self-hybridized *Arabidopsis* slide dataset the row effect is the same for the rows 1,11, ..., 231.

Type III MSq values measure variability due to a factor after adjustment for the other factors in the model and point out which effects need specific correction (see [7]). In summary, for the 18 slides analyzed, MSq values of the spotting effect were 10-fold lower to 4-fold higher than intensity-per-block MSq values and were equal to 10-fold higher than the print-tips effect MSq values. This result confirms that the spotting effect is present in many experiments and at least as important as other documented sources of variability.

Nature of the spotting effect

The spotting effect could be explained in different ways. The amount of material deposited on or bound to the slide and the shape of DNA spots can be affected by multiple factors, such as the time during which the print-tips are soaked in the DNA source microtiter plates, the time during which the print-tips touch the slides, the speed at which the print-tips move, the concentration and the salinity of the DNA solutions, the temperature and the relative humidity of the arrayer printing cabinet, and the physicochemical characteristics of the print-tips and of the

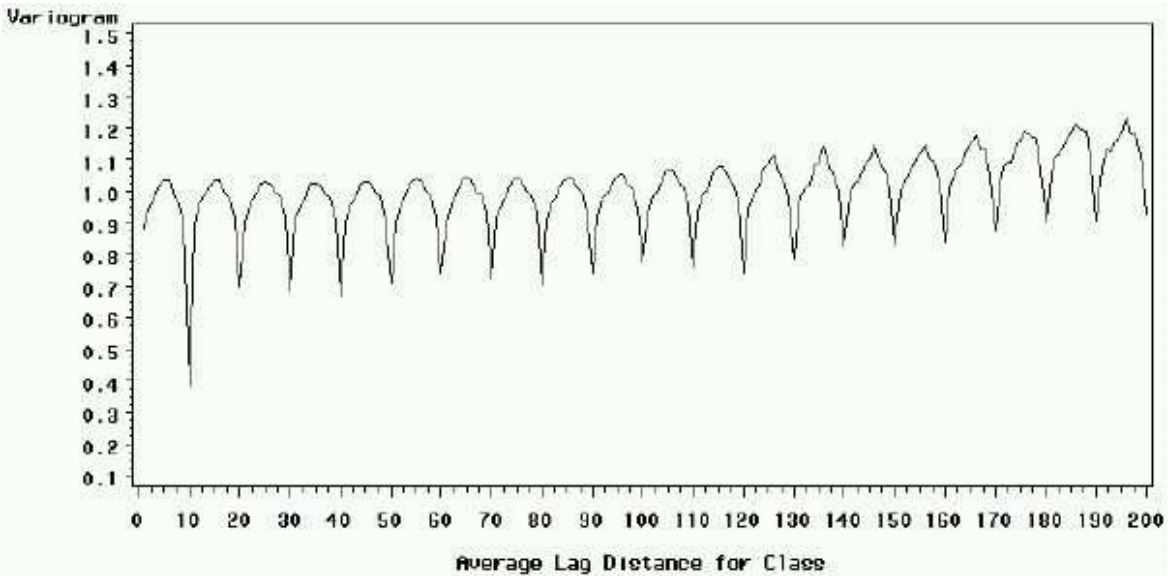


Figure 2
Variogram of the signal by row for the Arabidopsis slide, before normalization

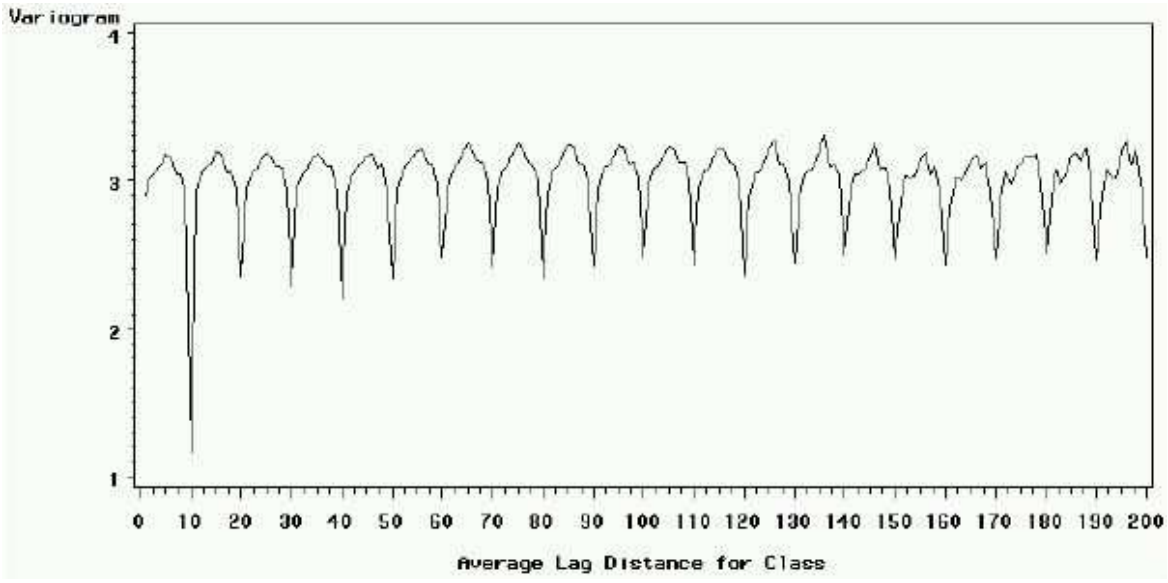


Figure 3
Variogram of the intensity ($log(R \times G)$) by row for the Arabidopsis slide

BMC Bioinformatics 2004, **5** http://www.biomedcentral.com/1471-2105/5/63

Table 1: Characteristics of the datasets studied

Lab. or Database	Robot	Print-tips Heads	Cols × Rows/Block	Nber of Rep.
Arabidopsis (URGV)	(*Micro*grid II)	48	21 × 20	row *N* and *N* + 10
UHN Toronto (Tor270)	ChipWriters	32	24 × 25	col *N* and *N* + 1
SMD Zhu473	?	16	24 × 24	?
SMD Lieb3727	?	32	24 × 21	None

glass surface. With most spotting robots, the printing of high density arrays containing thousands of features lasts for hours and subtle changes in spotting conditions may, therefore, alter all these factors. For example, DNA solutions may evaporate over time. In that regard, the spotting effect may be related to the "time-of-print" effect reported in [8].

Alternatively, the spotting effect may reflect DNA source plate variations because all DNA features printed simultaneously originate from the same plate. To test this hypothesis, we analyzed results from two microarray experiments for which the plate effect was controlled. In the first experiment, slides were printed with a unique 384-well-plate containing human cDNA amplicons and hybridized to targets prepared from RNA isolated from primary $CD4^+$ T cell. In the second experiment, slides were printed with oligonucleotides of 70 bases, synthesized according to a different chemistry, and hybridized to cDNA from various developmental stages of *Plasmodium falciparum*. The oligonucleotides have the same length and are resuspended in solutions showing a narrow concentration range. In both cases, the spotting effect was greatly reduced (data not shown). This observation suggests that the plate effect is a major component of the spotting effect.

Spotting effect and normalization

Many authors have already pointed out the dangers of systematic normalization procedures. It is important to determine the conditions in which the correction of the spotting effect is appropriate and to verify that no biological effect can be confounded with the experimental biases corrected by the normalization. Taking into consideration the association between spotting and plate effects described above, three cases can be considered.

1. Probes are arranged according to their biological characteristics, for instance intergenic regions separated from transcription units, or genes expected to be differentially expressed grouped together in particular plates. In this case, it is impossible to distinguish between a significant plate effect due to coexpression of genes belonging to the same class, or due to technical artifacts.

2. Probes are arranged according to their chromosomal order. Such structure may lead to significant differences between plates if genes with similar expression profiles are spatially clustered in the genome (silent neighboring genes in heterochromatic regions, for example). Such spatial clustering has been recently observed in several organisms ([9,10]) and may affect many others.

3. Probes are randomly distributed among plates. Most human array experiments verify this hypothesis. The results presented in Section 4 prove that this configuration does not cause the spotting effect to disappear.

A normalization procedure to correct the effect is advisable only in the last case because, in the first two, regardless of the importance of the spotting bias, the correction would unavoidably alter the biological information contained in the data. Thus, the effect can considerably affect the conclusions of experiments corresponding to the first two cases. In particular, results of experiments studying gene similarity or the relationship between relative chromosomal position and coexpression could be essentially twisted, as also pointed out by Balazsi *et al*. in [11].

Assuming that the experiment of interest corresponds to the third case, one has to investigate whether a specific normalization for the spotting effect is needed or if standard normalization is sufficient. We present here the consequences of the normalization procedure proposed by Yang *et al*. in [1], one of the most widely used methods in the microarray community, on the self-hybridized *Arabidopsis* cDNA array data. Only results obtained with background-corrected signals and the global loess normalization procedure are presented. The analysis performed on background-uncorrected data, or with the print-tip loess normalization procedure gave similar results.

The *Arabidopsis* slide signal normalized with the reference procedure (residual) still shows a periodic pattern as illustrated in Figure 5 and calculated with the variogram (Figure 6). This observation indicates that the bias introduced by the spotting effect is not fully corrected. According to

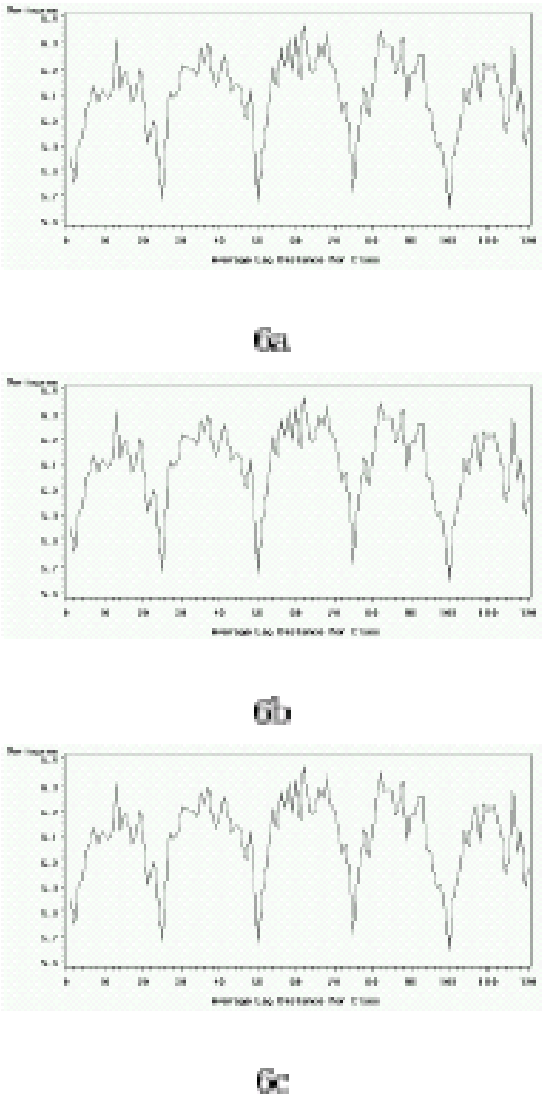


Figure 4
Variogram for the raw signal by row (A) Tor270 slide; (B) Zhu473 slide; (C) Lieb3727 slide. No normalization is performed.

our experience, other usual normalization procedures do not perform an efficient correction of the spotting effect.

Because of the strong pattern observed in rows, the spotting effect may be treated as row effect per block or as a global row effect across the slide. Preliminary results suggest that such models adequately correct the periodic spatial bias described in various microarray datasets. Row models are advantageous because they rely exclusively on the geometrical information that is embedded in the data and that is mandatory according to the "Minimum Information About a Microarray Experiment" (MIAME) guidelines. In contrast, plate origin information is very rarely available and is difficult to integrate into statistical analysis considering that successive technical steps usually take place in multi-titer plates of different formats (e.g. 96-, 384- and 1536-well plates) during spotted microarray DNA production.

Discussion

We have observed that transcription profiling datasets obtained with spotted glass microarrays and the two-color labeling strategy show a bias that leads to periodic patterns according to rows and columns of the array grid. These patterns affect the entire area and alter both signal and intensity. We propose that such patterns result from artifacts introduced during the DNA feature preparation into microtiter plates or the slide printing procedure because features spotted together yield the most similar signals.

Color swaps are now routinely included in microarray experimental designs to correct for labeling biases. They consist in repeated hybridizations in which the case and reference samples are labeled at least once with each of the Cy3 and Cy5 fluorophores. Preliminary analyses indicate that the spotting effect is reduced when raw data from opposite color swap hybridizations are combined (unpublished results). This observation is consistent with the fact that the spotting effect depends on the position of the spots on the slide and that the relative spot position remains the same from slide to slide in most setups. We suggest that the reduction of the spotting effect resulting from the combination of raw opposite color datasets may constitute additional justification for the inclusion of color swaps in microarray experiments.

We have shown that the variogram is an efficient tool to display spatial correlations between spots. Furthermore, it is possible to test the null hypothesis that no spatial correlation exists (for instance the Moran test described in [12]). Such tests could be performed together with the variogram analysis as part of the data normalization procedure to investigate the significance of observed spatial biases and to evaluate the need and efficiency of different correction methods.

Conclusions

We have proved that the spotting effect is statistically significant, is as important as other effects that are

BMC Bioinformatics 2004, **5** http://www.biomedcentral.com/1471-2105/5/63

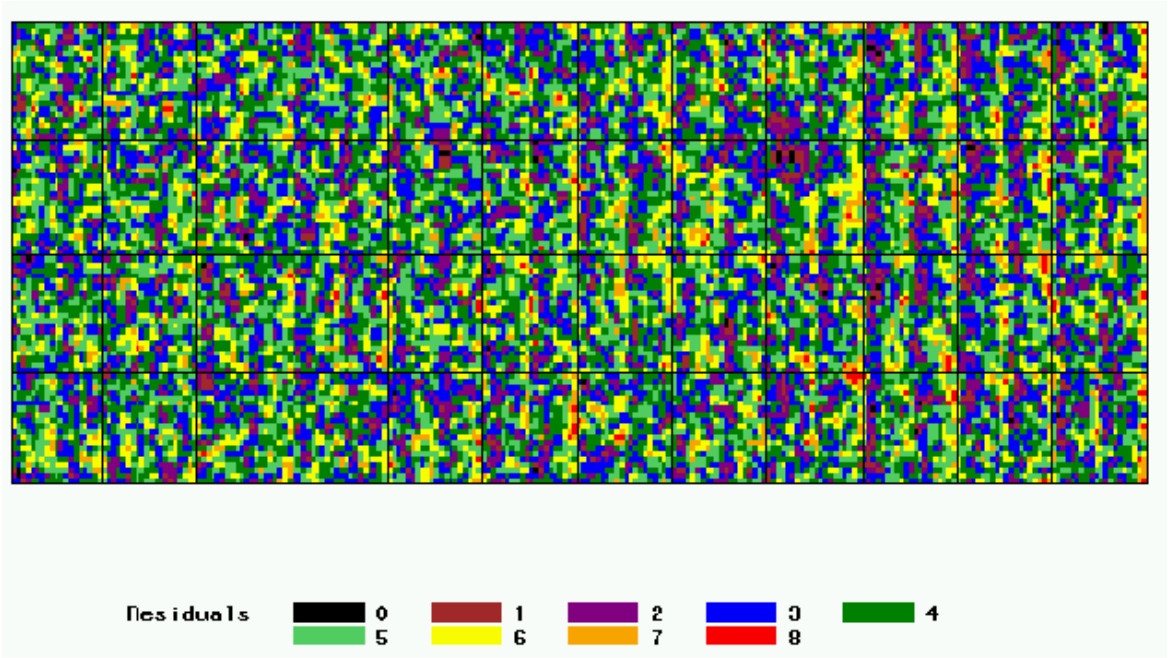


Figure 5
Distribution of the residuals (i.e. corrected signal) after reference normalization, for the Arabidopsis slide
Labels correspond to the 9-quantiles of the residuals distribution.

commonly corrected, and should be taken into account in normalization procedures. This effect is expected to increase the number of false positives and negatives in classical microarray studies. In differential analysis, some rows or columns may contain artificially high or low numbers of "differentially expressed genes". In clustering analysis, genes may be associated because of a similarity caused by the spotting effect.

Methods

Description of the first test slide

The *Arabidopsis* glass slide (Corniong GAP II) studied in section Results was hybridized with Cy3 and Cy5 labeled cDNA samples, both prepared from the same mRNA extracted from *Arabidopsis* flower buds (self-hybridization). The microarray structure consists of 4 × 12 blocks, each with 20 rows and 21 columns. cDNA sequences were spotted in duplicate, i.e. rows N and N+10 (for N = 1 to 10) in the same block were printed with the same series of amplicons. The robot printing head consisted of 48 (4 × 12) print-tips, each defining a block. During a single printing step, the robot printed 48 spots on a slide (each corresponding to a different DNA feature) distant by 20 rows in one direction and 21 columns in the other (the distance between print-tips); then, within a fraction of a second, the robot arm moved laterally and printed the duplicate spots 10 rows away before moving to the next slide. Once all slides were spotted with a given set of 48 duplicated amplicons, the robot washed all print-tips simultaneously, loaded them with the next set of 48 amplicons, and resumed printing. Each set was printed on all slides in approximatively 2 min and the entire procedure lasted 16 h for the 10000 duplicated cDNAs.

Variogram

The structure of the spatial distribution of the signal on a slide can be studied with a geostatistical tool called a variogram ([13,14]). In geostatistics, the variogram has been used to detect departure of stationarity in the data. In the microarray data analysis context, it represents a useful exploratory tool to study spatial correlations due to systematic biases. A variogram (also called semi-variogram) is defined (2), and estimated (3), for a distance d and a variable Y, as follows:

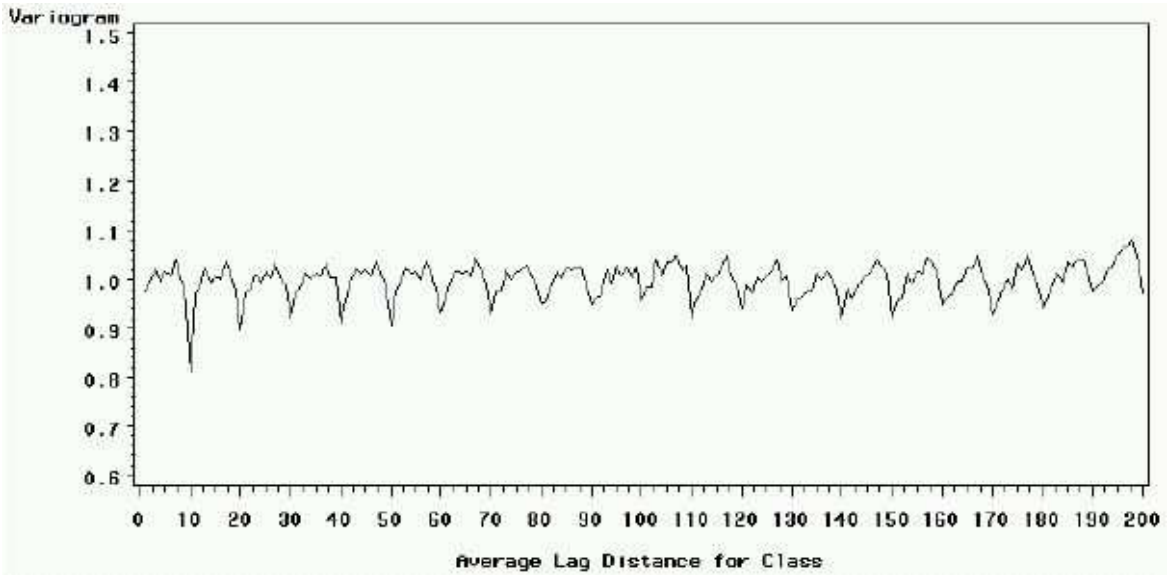


Figure 6
Variogram of the residuals (i.e. corrected signal) after reference normalization, for the Arabidopsis slide

$$V(d) = \mathbb{E}\left\{\frac{1}{2}[Y(t) - Y(t-d)]^2\right\} \qquad (2)$$

$$\hat{V}(d) = \frac{1}{2|N(d)|}\sum_{N(d)}\left(Y(s_i) - Y(S_i)\right)^2 \qquad (3)$$

where $N(d)$ is the set of all possible pairs of spots (s_i, s_j) with a distance d between one another, and with $|N(d)|$ the cardinal of $N(d)$. As implied by expression (2), $V(d)$ decreases when the number of similar points separated by the distance d increases.

Authors' contributions

TMH did the major part of the data analysis, and created all tables and figures. JJD and SR proposed the statistical methods and supervised their application in collaboration with TMH. FB completed all wet laboratory Arabidopsis microarray work supervised by PH, and they provided the initial dataset for testing. Tor270 experiment was performed by EC. EC and PH conducted the interpretation of the statistical results in the light of hardware experimental settings. The manuscript was written by PH and TMH. All authors read and approved the final manuscript.

Acknowledgements

E.C. was supported by a fellowship from Agence Nationale de Recherches sur le SIDA

References

1. Yang Y, Dudoit S, Luu P, Speed T: **Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation.** *Nuclear Acids Res* 2002, **30(4):**e15.
2. Quackenbush J: **Microarray data normalization and transformation.** *Nature Genet* 2002, **32:**496-501.
3. Schuschhardt J, Beule D, Malik A, Wolski E, Eickhoff H, Lehrach H, Herzel H: **Normalization strategies for cDNA microarrays.** *Nucleic Acids Res* 2000, **28:**e47.
4. Workman C, Jensen L, Jarmer H, Berka R, Gautier L, Nielser H, Saxild H, Nielsen C, Brunak S, Knudsen S: **A new non-linear normalization method for reducing variability in DNA microarray experiments.** *Genome Biol* 2002, **3(9):**1-16.
5. Lieb J, Liu X, Botstein D, Brown P: **Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association.** *Nature Genet* 2001, **28(4):**327-34.
6. Zhu G, Spellman P, Volpe T, Brown P, Botstein D, Davis T, Futcher B: **Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth.** *Nature* 2000, **406(6791):**90-4.
7. Searle S: *Linear Models* New York: John Wiley & Sons, Inc; 1971.
8. Ball C, Chen Y, Panavally S, Sherlock G, Speed T, Spellman P, Yang Y: **Section 7: An introduction to microarray bioinformatics.** In *DNA Microarrays: A Molecular Cloning Manual* Cold Spring Harbor Press; 2003.
9. Cohen B, Mitra R, Hughes J, Church G: **A computational analysis of whole-genome expression data reveals chromosomal domains of gene espression.** *Nature Genet* 2000, **26:**183-186.
10. Spellman P, Rubin G: **Evidence for large domains of similarly expressed genes in the *Drosophila* genome.** *J Biol* 2002, **1:**1-5.

11. Balazsi G, Kay K, Barabasi A, Oltvai Z: **Spurious spatial periodicity of co-expression in microarray data due to printing design.** *Nucleic Acids Res* 2003, **31:**4425-4433.
12. Banerjee S, Carlin B, Gelfand A: **Hierarchical Modeling and Analysis for Spatial Data.** *Monographs on Statistics and Applied Probability* Chapman and Hall/CRC Press; 2004.
13. Jowett G: **The Accuracy of systematic sampling from conveyor belts.** *Applied Statistics* 1952, **1:**50-59.
14. Cressie A: **Statistics for spatial data.** *Wiley series in probability* Wiley; 1997.

Annexe C

Pertinence statistique du mélange d'échantillons dans les expériences de biopuces

La quantification de l'expression des gènes est basée sur l'extraction de l'ARN messager des tissus d'intérêt. Suivant l'organisme considéré, il arrive que le nombre de cellules disponibles pour l'extraction soit très faible. Dans les études sur le SIDA par exemple, l'ARNm est extrait des cellules sanguines, qui ne peuvent pas être prélevées de manière massive sur un individu. L'une des possibilités pour disposer d'une quantité suffisante de matériel génétique est alors de "pooler" les ARNm, c'est-à-dire de mélanger les ARNm de différents échantillons. Cette étape de mélange a lieu très tôt dans l'expérience, bien avant l'hybridation des puces à ADN, et donc bien avant l'étape d'analyse des données. La conséquence de ce mélange peut être de taille sur l'analyse : un biais introduit lors de cette étape peut modifier les mesures d'expression des gènes, et fausser l'analyse statistique.

L'étude des biais potentiels que peut introduire l'étape de mélange a fait l'objet d'un projet d'action intégrée franco-italien du programme européen Galileo, intitulé *Plan d'expérience pour l'analyse de l'expression des gènes*, en collaboration avec le Dipartimento di Statistica G. Parenti, de l'université de Florence. Contrairement aux deux études précédemment présentées sur le critère BIC et l'effet du spotting, cette étude n'est pas encore achevée et n'a pas fait l'objet d'une publication. C'est pourquoi nous ne reproduisons ici que le résumé de cette étude.

Biases induced by pooling RNA samples in microarray experiments

Tristan Mary-Huard*[1] , Simona Toti[2,3] , Jean-Jacques Daudin[1] , Michela Baccini[2,3] , Annibale Biggeri[2,3] and Avner Bar-Hen[1]

[1] UMR INA-PG/INRA/ENGREF MIA 518, 16 rue Claude Bernard 75231 Paris Cedex 5 (France)
[2]Department of Statistics 'G. Parenti', viale Morgagni 59, 50100, Florence (Italy)
[3]CSPO Biostatistics Unit, via San Salvi, Florence (Italy)

Email: Tristan Mary-Huard*- maryhuar@inapg.fr; Simona Toti - toti@ds.unifi.it; Jean-Jacques Daudin - daudin@inapg.fr; Michela Baccini - baccini@ds.unifi.it; Annibale Biggeri - abiggeri@ds.unifi.it; Avner Bar-Hen - avner@inapg.fr;

*Corresponding author

Abstract

Background: If there is insufficient RNA from the tissues under investigation from one organism, then it is common practice to pool RNA. Other reasons for pooling include provision of adequate quantities of a standard reference that can be maintained consistent over time and to reduce variation. Although amplification is an alternative to pooling, the resultant output can be very non-linear. However, one problem with pooling is the lost of information that can lead to inaccurate results.

Methods: In this article we decompose the bias introduced by pooling in two parts: specification bias and pool bias. The two parts are studied from a theoretical as well as a practical point of view, assuming additive and multiplicative models on raw intensity.

Results: Bias of the pooling can come from two sources: (i) specification bias: the log of the mean is not the mean of the log. The sign of this bias is known thanks to Jensen's inequality. Analytic expressions of this bias are given under multiplicative and additive models for raw signal. (ii) pool-bias: difference between the mean of intensity and the raw pooled intensity. We use public domain data of Kendziorski to study this question.

Conclusions: We derived theoretical approximations of the specification biases under the multiplicative and additive error model, which are both a function of the biological variability and the mean expression. We then applied these results and found that the empirical behavior of the bias matches the theoretical expression obtained. We show that gain of pooling is generally overestimated in comparative studies.

Résumé : La thèse se place dans le contexte de l'apprentissage statistique. On s'intéresse au problème de la classification supervisée des données de grande dimension. Dans un premier temps, nous introduisons les principes de la théorie de Vapnik et de la sélection de modèles, et présentons les grandes familles de méthodes de sélection de variables. Dans un deuxième temps, nous nous proposons de nouveaux outils de sélection de modèles et de réduction de la dimension. Au chapitre 3, nous proposons un estimateur du biais entre le risque conditionnel et le risque empirique d'une règle de classification. Cet estimateur est ensuite utilisé pour l'élaboration d'un critère pénalisé de sélection de modèles appelé Swapping. La pénalité est alors basée sur les seuls points de l'échantillon d'entraînement pour lesquels un changement de label entraîne un changement de prédiction. Les performances pratiques du critère Swapping sont illustrées par l'application à l'algorithme kNN pour le choix du nombre de voisins. Au chapitre 4, nous présentons un critère pénalisé pour la sélection de variables en classification supervisée. La qualité du critère obtenue est garantie par une inégalité oracle. Ce résultat théorique est ensuite utilisée pour justifier la phase d'élagage de l'algorithme CART en tant que méthode intégrée de sélection de variables. Le dernier chapitre est consacré à l'agrégation de variables. Nous présentons une stratégie générale d'agrégation de variables dédiée à l'algorithme de classification choisi par l'expérimentateur. Cette méthode d'agrégation est décrite pour les algorithmes kNN et CART. Nous présentons une application de cette méthode de réduction de la dimension à des données fonctionnelles ainsi qu'à des données de biopuces.
Mots-clés : Apprentissage statistique, sélection de modèles, sélection de variables, agrégation de variables, critère pénalisé, inégalité oracle.

Abstract: This thesis takes place within the framework of statistical learning. We study the supervised classification problem for large dimension data. The first part of the document presents the state of the art regarding model selection in the Vapnik theory framework and different variable selection approaches in statistical learning. The second part presents some new tools for model selection and dimension reduction. In Chapter 3, we provide an estimator of the bias between the conditional and the training error of a classification rule. This estimator is then used to derive a penalized criterion called Swapping. The penalty is based on the points for which a change of label induces a change of prediction. An application to the choice of k in the kNN algorithm is presented. In Chapter 4, we propose a penalized criterion for variable selection in supervised classification. This criterion provides a theoretical justification of the pruning step of the CART algorithm as an embedded variable selection method. The last chapter is dedicated to a general strategy to aggregate variables in view of classification. The reduction dimension method is adapted to the classification algorithm. Applications to functional and microarray data are provided.
Keywords: Statistical Learning, model selection, variable selection, variable aggregation, penalized criterion, oracle inequality.

AMS Classification: 62H30, 68T10, 62G05, 62P10, 62P99