Apprentissage supervisé de données déséquilibrées par forêt aléatoire

Julien Thomas

▶ **To cite this version:**

Julien Thomas. Apprentissage supervisé de données déséquilibrées par forêt aléatoire. Autre [cs.OH]. Université Lumière - Lyon II, 2009. Français. NNT : 2009LYO22004 . tel-01540283

HAL Id: tel-01540283
https://theses.hal.science/tel-01540283

Submitted on 16 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Lumière Lyon 2
École Doctorale Informatique et Information pour la Société

THÈSE

pour obtenir le grade de
Docteur
en
Informatique

présentée et soutenue publiquement par

Julien Thomas

le 12 Février 2009

Apprentissage supervisé de données déséquilibrées par forêt aléatoire

préparée au sein du laboratoire ERIC et de la société Fenics
sous la direction de Nicolas Nicoloyannis et Djamel Abdelkader Zighed

COMPOSITION DU JURY

Mme. Sylvie PHILIPP-FOLIGUET	Rapporteur	(ENSEA)
M. Gilles VENTURINI	Rapporteur	(Université de Tours)
Mme. Nicole VINCENT	Examinateur	(Université Paris 5)
M. Behzad SHARIAT	Examinateur	(Université Lyon 1)
M. Djamel Abdelkader ZIGHED	Directeur de thèse	(Université Lyon 2)
M. Pierre-Emmanuel JOUVE	Invité	(Société Fenics)

Remerciements

Merci Nicolas.

Résumé

La problématique des jeux de données déséquilibrées en apprentissage supervisé est apparue relativement récemment, dès lors que le data mining est devenu une technologie amplement utilisée dans l'industrie. L'aide au diagnostic médical, la détection de fraudes, de phénomènes anormaux, ou encore d'éléments spécifiques sur des images satellites, sont autant d'exemples d'applications industrielles basées sur l'apprentissage supervisé de données déséquilibrées. Le but de nos travaux est d'adapter différents éléments de l'apprentissage supervisé à cette problématique. Nous cherchons également à répondre aux exigences spécifiques de performances souvent liées aux problèmes de données déséquilibrées, comme un taux de rappel élevé pour la classe minoritaire. Ce besoin se retrouve dans notre application principale, la mise au point d'un logiciel d'aide à la détection des cancers du sein, où la priorité est de trouver un maximum de lésions avant de chercher à diminuer le nombre de fausses alarmes.

Pour cela, nous proposons de nouvelles méthodes modifiant trois différentes étapes d'un processus d'apprentissage. Tout d'abord au niveau de l'échantillonnage, nous proposons lors de l'utilisation d'un bagging, de remplacer le bootstrap classique par un échantillonnage dirigé. Nos techniques FUNSS et LARSS utilisent des propriétés de voisinage pour la sélection des individus. Ensuite au niveau de l'espace de représentation, notre contribution consiste en une méthode de construction de variables adaptées aux jeux de données déséquilibrées. Cette méthode, l'algorithme FuFeFa, est basée sur la découverte de règles d'association prédictives. Enfin, lors de l'étape d'agrégation des classifieurs de base d'un bagging, nous proposons d'optimiser le vote à la majorité en le pondérant. Pour ce faire nous avons mis en place une nouvelle mesure quantitative d'évaluation des performances d'un modèle, PRAGMA, qui permet la prise en considération de besoins spécifiques de l'utilisateur vis-à-vis des taux de rappel et de précision de chaque classe.

Table des matières

1 Introduction **17**

1.1 Contexte industriel 17

1.2 Problématique et positionnement 21

1.3 Organisation du mémoire 22

2 Eléments fondamentaux **25**

2.1 Fouille de données 25

2.2 Fouille de Données Complexes : particularités 28

2.3 Apprentissage 32

2.3.1 Notations et Définitions 32

2.3.2 Apprentissage supervisé 34

2.4 Conclusion 36

3 Méthodes d'apprentissage de classes **39**

3.1 Introduction 39

3.2 Analyse discriminante 41

3.3 SVM : Support Vector Machines 43

3.4 Régression logistique 47

3.5 Arbres de décision 51

3.6 Agrégation de classifieurs 54

3.6.1 Bagging 55
3.6.2 Boosting 56
3.6.3 Forêt aléatoire 56

4 Evaluation de modèles 59
4.1 Erreur en apprentissage et généralisation 59
4.2 Déséquilibre et notion de symétrie 61
4.3 Indicateurs 64
4.3.1 Taxonomie des critères d'évaluation 64
4.3.2 Mesures de performance 65

5 Espace des individus 73
5.1 Introduction 73
5.2 Etat de l'art 74
5.2.1 Sur et sous-échantillonnage 74
5.2.2 Echantillonnage et ensemble 75
5.3 FUNSS : une approche guidée par le coût 76
5.3.1 Principe 77
5.3.2 Expérimentations 78
5.4 LARSS : vers une adaptation localisée 84
5.4.1 Adaptations 84
5.4.2 Expérimentations 87
5.5 Conclusion 91

6 Espace de représentation 95
6.1 Introduction 95
6.2 Etat de l'art 96
6.2.1 Taxonomie 96

6.2.2 Méthodes par analyse topologique des arbres 97
6.2.3 Méthodes par analyse et exploration des données . . . 98
6.2.4 Méthodes basée sur l'utilisation des connaissances du domaine 99
6.2.5 Méthodes multi-stratégiques 99
6.3 FuFeFa : Fuzzy Feature Factory 100
6.3.1 Règles d'association 100
6.3.2 Relâchement des bornes des items 102
6.3.3 Création des variables et utilisation en forêt aléatoire . 103
6.4 Expérimentations 104
6.5 Conclusion 107

7 Mesure de performance 111

7.1 Introduction 111
7.2 PRAGMA : Precision and RecAll rates Guided Model Assessment 112
7.3 Exemple d'optimisation des forêts aléatoires 115
7.3.1 Stratégie de vote 115
7.3.2 Recherche automatique 116
7.4 Expérimentations 118
7.4.1 Jeux de données équilibrés 118
7.4.2 Jeux de données déséquilibrées 119
7.5 Conclusion 123

8 Conclusion 125

8.1 Bilan 125
8.2 Perspectives 128

9 Annexe : Données utilisées 131

9.1 Jeu de données SATIMAGE 131
9.2 Jeu de données LETTERS 135
9.3 Jeu de données AUTOS 136
9.4 Jeu de données HYPOTHYROID 139

Table des figures

1.1 Flux général de mise au point et d'application du système de détection des cancers du sein. 19

1.2 Exemples de visuels à classifier, de gauche à droite : (A) Masses : Ruptures d'architecture (cancer), Opacité (cancer) et Image construite (non cancer) - (B) Foyers de microcalcifications : 3 différents cancers, Macrocalcifications (non cancer) et Poussières (non cancer) 20

1.3 Affichage du logiciel Smart Look : on observe une masse détourée en trait continu, celle-ci est microcalcifiée, ce qui explique la détection d'un foyer malin de microcalcifications (trait pointillé et centre d'intérêt en X) 20

2.1 Technologies et modèle général d'ECD [ZR02]. 26

2.2 Processus général d'ECD [ZR02]. 27

2.3 Objectif du processus d'apprentissage 34

2.4 Phase d'apprentissage 35

2.5 Validation sur échantillon test 36

2.6 Principe de la généralisation 37

3.1 Illustration des voisinages utilisés pour différentes approches à base d'instance 41

3.2 Visualisation de l'analyse discriminante du jeu de données Iris 44

3.3 Illustration du compromis à trouver entre sous ajustement et sur ajustement 46

3.4 Illustration de l'hyperplan à marge maximal dans un cas linéairement séparable 47

3.5 Exemple d'arbre de décision construit avec C4.5 sur le jeu de données Iris [HB99] à trois classes. 51

3.6 L'entropie de Shannon et l'indice de Gini dans un cas à deux classes. 54

4.1 Principe d'une validation croisée à 5 subdivisions. 60

4.2 L'effet d'un manque "absolu" de données. 62

5.1 Principe de construction d'une forêt aléatoire utilisant l'échantillonnage FUNSS. 79

5.2 Exemple de sélection d'un individu de la classe majoritaire lors d'un échantillonnage FUNSS. 80

5.3 Visualisation dans le plan formé par les 2 premiers axes ACP de 3 échantillons issus du jeu Satimage : (A) FUNSS mode "éloigné", (B) bootstrap, (C) FUNSS mode "proche". 81

5.4 Evolution du rappel estimé lors de la construction d'une forêt aléatoire utilisant l'échantillonnage FUNSS80. Résultats issus d'une 5-CrossValidation sur le jeu Satimage. 83

5.5 Principe de construction d'une forêt aléatoire utilisant l'échantillonnage LARSS. 86

5.6 Exemple de sélection d'un individu de la classe majoritaire lors d'un échantillonnage LARSS. 88

5.7 Visualisation dans le plan formé par les 2 premiers axes ACP de 2 échantillons issus du jeu Satimage : (A) Bootstrap, (B) FUNSS localisé (50^e arbre). 89

6.1 Réponse de satisfaction à un item I et distribution des individus selon une variable quantitative X_j 102

6.2 Fonction $g_I(\omega_i)$ associée à un item I et distribution des individus selon une variable quantitative X_j 103

7.1 2 modèles, A($r = 0.3; p = 0.8$) et B($r = 0.8; p = 0.3$), sont localement évalués à l'aide de 3 différentes $f(r_i, p_i)$: Cas I (symétrique) A et B sont équivalents ; Cas II (rappel préféré) B est meilleur ; Cas III (précision préférée) A est meilleur. . . 114

7.2 Exemples de distribution de votes pour : (A) le jeu de données Letters [HB99] (les objets sans vote pour la modalité 'S' ne sont pas représentés) ; (B) le jeu de données Mammo (voir 7.4)(les objets ayant moins de 2 votes pour la modalité 'Cancer' ne sont pas représentés). 117

7.3 Résultats détaillés pour Letters : (A) Forêt aléatoire classique ; (B) Forêt aléatoire optimisée, le rappel et la précision "s'organisent" selon les préférences de l'utilisateur. 121

7.4 Résultats détaillés pour le jeu de données Mammo. 124

Liste des tableaux

3.1 Répartition de différentes classes de méthodes de construction de modèles de prédiction en fonction des espaces de prédiction 40

4.1 Matrice de confusion issue de la prédiction d'un problème à n classes 66

5.1 Composition des jeux de données déséquilibrées Satimage et MammoClusters. 80

5.2 Résultats (en %) de différents algorithmes obtenus en validation croisée (5 subdivisions) sur le jeu Satimage.(R : Rappel ; P : Précision) 84

5.3 Résultats (en %) de différents algorithmes obtenus en validation croisée (5 subdivisions) sur le jeu Satimage.(R : Rappel ; P : Précision) 90

5.4 Résultats (en %) de différents algorithmes obtenus en validation croisée (5 subdivisions) sur le jeu MammoClusters.(R : Rappel ; P : Précision) 91

6.1 Classification des méthodes de construction de variables. . . . 97

6.2 Paramètres utilisés 106

6.3 Résultats (en %) de différentes forêts aléatoires obtenus en validation croisée (5 subdivisions) sur le jeu Satimage.(R : Rappel ; P : Précision) 107

6.4 Résultats (en %) de différentes forêts aléatoires obtenus en validation croisée (5 subdivisions) sur le jeu MammoMasses.(R : Rappel ; P : Précision) 108

7.1 Résultats pour Autos : le taux de correction global de la forêt aléatoire classique (RF Class.) est de 78.0%, celui de la forêt aléatoire optimisée (RF Opt.) est de 78.5%, soit une amélioration +0.5pts. Légende : R Rappel ; P Précision ; MP Moyenne des classes Prioritaires ; MA Moyenne des Autres classes ; MG Moyenne Globale. 119

7.2 Résultats pour Letters : le taux de correction global de la forêt aléatoire classique (RF Class.) est de 88.1%, celui de la forêt aléatoire optimisée (RF Opt.) est de 87.2%, soit une perte -0.9pts. 120

7.3 Composition des jeux de données déséquilibrées utilisés. 120

7.4 Résultats (en %) obtenus en 10-CrossValidation pour Hypothyroïd, Satimage et Mammo. 121

Chapitre 1

Introduction

1.1 Contexte industriel

Les travaux que nous présentons dans cette thèse ont été réalisés dans le cadre d'une Convention Industrielle de Formation par la Recherche (CIFRE)[1]. Les conventions CIFRE associent une entreprise, un laboratoire et un doctorant dans le cadre d'un projet de recherche industriel. La présente thèse a été réalisée au sein du laboratoire ERIC (Equipe de Recherche en Ingénierie des Connaissances) de l'université Lumière Lyon 2, et de la société Fenics, PME lyonnaise créée en 2003. Fenics est une société d'édition de logiciels dédiés à l'aide au diagnostic du cancer du sein.

Selon l'Institut National du Cancer (INCA)[2], le cancer du sein est le cancer féminin le plus fréquent. Le moyen le plus efficace pour détecter un cancer du sein le plus tôt possible est l'examen mammographique. Le défi actuel est de proposer des systèmes d'aide au diagnostic (*Computer Aided Diagnosis*, CAD) performants capables d'assister les radiologues afin de rendre leurs diagnostics plus rapides et plus sûrs. Pour répondre à ce défi la société pro-

[1]Nous remercions l'ANRT et le Ministère de la Recherche et de l'Industrie pour leur participation au financement de ces travaux

[2]http ://www.e-cancer.fr/

pose le logiciel Smart Look qui permet de détecter les zones cancéreuses sur les mammographies, le système argumentant son choix auprès du radiologue en proposant également une caractérisation des anomalies. Ainsi les radiologues sont assistés durant leurs lectures, en attirant leur attention vers des anomalies difficiles à détecter dans le but de réduire le nombre de cancers non détectés. Pour ce faire deux domaines technologiques sont utilisés : l'imagerie, et plus précisément la segmentation d'images, et le data mining. Le flux général du processus est présenté en figure 1.1.

Il existe deux grands types de tumeurs de cancers du sein : les masses et les foyers de microcalcifications, chacun faisant l'objet d'un flux de traitement distinct. Les masses, regroupant opacités, densités et ruptures d'architecture, apparaissent globalement sous formes de taches blanches de diamètre allant de moins d'un centimètre à un décimètre. Celles-ci sont à distinguer des "images construites", visuellement très proches d'une masse, mais dûes à la superposition de tissus lors de la prise de la radiographie, ou encore d'artefacts créés lors de la segmentation. Les foyers de microcalcifications sont des regroupements de petites calcifications apparaissant sous forme de points blancs de moins d'un millimètre. Tous les foyers (ou clusters) de microcalcifications ne sont pas cancéreux. Ils peuvent naître naturellement dans les tissus mammaires. Il faut donc distinguer la forme bénigne de la forme maligne des foyers, mais aussi les séparer d'éventuelles poussières ou artefacts. Différents exemples de visuels sont donnés en figure 1.2.

C'est dans le cadre de la mise au point des modèles de prédiction pour le logiciel Smart Look (voir figure 1.3) que se sont inscrits nos travaux. Pour réaliser cette application, des outils internes ont été développés au sein de la société Fenics, comme notamment une plate-forme de data mining (principalement dédiée à l'apprentissage supervisé) appelée LearnIt. Tous les tests présentés dans le présent mémoire ont été effectués à l'aide de cette plate-forme.

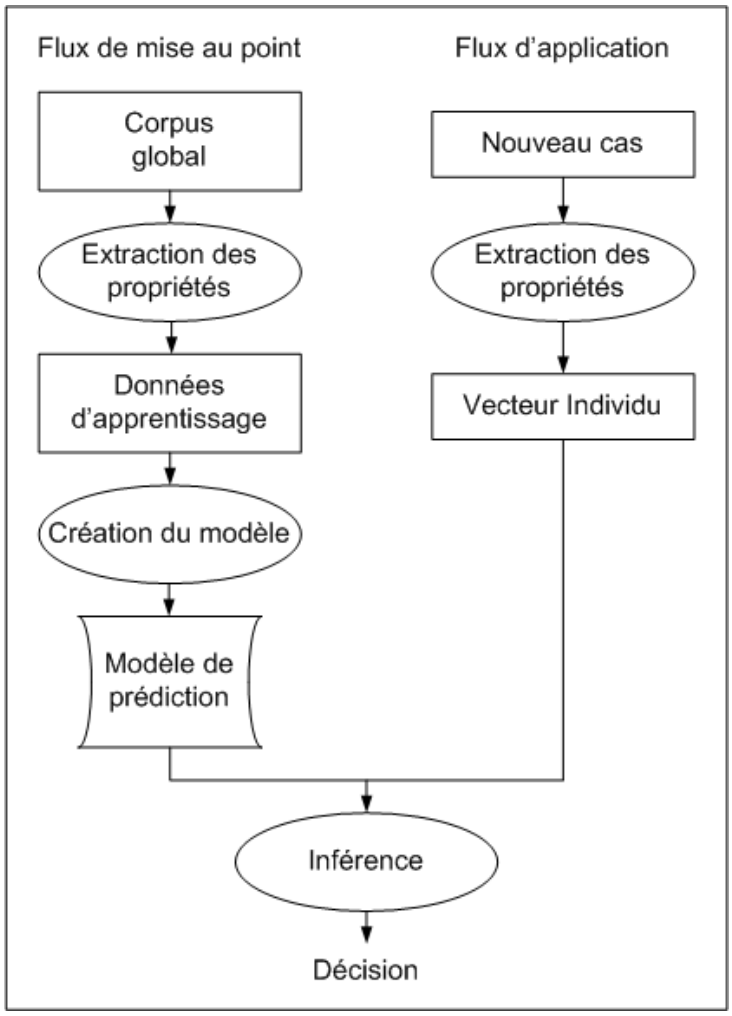


FIG. 1.1 – Flux général de mise au point et d'application du système de détection des cancers du sein.

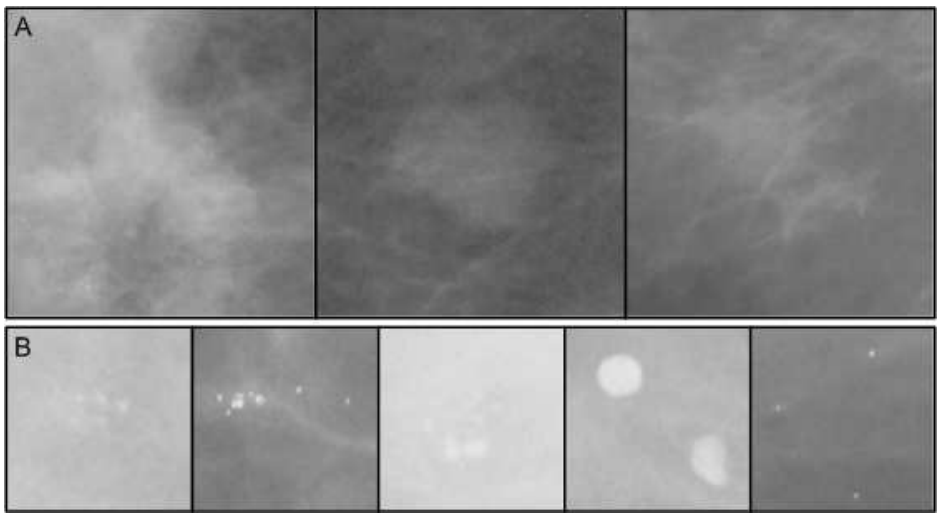


FIG. 1.2 – Exemples de visuels à classifier, de gauche à droite : (A) Masses : Ruptures d'architecture (cancer), Opacité (cancer) et Image construite (non cancer) - (B) Foyers de microcalcifications : 3 différents cancers, Macrocalcifications (non cancer) et Poussières (non cancer)

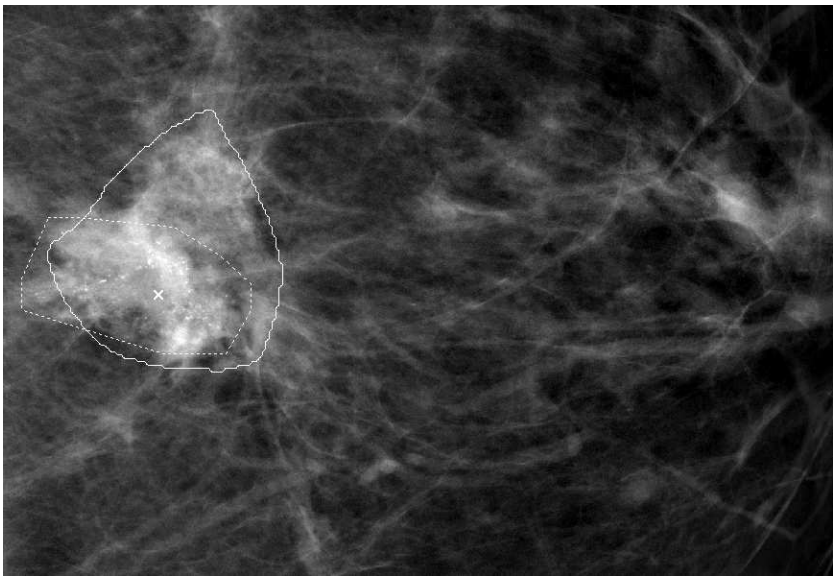


FIG. 1.3 – Affichage du logiciel Smart Look : on observe une masse détourée en trait continu, celle-ci est microcalcifiée, ce qui explique la détection d'un foyer malin de microcalcifications (trait pointillé et centre d'intérêt en X)

1.2 Problématique et positionnement

On retrouve à travers cet objectif applicatif de mise au point de détection de zones cancéreuses, différents problèmes théoriques du domaine du data ming (voir 2.1) et plus précisemment de l'apprentissage supervisé (voir 2.3.2) :

Vis-à-vis de la notion de donnée complexe, pour laquelle nous reviendrons en 2.2, on peut identifier différents points-clé, dont :

- La dimensionnalité : Le fait que les individus présentent un très grand nombre de descripteurs peut perturber l'apprentissage. C'est le cas par exemple dans notre application industrielle où les données à traiter sont des zones extraites d'images, sur lesquelles des milliers de propriétés mathématiques sont calculées, engendrant ainsi des jeux de grande dimensionnalité.
- La subjectivité : L'analyse des données complexes est sujette à des interprétations qui relève de la subjectivité. Cela crée une étendue sémantique sur le concept et rend de ce fait la modélisation plus difficile. Par exemple, lorsqu'on cherche à détecter toutes zones susceptibles d'être cancéreuses, la labélisation des données d'apprentissage se fait avec le concours d'experts sénologues, il reste une part de subjectivité, les avis des différents experts pouvant diverger. De plus le détourage des zones à étudier est automatique, et ne correspond parfois que grossièrement aux contours "idéaux" décrits par l'expert.

Vis-à-vis du déséquilibre des jeux de données, sujet que nous détaillerons par la suite, on retrouve deux éléments principaux :

- Non équi-répartition des classes : Un jeu de données déséquilibrées est un jeu où les effectifs des différentes classes sont très différents. Cela peut perturber l'apprentissage car la plupart des métriques usuelles du processus sont bâties sur une hypothèse d'équirépartition, comme les mesures d'entropie ou les critères de décision. Dans notre contexte industriel les jeux de données sont constitués entre autres à partir de campagnes de dépistage du cancer du sein, où les cas cancéreux ne représentent que $0,03\%$ ou $0,04\%$ de la totalité des examens réalisés.

Sur chaque cliché mammographique plusieurs centaines de zones sont extraites, pour la plupart du temps ne trouver qu'une seule localisation cancéreuse si celle-ci existe. Ces éléments rendent la modalité "cancer" de nos jeux d'apprentissage très minoritaire malgré les étapes de sélection d'examens. Il nous faudra pour cela opter pour des stratégies d'apprentissage adaptées aux jeux de données déséquilibrées.

– Non équivalence des conséquences des décisions : L'utilisation d'un modèle issu d'un apprentissage supervisé permet d'obtenir des prédictions. Celles-ci engendrent différentes actions aux conséquences plus ou moins importantes et coûteuses. Chaque prédiction, et plus précisément chaque type d'erreurs de prédiction, est en cela rattachable à un coût. Si ce dernier diffère fortement selon le type d'erreur, il est important d'en tenir compte directement dans la construction du modèle de prédiction. Dans notre exemple applicatif, il est bien plus grave de ne pas détecter un cancer chez une patiente, que de présenter au radiologue une zone bénigne qui engendra au pire un complément d'examen.

Le problème de complexité, et plus particulièrement la haute dimensionnalité, comme le déséquilibre impactent fortement les algorithmes d'apprentissage. Notre objectif est de proposer des stratégies capables de contenir ces deux difficultés pour améliorer les performances. Ce travail se positionne donc dans le cadre de l'apprentissage supervisé et nous allons, dans ce contexte, examiner les travaux existants et tenter d'apporter des réponses adaptées tant sur le plan théorique que pratique.

1.3 Organisation du mémoire

Après avoir présenter le contexte industriel de cette thèse et présenter notre problématique, la suite de ce mémoire s'organise en deux parties : une première, composée des chapitres 2 à 4, a pour vocation d'appronfondir notre problématique et notre positionnement à travers un état de l'art du domaine ; la deuxième, les chapitres 5,6 et 7 présentera nos contributions, avant une

conclusion. Le Chapitre 9 est une annexe donnant les descriptions des jeux de données publics utilisés.

PARTIE 1, Etat de l'art.

- Le Chapitre 2 présente différents *éléments fondamentaux* du domaine du datamining et pose les notations que nous utiliserons en apprentissage supervisé.
- Le Chapitre 3 est dédié aux *méthodes d'apprentissage de classes*, illustrées par la présentation de quatres d'entre elles parmi les plus utilisées, ainsi que les possibilités d'aggrégation de classifieurs.
- Enfin le Chapitre 4 aborde la dernière étape du processus d'apprentissage supervisé, à savoir *l'évaluation des modèles*, et son lien avec la notion de symétrie.

PARTIE 2, Contributions.

- Le Chapitre 5 est relatif aux techniques d'*échantillonnage*, et présente les deux nouvelles méthodes que nous proposons : **FUNSS** et **LARSS**. Celles-ci tentent de contrer les difficultés du déséquilibre sans modifier les rapports d'effectifs entre les différentes classes tout en intégrant les besoins spécifiques de l'utilisateur.
- Le Chapitre 6 se rapporte à *l'espace de représentation* et la construction de variables. Nous y proposons la méthodes **FuFeFa**, basée sur la découverte de règles d'association prédictives et spécifique aux jeux de données déséquilibrées.
- Nous terminons par le Chapitre 7, présentant notre *mesure de performance* **PRAGMA**. Cette dernière, dont la spécificité est la prise en considération intuitive des besoins de l'utilisateur vis-à-vis des taux de rappel et précision sur chaque classe, permet la mise en place d'optimisations.

Chapitre 2

Eléments fondamentaux

2.1 Fouille de données

La fouille de données ou *data mining* est l'art d'extraire des connaissances à partir de données. Ces dernières peuvent être stockées dans des entrepôts, des bases de données distribuées ou sur Internet. Le *data mining* ne se limite pas au traitement de données structurées, il offre également des moyens pour aborder des corpus en langue naturelle, on parle alors de *text mining*, des images, des sons ou de la video, on parle alors généralement de *multimedia mining*. L'Extraction de connaissances à partir de données (ou ECD) est le processus général qui utilise la fouille de données. L'ECD peut être vu comme une ingénieurie pour extraire de la connaissance utile à partir de grandes bases de données.

L'ECD est un processus itératif qui met en oeuvre un ensemble de techniques provenant des bases de données, de la statistique, de l'intelligence artificielle, ou encore des interfaces homme-machine [ZR02] (voir figure 2.1). L'ECD vise à transformer des données en connaissances pouvant s'exprimer sous la forme d'un concept général qui enrichit le champ sémantique de l'usager par rapport à une question qui le préoccupe. Les réponses apportées peuvent s'exprimer par un rapport, un graphique, ou encore un modèle

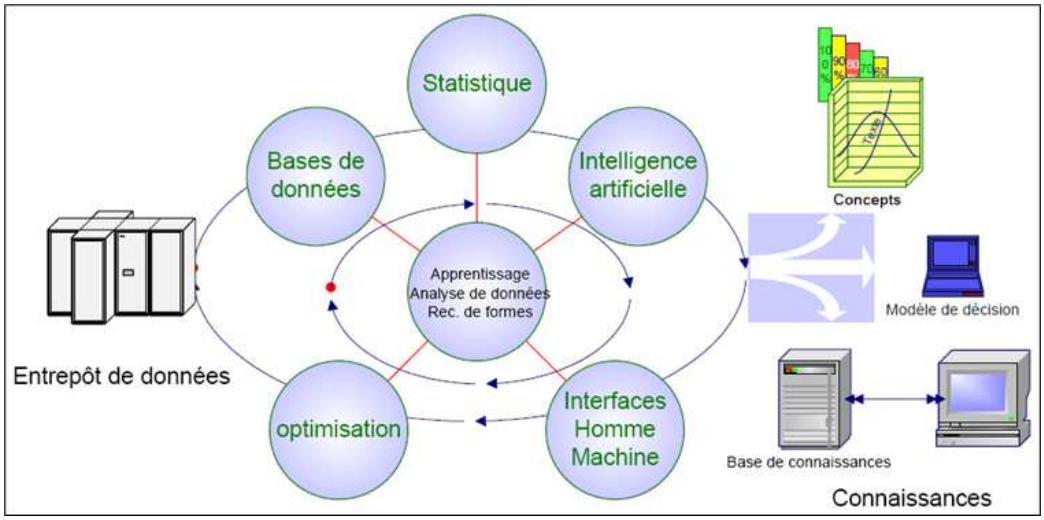


FIG. 2.1 – Technologies et modèle général d'ECD [ZR02].

mathématique ou logique pour la prise de décision. Ces modèles explicites peuvent également alimenter des systèmes à base de connaissances ou des systèmes expert.

D'autres définitions du *data mining* ou plus généralement de l'ECD existe, mais ces dernières sont souvent plus restrictives. On peut par exemple citer celle donnée en 1996 par Fayyad [FPSS96] : *"l'extraction de connaissances à partir de données est un processus non trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données"*.

Le processus général de l'ECD est anthropocentré, les connaissances extraites doivent être les plus intelligibles possibles à l'utilisateur. Elles doivent être validées, mises en forme et agencées. Le cycle de ce processus est donné par la figure 2.2. On y retrouve une analyse des données en quatre étapes principales successives :

1. L'étape d'acquisition des données : on y réalise la collecte initiale des données, en produit une description, étudie éventuellement quelques hypothèses à l'aide de visualisations et vérifie le niveau de qualité des données en vue d'éventuelles sélections ou nettoyage des données.

2. L'étape de préparation des données : elle consiste en la construction d'une table de données pour permettre la fouille en elle-même. Cette mise en forme nécessite entre autres des transformations et la construction d'attribut.

3. L'étape de fouille de données : Il s'agit ici du *data mining* au sens strict. On y retrouve selon les cas, la description, la structuration des données ou encore la construction de modèles explicatifs.

4. L'étape de gestion des connaissance : cette étape permet l'évaluation des résultats obtenus et leur mise en forme pour produire des connaissances intelligibles pour l'utilisateur final.

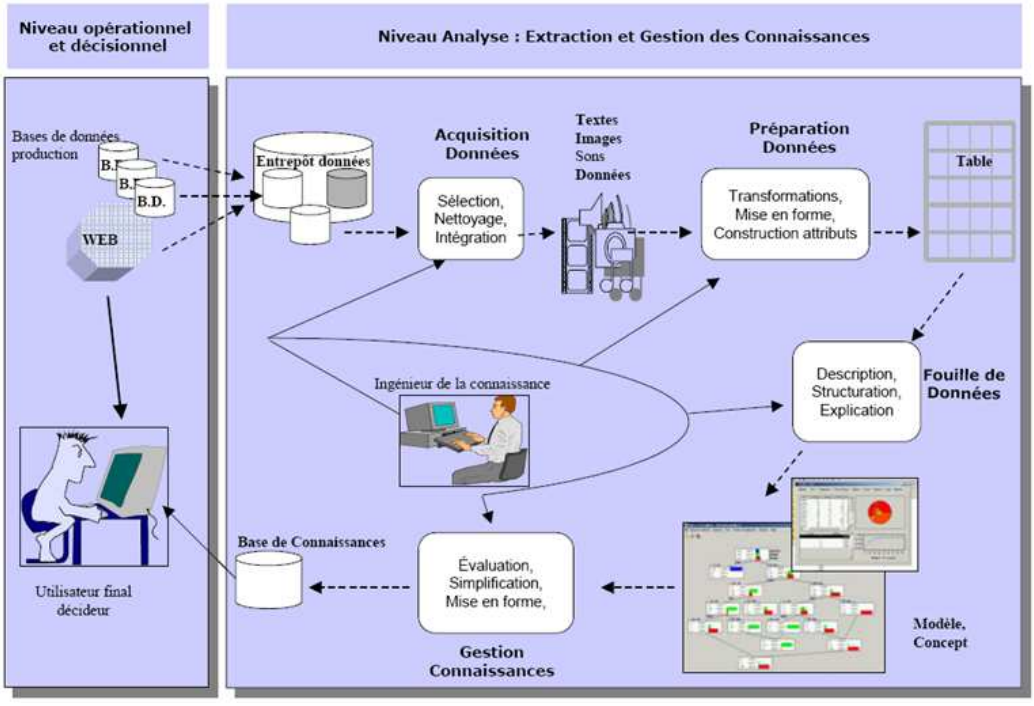


FIG. 2.2 – Processus général d'ECD [ZR02].

2.2 Fouille de Données Complexes : particularités

La place des données complexes (image, vidéo, texte non structuré ou combinaison de ces médias) n'a cessé de croître, pour être aujourd'hui le principal véhicule d'information. La problématique d'une diffusion massive et de qualité est quasi-réglée grâce aux technologies de l'entreposage massif des données et aux réseaux à haut débit. Nous disposons de volumineuses bases de données complexes dont la croissance est exponentielle mais dont la valorisation reste encore très faible.

Le défi qu'il faut relever est de tirer profit, dans tous les sens du terme, de ces données : recherche d'information, extraction de connaissances, création de valeurs économiques etc. La Fouille de données complexe tente de répondre à ce besoin de valorisation. Elle propose, pour cela, de définir un cadre méthodologique et des outils pour structurer les données complexes, les analyser en vue d'extraire des connaissances ou des informations non accessibles par des moyens classiques. Les particularités des données complexes peuvent être résumées comme suit :

- Volumineuses : Plusieurs téraoctets. Par exemple le Dossier Médical Personnalisé (DMP) peut atteindre plusieurs dizaines de giga-octets par patient.
- Distribuées : Le DMP par exemple, peut être stocké dans différentes bases de données distribuées selon les services médicaux où le patient a séjourné.
- Hétérogènes : Les données peuvent être de différentes natures. Dans le cas du DMP par exemple, on aura des images radiologiques, des comptes-rendus textuels, des tableaux de chiffres de mesures biologiques, des courbes d'électrocardiogramme, des enregistrements vidéo d'échographie, etc.
- Evolutives : Différents enregistrements avec des contenus différents. Par exemple, le DMP contient divers examens réalisés à des instants différents et qui ne portent pas nécessairement sur les mêmes tests médicaux.

– Non structurées : Elles ne sont généralement pas modélisées dans le cadre d'un schéma de base de données relationnelles mais stockées quasiment en vrac et dans le meilleur des cas dans des formats ad hoc comme le format MPGE7 pour les données multimédia ou le format DICOM pour le DMP.

Nous sommes alors confrontés à deux défis

Défis scientifiques dans la FDC Les défis scientifiques que soulèvent ces particularités des données complexes sont multiples. Parmi ceux que l'on peut assez facilement identifier, on peut lister :

– La grande dimensionnalité. Les attributs issus des images, des textes et des autres modalités peuvent atteindre plusieurs centaines, voire des milliers de variables. Comment évaluer la pertinence de cet espace de description par rapport aux taches que l'on souhaite effectuer comme l'apprentissage supervisé ou la classification ? comment réduire l'espace si tant est que cela soit possible et/ou souhaitable ?

– Absence de structure mathématique. Généralement les codages effectués sur les données complexes sont faits de sorte que les tableaux qui en résultent sont assimilés à des points plongés dans des espaces multidimensionnels et, dans le meilleur des cas, dans des espaces vectoriels. Dans ce cadre, l'outillage mathématique, issu notamment de l'algèbre linéaire et de la programmation mathématique, permet de traiter ces données. Or ce codage n'est pas toujours possible notamment en présence de données hétérogènes qualitatives (état civil, localisation géographique etc.) et quantitatives ou de données non structurées comme des graphes orientés ou de données symboliques (courbes, intervalles, distribution etc.). Comment alors analyser ces ensembles de données ? Quel codage faut-il adopter ? comment construire des indices de proximités qui sont des outils indispensables pour les tâches d'apprentissage notamment non supervisé ? Quelles propriétés mathématiques résultent de ces choix pour savoir si oui ou non des algorithmes classiques de fouille peuvent être utilisés ?

– Différence de niveau sémantique. Les données qui se rapportent à un objet complexe ne se situent pas toujours toutes sur le même niveau

d'abstraction. Ce phénomène bien connu dans le domaine de la représentation des connaissances et notamment dans les ontologies prend une nouvelle dimension encore plus difficile à maîtriser en fouille de données. Par exemple, un compte-rendu médical écrit par un médecin sur un patient peut être le résultat d'une interprétation de clichés et d'examens biologiques. Par conséquent, le niveau sémantique du texte est différent de celui des images. Dans ce cas, les attributs issus des comptes-rendus textuels auront du mal à être alignés sur ceux issus des images radiologiques ou des examens biologiques. Comment alors intégrer ces niveaux sémantiques pour ensuite pouvoir décrire des patients ou les comparer ? Le texte devrait-il être vu comme un subsumant des images et des données biologiques ? Difficile d'y répondre promptement.

- Fusion des données et intégration des connaissances du domaine. Souvent, dans nos processus d'interprétation des situations qui nous entourent, comme être humains, nous pouvons mieux inférer grâce à une contextualisation des données que nous recevons par rapport à d'autres qui leurs sont liées indirectement. Par exemple, un grand opérateur de téléphonie dont les entrepôts de données sont extrêmement volumineux cherchera à contextualiser ses clients par rapport aux caractéristiques de leur quartier d'habitation, par rapport aux spécificités des moments d'appel (fin de semaine, jour ou nuit, période de vacances etc.). La base clients peut ainsi être enrichie par des informations indirectes qui fournissent un contexte susceptible d'améliorer l'interprétation. On peut également y introduire des connaissances formelles, par exemple, pour le diagnostic médical, certaines hypothèses peuvent être renforcées grâce aux connaissances médicales disponibles. Ainsi, compte tenu d'un certain profil, on doit pouvoir adjoindre d'autres informations dans le dossier patient. Ce procédé est particulièrement développé dans la fouille de données textuelles et est généralement destiné à améliorer la désambigüisation et fait souvent appel à des ontologies de domaine.
- Rareté de certains phénomènes. En effet, certains phénomènes peuvent être rares et donc noyés dans la masse des données. Par exemple,

dans les transactions bancaires, celles qui sont frauduleuses sont assez rares. Dans un processus de fouille, ces phénomènes peuvent être difficile à déceler et dans ce cas, il faut les traiter de manière plus spécifiques.

Défis Technologiques Ils sont étroitement liés aux défis scientifiques et ils sont parfois des conséquences des difficultés scientifiques. Parmi les défis technologiques identifiés, on peut citer :

- Le passage à l'échelle ("scalabilité"). Nous pouvons en effet disposer de solution formelle et même opérationnelle sans pour autant être en mesure de l'utiliser sur des corpus réels de fouille, soit pour des raisons de temps de calcul soit pour des raisons d'espace mémoire. Par exemple, et pour rester dans le cas DMP, une classification d'une population de patients s'avère quasi impossible de façon directe en prenant en compte la totalité des informations. Outre le problème de l'alignement des attributs, le mélange du type de données, la dimension élevée de l'espace de représentation, le grand nombre d'observations etc. rendent cette opération quasi-impossible de façon directe. On peut alors s'interroger sur : Comment revenir sur les aspects méthodologiques pour développer des algorithmes appropriés incrémentaux par exemple ? Ou comment mieux exploiter les ressources physiques des machines ? Le GRID computing est l'une des réponses possibles, est-elle la réponse sur des applications réelles ? Une autre approche serait : Comment travailler sur des vues partielles des objets ? La fouille de données multi-tables tente d'y répondre.
- Un processus de fouille de données destiné à produire des connaissances à partir de données en perpétuelle évolution. Le processus de fouille devrait alors être continu pour identifier à temps les éventuelles modifications majeures au niveau des connaissances qui pourraient survenir sur un phénomène modélisé. Comment alors assurer le couplage fort entre inférence sur des cas et amélioration incrémentale des connaissances qui sous-tendent cette inférence.

Bien que le cadre général des travaux que nous réalisons au sein de la société Fenics couvrent une large partie des problèmes évoqués, dans le cadre

de cette thèse nous nous sommes attelés à traiter deux question de manière plus spécifiques, à savoir la haute dimensionnalité des données d'une part et la rareté des phénomènes d'autres part. Ces deux problèmes impactent en effet de manière direct les algorithmes d'apprentissage.

2.3 Apprentissage

2.3.1 Notations et Définitions

Soit Ω une population d'individus ou d'objets concernés par le problème d'apprentissage. Cette population est généralement de taille infinie. On notera par ω un individu de Ω et en cas de besoin, on utilisera un indice pour différencier deux individus, ω_i et ω_j par exemple. A chaque individu de cette population sont associées deux catégories de variables ou d'attributs :

Attributs Exogènes Il s'agit de l'ensemble des variables descriptives des individus. Dans le contexte de l'apprentissage supervisé, on les appellera également variables explicatives et elles seront notées $X_1, \ldots, X_p$. Ainsi :

- toute variable exogène X_j, pour tout $j = 1, \ldots, p$, peut être vue comme une application qui à tout individu $\omega \in \Omega$ associe une valeur $X_j(\omega)$ prise dans un domaine de valeurs noté D_j :
$$X_j : \Omega \longmapsto D_j$$
$$\forall \omega \in \Omega \longmapsto X(\omega) \in D_j$$
- De manière analogue, l'ensemble des variables $\mathbf{X} = (X_1, \ldots, X_p)$ peut être vu comme une application qui à tout individu $\omega \in \Omega$ associe une valeur $X(\omega) = (X_1(\omega), \ldots, X_p(\omega))$ prise dans un espace à p dimensions $D = D_1 \times D_2 \times \ldots, \times D_p$:
$$\mathbf{X} : \Omega \longmapsto D$$
$$\forall \omega \in \Omega \longmapsto \mathbf{X}(\omega) \in D$$

Selon les propriétés mathématiques des domaines D_j on parlera de variable quantitative ou qualitative. Par exemple, si chaque D_j peut être assimilé à la droite des réels R, alors D est espace vectoriel à p

dimensions.

En toute généralité, les variables exogènes peuvent être quantitatives ou qualitatives, par conséquent, aucune structure mathématique particulière n'est imposée à D.

Attributs Endogènes Il s'agit des variables à prédire. Contrairement aux variables exogènes, cette catégorie de variables, qui va jouer un rôle spécifique, n'est identifiée que dans le contexte de l'apprentissage supervisé. On notera $Y_1, \ldots, Y_m$ ces variables endogènes. Pour distinguer les domaines de valeurs des X_j de ceux des Y_k, on notera pour cette dernière catégorie E_k. Dans la plus part des méthodes d'apprentissage supervisé, on ne dispose que d'une variable endogène notée simplement Y et son domaine de valeur est noté E. Si E peut être assimilé à l'ensemble des réels, on parlera de variable endogène continue et on considère alors que le problème d'apprentissage relève des méthodes de régression. Dans le cas ou E est un ensemble discret et de cardinal relativement faible, 2 ou 3, on parlera d'un problème de classement et les valeurs de $= \{e_1, \ldots, e_t\}$ sont appelées des étiquettes ou des classes.

Pour simplifier notre exposé, sans pour autant nuire à sa généralité, supposons que nous sommes dans le cas courant d'une variable endogène Y à prédire. C'est en tout cas, le cadre dans lequel se place cette thèse. Supposons également que nous sommes dans un cadre de classement, c'est-à-dire, dans le cas où la varibale endogène prend ses valeurs dans un ensemble d'étiquettes restreint.

Par exemple, si la population Ω est celle des femmes on désignera par Y le résultat de l'examen mammographique qui peut prendre deux valeurs par exemple : e_1=Présence d'un Cancer ou e_2=Absence d'un cancer. Dans la réalité, l'observation de $Y(\omega)$ n'est pas toujours facile et ceci pour des raisons diverses. Par exemple, le cancer est présent mais ne s'est pas manifesté de manière flagrante. Ou bien, malgré un examen mammographique, le cancer reste indécelable à l'oeil nu ou encore, comme cela arrive fréquemment, pour des raisons de fatigue ou d'inattention le radiologue ne le voit pas. C'est la raison pour laquelle nous cherchons un moyen φ capable de déceler la classe (cancer ou non) Y de manière plus systématique et plus fiable. La

détermination du modèle de prédiction φ est liée à l'hypothèse selon laquelle les valeurs prises par la variable statistique Y ne relèvent pas du hasard, mais de certaines situations particulières que l'on peut caractériser à partir des variables exogènes. Par exemple, on peut s'attendre à ce que des individus ayant des valeurs identiques sur les variables exogènes devraient avoir des valeurs identiques sur la variable endogène. Même si cette attente peut être contredite à cause du bruit ou du manque d'information, elle devrait être vraie dans la plus part des cas.

2.3.2 Apprentissage supervisé

Du fait que la variable endogène est difficile d'accès, l'objectif de l'apprentissage supervisé vise à mettre au point la fonction de prédiction φ qui se substitue à Y et qui se calcul à partir de l'espace de représentation. La figure 2.3 résume cet objectif.

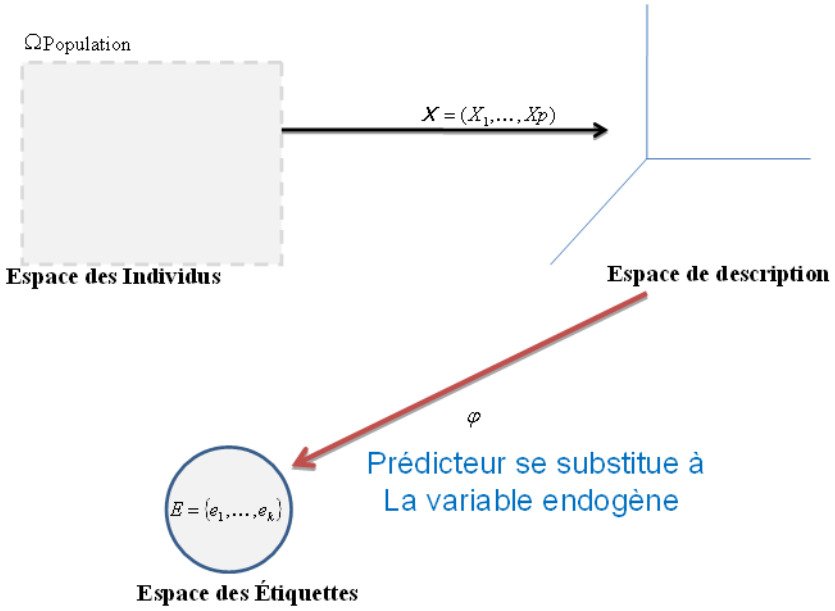


FIG. 2.3 – Objectif du processus d'apprentissage

Pour cela, il est nécessaire de de prélever dans la population Ω un échantillon Ω_a dit d'apprentissage pour lequel nous supposons connus pour tous ses individus les valeurs des variables exogènes et endogène : $\forall \omega \in \Omega_a, (X(\omega)), Y(\omega))$ sont connus. L'apprentissage va consister à identifier, au moyen d'un algorithme d'apprentissage donné, le modèle de prédiction φ. Il existe une large variété d'algorithmes d'apprentissage susceptibles d'exhiber φ. Nous en présentons quelques algorithmes dans la section 3. La figure 2.4 résume cette phase d'apprentissage.

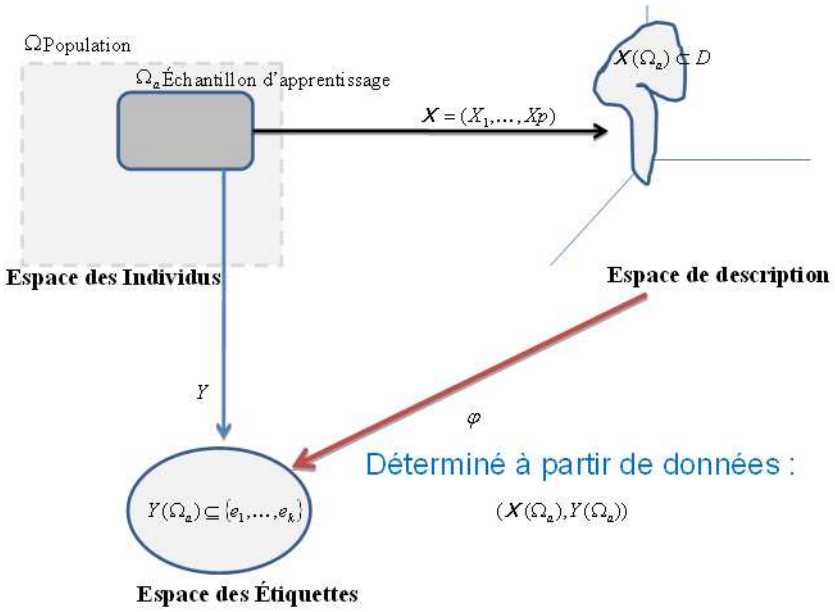


FIG. 2.4 – Phase d'apprentissage

A l'issue de l'apprentissage, il convient d'évaluer le modèle sur un nouvel échantillon $\Omega_t \subset \Omega$ dit de test, pour lequel nous connaissons également pour chacun de ses individus ω la valeur du couple $(X(\omega), Y(\omega))$. Le modèle sera d'autant meilleure que $\sum_{\omega \in \Omega_a} (\varphi(X(\omega)) \neq Y(\omega))$ est proche de zéro sur un échantillon suffisamment grand. Il y a d'autres procédés plus sophistiqués pour évaluer la qualité des modèles, nous y reviendrons plus loin. Le schéma 2.5 résume le principe de la validation.

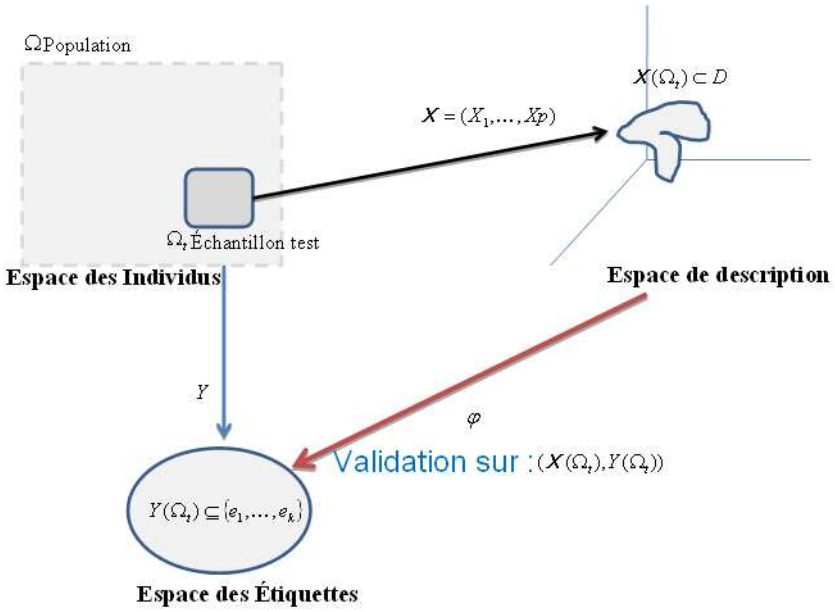


FIG. 2.5 – Validation sur échantillon test

Enfin, une fois le modèle de prédiction explicité et jugé acceptable par l'utilisateur au regard de l'application visé, il est alors mis en service dans une application et servira à estimer la variable endogène par la seul observation des valeurs des variables exogènes comme indiqué dans le schéma 2.6.

2.4 Conclusion

La présentation que nous venons de donner sur l'apprentissage dans un cadre limité à une variable endogène discrète peut parfaitement être étendu à une variable endogène quantitative. Dans ce cas, on parlera de régression et la validation du modèle consiste à minimiser l'erreur quadratique moyenne :

$$\sum_{\omega \in \Omega_a} (\varphi(X(\omega)) - Y(\omega))^2$$

Si le nombre de variables dépasse 1, on pourra toujours imaginer une formule de calcul d'écart entre le vecteur prédit et le vecteur calculé et cet écart devra

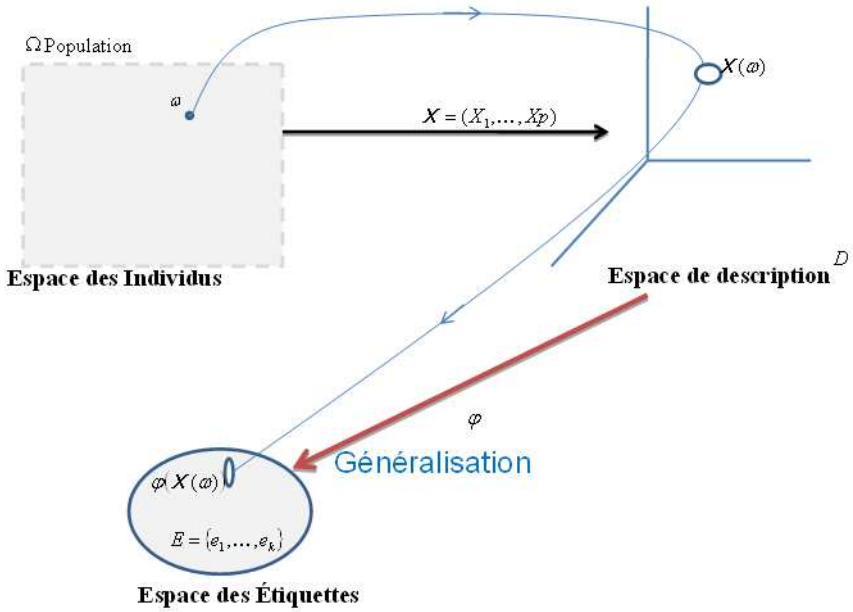


FIG. 2.6 – Principe de la généralisation

être minimum. Dans la suite de la thèse, nous ne nous intéresserons qu'au cas d'une variable endogène à nombre de classe restreint.

Chapitre 3

Méthodes d'apprentissage de classes

3.1 Introduction

Il existe différentes approches dans le domaine de l'apprentissage supervisé pour répondre au problème de la construction d'un modèle de prédiction. La détermination des différentes classes de méthodes va essentiellement dépendre de la nature et des propriétés mathématiques des espaces de prédiction où sont définies la ou les variables endogènes. La table 3.1 montre cette répartition.

D'autres familles de méthodes ne figurent pas dans cette table car elles reposent non pas sur les propriétés de l'espace de prédiction mais sur les relations entre les individus. Ils s'agit des approches "à base d'instances". Elles s'appuient sur la construction d'une structure de voisinage à partir d'une matrice de similarité entre chaque individu. La prédiction affectée à un nouvel individu se fait par analyse de son voisinage, et est utilisé principalement pour des problèmes de classification même si des extensions peuvent être imaginées pour des problèmes de type régression. Les différentes méthodes se distinguent principalement par la définition du voisinage utilisé (voir figure

Espace de prédiction			
$\Re$	variable qualitative	$\Re^p$	espace quelconque
Régression Ex : régression simple (méthode des moindres carrés), régression polynomiale, arbre de régression, régression PLS, ...	*Classification* Ex : SVM, arbre de classification, analyse discriminante, régression logistique, forêt aléatoire, ...	*Analyse canonique* Ex : analyse canonique généralisée, Analyse en composante principale de variables instrumentales (ACPVI),...	nécessite une structuration pour se ramener à l'un des précédents cas.

TAB. 3.1 – Répartition de différentes classes de méthodes de construction de modèles de prédiction en fonction des espaces de prédiction

3.1).

Citons par exemple :

- les $K - PPV$ (k-plus proches voisins) où le voisinage est défini par un nombre de voisins les plus proches à prendre en considération.
- les ϵ-*voisins* où chaque voisin à considérer se trouve dans une hypersphère de rayon ϵ autour de l'individu que l'on cherche à prédire.
- les fenètres de Parzen où le voisinage considéré est une forme lissée autour de l'individu à prédire fonction de la densité locale.

Comme annoncé dans les chapitres précédents, notre recherche a pour cadre le cas de la classification, nous allons en détailler quelques méthodes phares. On peut trouver dans [HTF01, Sap90, Ten07] un exposé plus détaillé de ces différentes méthodes. Nous nous contenterons ici de rappeller les principes généraux.

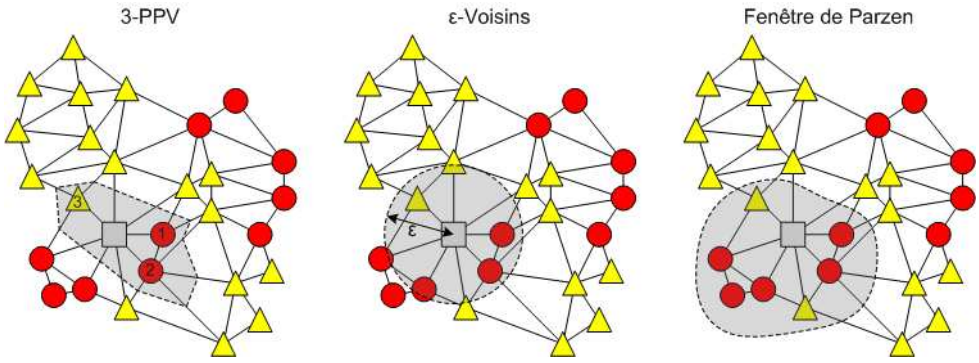


FIG. 3.1 – Illustration des voisinages utilisés pour différentes approches à base d'instance

3.2 Analyse discriminante

L'analyse discriminante est l'une des plus anciennes techniques de discrimination. Elle a été proposée par Fischer en 1936 [Fis36]. Cette technique est restée très populaire. Le problème pratique traité par Fischer pour illustrer cette méthode concerne la discrimination de trois classes de la famille des iris (versicolor, setosa et verginica). Pour cela, il a pris un échantillon de 150 iris répartis sur les trois classes et, pour chaque fleur, il a mesure la longueur et la largeur des pétales et des sepales. Le fichier des ces données se trouve dans la plupart des ouvrages consacrés à cette méthode et sur les sites web des communautés d'apprentissage automatique comme celui de l'Universite de Californie à Irvine [HB99] ou celui de la communauté de *data mining* comme Kdnuggets [1].

Le principe de l'analyse discriminante est relativement simple. Considérons p variables exogènes toutes quantitatives $(X_1, X_2, ..., X_p)$, $p = 4$ dans le cas des iris de Fischer, et une variable endogène Y qualitative qui prend ses valeurs dans un ensemble $E = e_1, e_2, ..., e_t$, avec $t = 3$ dans le cas des iris. Supposons que ces variables ont été centrées, c'est-à-dire que $\overline{X}_j = 0 \ \forall j$.

Le problème que resout l'analyse discriminante consiste à construire une

[1] http ://www.kdnuggets.com/

variable Z, combinaison linéaire de $(X_1, X_2, ..., X_p)$, telle que :

- Pour tout individu ω_i de la classe e_k on ait $Z(\omega_i)$ "peu différent" de $\overline{Z}_k$ $\forall k$ où $\overline{Z}_k$ désigne la moyenne de Z dans la classe e_k.
  Nous pouvons traduire cela en exigeant que la dispersion de Z soit minimale dans chaque classe. Nous cherchons donc à minimiser la variance de Z à l'intérieur de chaque classe, ce qui donne globalement, pour toutes les classes, le critère de la variance intra classe à minimiser, dont l'expression est :

  $$V_{intra} = \frac{1}{card(\Omega)} \sum_{k=1}^{t} card(\Omega_k) V_k(Z)$$

  où $card(\Omega_k)$ représente l'effectif de la classe e_k et $V_k(Z)$ désigne la variance de Z dans la classe e_k.

- Pour tout individu ω_a n'appartenant pas à la classe e_k on ait $Z(\omega_i) \neq Z(\omega_a)$. Cela représente le contraste entre les classes. Pour que ce contraste entre classes soit le plus fort possible, on cherchera à déterminer Z telle que les moyennes $\overline{Z}_k$ $\forall k$ soient les plus dispersées possibles. Cela revient à maximiser la variance interclasses dont l'expression est :

  $$V_{inter} = \frac{1}{card(\Omega)} \sum_{k=1}^{t} card(\Omega_k)(\overline{Z}_k - \overline{Z})^2$$

Nous sommes ainsi en face d'un problème d'optimisation que l'on formule ainsi :
Trouver Z, une combinaison linéaire des variables exogènes $(X_1, X_2, ..., X_p)$, qui minimise la variance intra classes et qui maximise la variance inter classes. Ce problème se resout facilement car cela revient à chercher des valeurs propres et des vecteurs propres associés à une matrice de variances.

Dans le cas où nous chercherions à discriminer entre t classes, on démontre aisément que l'on peut trouver au plus $(t-1)$ droites. Ces droites

s'appellent axes factoriels discriminants. On peut donc les utiliser par paire pour visualiser les classes. Un nouvel individu à classer est affecté à la classe dont le centre de gravité est le plus proche. On peut définir géometriquement des surfaces de décision par l'intersection des médiatrices sur les droites qui relient les centres de gravité des classes comme illustré en figure 3.2 sur le jeu de données des iris de Fisher. Dans cet exemple, deux solutions Z_1 et Z_2 sont trouvées. En notant respectivement les 4 variables exogènes centrées : largeur du pétale, longueur du pétale, largeur du sépale et longueur du sépale, X_1, X_2, X_3 et X_4, Z_1 et Z_2 s'expriment par les combinaisons linéaires suivantes :
$Z_1(\omega) = 0.58X_1 + 0.94X_2 - 0.53X_3 - 0.42X_4$
$Z_2(\omega) = -0.57X_1 + 0.4X_2 - 0.73X_3 - 0.02X_4$

La présentation de l'analyse discriminante que nous venons de donner utilise trois hypothèses qui lui offre d'interessantes propriétés mathématiques. Ces trois hypothèses sont : toutes les variables sont quantitatives, elles sont normalement distribuées et les classes sont linéairement séparables. Ces hypothèses sont hélas rarement toutes vérifiées, mais ne sont pas toujours indispensables dans la pratique. Notons que de nouveaux développements ont été introduits pour élargir le champ d'application à des frontières quadratiques ou à des données catégorielles.

3.3 SVM : Support Vector Machines

Les *Support Vector Machines* ou SVM sont une famille d'algorithmes d'apprentissage définis pour la prévision d'une variable endogène qualitative initialement binaire, c'est-à-dire un problème de classification à deux classes. Ils ont été ensuite généralisés pour les problèmes à plus de deux classes et à la prévision d'une variable quantitative. Dans le cas de la discrimination d'une variable dichotomique, ils sont basés sur la recherche de *l'hyperplan de marge optimale* qui, lorsque c'est possible, sépare correctement les données tout en étant le plus éloigné possible de toutes les observations. Le principe

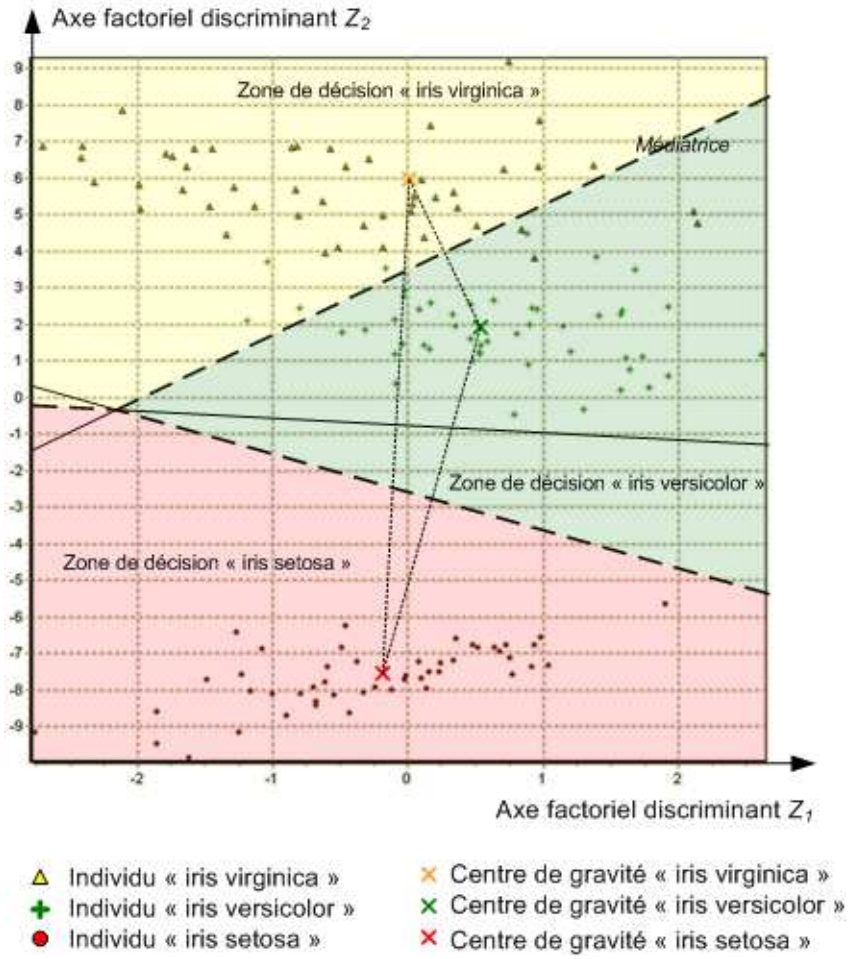


FIG. 3.2 – Visualisation de l'analyse discriminante du jeu de données Iris

est de trouver une fonction de discrimination, $\varphi(X)$, dont la capacité de généralisation est la plus grande possible.

Les débuts des SVM proviennent des travaux de Vapnik en apprentissage en 1995 [Vap95]. Le principe de base des SVM consiste à ramener le problème de classification à celui, linéaire, de la recherche d'un hyperplan optimal. Deux idées permettent d'atteindre cet objectif :

1. La première consiste à définir l'hyperplan comme solution d'un problème d'optimisation sous contraintes dont la fonction objectif s'exprime à l'aide de produits scalaires entre vecteurs.
2. Le passage à la recherche de surfaces séparatrices non linéaires est obtenu par l'introduction d'une fonction noyau dans le produit scalaire induisant implicitement une transformation non linéaire des données vers un espace intermédiaire.

Les SVM sont largement utilisés dans de nombreux types d'application. L'introduction de noyaux, pouvant être spécifiquement adaptés à une problématique donnée, lui confère une grande flexibilité pour s'adapter à des situations très diverses (reconnaissance de formes, de caractères, détection de spams, diagnostics...).

Considérons une variable endogène unique et quantitative Y, par soucis de simplicité nous la considérerons dans un premier temps binaire à valeurs dans $-1; 1$. Soit $X = (X_1, X_2, ..., X_p)$ les variables exogènes et $\omega_i \in \Omega_a$ un individu de notre échantillon d'apprentissage Ω_a. Nous cherchons à déterminer une fonction $\varphi(X)$ tel que $\sum_{\omega \in \Omega_a} (\varphi(X(\omega)) \neq Y(\omega))$ soit minimale.

Dans ce cas ($Y(\omega) \in -1; 1 \ \forall \omega$), le problème se pose comme la recherche d'une frontière de décision dans l'espace de description. Un compromis doit être trouvé entre la complexité de cette frontière (sa capacité d'ajustement), et les qualités de généralisation de ce modèle (figure 3.3).

Plutôt que de rechercher directement φ, on va essayer de trouver une fonction f à valeur dans $\Re$ dont le signe fournira la prédiction : $\varphi = signe(f)$. L'erreur s'exprime alors comme le nombre de fois où le produit $Y \times f(X)$ est négatif. De plus, la valeur absolue de cette quantité $|Y \times f(X)|$ fournit une

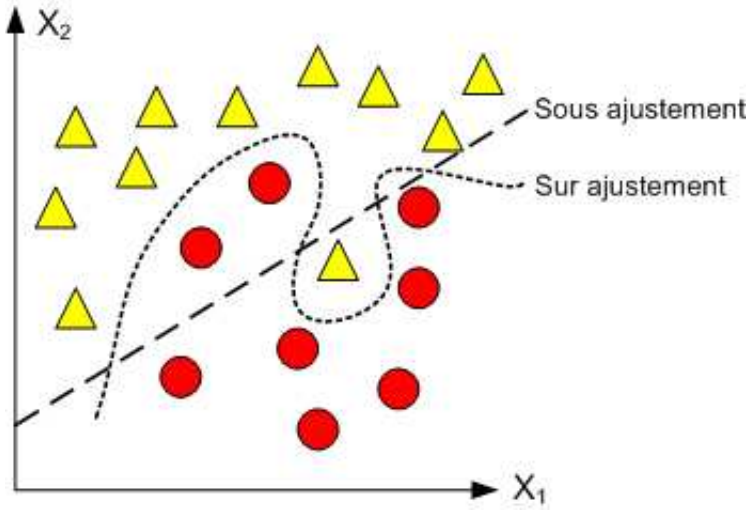


FIG. 3.3 – Illustration du compromis à trouver entre sous ajustement et sur ajustement

indication sur la confiance à accorder au résultat du classement, il s'agit de la marge de f.

Dans le cas linéaire, un hyperplan est défini par un produit scalaire $S.X(\omega) + b = 0$ où S est un vecteur orthogonal au plan et b une constante, d'où $f(\omega_i) = S.X(\omega_i) + b$ dont le signe fournit le côté du plan où se trouve l'individu. Ainsi un plan sera séparateur si :
$Y \times f(\omega_i) > 0 \ \forall i$
Dans le cas où la séparation est possible, parmi tous les hyperplans solutions pour la séparation des individus, on choisit celui qui se trouve le plus "loin" possible de tous les exemples, il est alors appelé *hyperplan de marge maximale*. La figure 3.4 illustre cette notion dans le cas linéaire. Si les individus ne sont pas séparables ont introduit un terme d'assouplissement des contraintes ϵ, un plan sera candidat si $Y \times f(\omega_i) > (0 - \epsilon) \ \forall i$.

Comme énoncé plus haut, la deuxième spécificité des SVM est la recherche de surfaces séparatrices non linéaires. Celle-ci se fait par l'introduction d'une fonction noyau, notée k, dans le produit scalaire induisant une transformation non linéaire des données vers un espace intermédiaire. En théorie, une

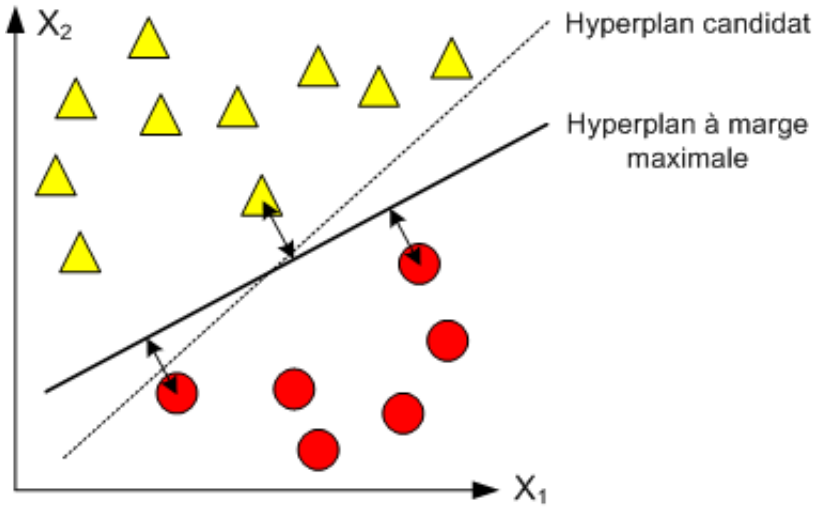


FIG. 3.4 – Illustration de l'hyperplan à marge maximal dans un cas linéairement séparable

fonction k symétrique est un noyau si, pour tous les ω_i, la matrice de terme général $k(\omega_i, \omega_j)$ est une matrice définie positive c'est-à-dire quelle définit une matrice de produit scalaire. Malheureusement, cette condition ne donne aucune indication sur la construction de la fonction noyau. En pratique, cela consiste à combiner des noyaux simples pour en obtenir des plus complexes, associés à la situation rencontrée. Les noyaux les plus fréquemment rencontrés sont linéaires, polynomiaux ou gaussiens. Beaucoup d'articles sont consacrés à la construction de noyaux plus ou moins complexes et adaptés à une problématique donnée. La grande flexibilité dans la définition des noyaux confère beaucoup d'efficacité à cette approche à condition biensûr de construire et tester le bon noyau.

3.4 Régression logistique

La régression logistique est une technique statistique qui a pour objectif de produire un modèle permettant de prédire les valeurs prises par une variable endogène quantitative Y, le plus souvent binaire, à partir de p variables

exogènes continues et/ou binaires $(X_1, X_2, ..., X_p)$. La régression logistique est largement répandue dans de nombreux domaines. On peut citer de façon non-exhaustive :

- En médecine, elle permet par exemple de trouver les facteurs qui caractérisent un groupe de sujets malades par rapport à des sujets sains.
- Dans le domaine des assurances, elle permet de cibler une fraction de la clientèle qui sera sensible à une police d'assurance sur tel ou tel risque particulier.
- Dans le domaine bancaire, pour détecter les groupes à risque lors de la souscription d'un crédit.

Le succès de la régression logistique repose notamment sur les nombreux outils qui permettent d'interpréter de manière approfondie les résultats obtenus.

Nous nous limiterons pour notre explication au cadre de la régression logistique binaire, où la variable Y prend deux modalités possibles 1 ou 0. Les variables exogènes $X = (X_1, X_2, ..., X_p)$ sont exclusivement continues ou binaires. Nous noterons :

- $p(1)$, la probabilité $p(Y = 1)$, c'est-à-dire la probabilité a priori pour que $Y = 1$, et de la même manière $p(0)$, $p(Y = 0)$.
- $p(X|1)$, respectivement $p(X|0)$, la distribution conditionnelle des X sachant la valeur de Y, respectivement 1 ou 0.
- $p(1|X)$ et $p(0|X)$, les probabilités a posteriori d'obtenir la modalité 1 et 0 de Y, sachant la valeur prise par X.

La régression logistique repose sur l'hypothèse fondamentale suivante :

$$\ln \frac{p(X|1)}{p(X|0)} = a_0 + a_1 X_1 + ... + a_p X_p$$

Une vaste classe de distributions répondent à cette spécification, la distribution multinormale par exemple, mais également d'autres distributions, notamment celles où les variables exogènes sont booléennes (0/1). Par rapport à l'analyse discriminante, ce n'est plus les densités conditionnelles $p(X|1)$, et $p(X|0)$, qui sont modélisées mais le rapport de ces densités. La restriction

introduite par l'hypothèse est moins forte.

La spécification ci-dessus peut être écrite de manière différente. On désigne par le terme LOGIT de $p(1|X)$, l'expression suivante :

$$\ln \frac{p(1|X)}{1-p(1|X)} = b_0 + b_1 X_1 + ... + b_p X_p$$

Il s'agit d'une *régression* car on veut montrer une relation de dépendance entre une variable endogène et une série de variables exogènes. Il s'agit d'une régression *logistique* car la loi de probabilité est modélisée à partir d'une loi logistique. En effet, après transformation de l'équation ci-dessus, nous obtenons :

$$p(1|X) = \frac{e^{b_0+b_1X_1+...+b_pX_p}}{1+e^{b_0+b_1X_1+...+b_pX_p}}$$

Le but est désormais d'estimer les coefficients b_j, de la fonction LOGIT. Il est très rare de disposer pour chaque combinaison possible des $X_j, (j = 1, ..., p)$, de suffisamment d'individus pour disposer d'une estimation fiable des probabilités $p(1|X)$ et $p(0|X)$. La solution passe alors par exemple par la méthode de maximisation de la vraisemblance.

La probabilité d'appartenance d'un individu ω à une classe peut être vu comme une contribution à la vraisemblance. Y suivant une loi de Bernouilli de paramètre $p(Y(\omega) = 1|X(\omega))$, la vraisemblance L d'un échantillon Ω_a, s'écrit :

$$L = \prod_{\omega \in \Omega_a} p(Y(\omega) = 1|X(\omega))^{Y(\omega)} \times (1 - p(Y(\omega) = 1|X(\omega)))^{1-Y(\omega)}$$

Les paramètres $b_j, (j = 0, ..., p)$, qui maximisent cette quantité sont les estimateurs du maximum de vraisemblance de la régression logistique.

Dans la pratique, les logiciels utilisent une procédure approchée pour obtenir une solution satisfaisante de la maximisation ci-dessus. Ce qui explique

d'ailleurs pourquoi ils ne fournissent pas toujours des coefficients strictement identiques. Les résultats dépendent de l'algorithme utilisé et de la précision adoptée lors du paramétrage du calcul. Notons β, le vecteur des paramètres à estimer. La procédure la plus connue est la méthode Newton-Raphson qui est une méthode itérative du gradient. Elle s'appuie sur la relation suivante :

$$\beta^{i+1} = \beta^i - \left(\frac{\partial^2 L}{\partial\beta\partial\beta'}\right)^{-1} \times \frac{\partial L}{\partial\beta}$$

- β^i, est la solution courante à l'étape i, $\beta^0 = (0, ..., 0)$, est une initialisation possible.
- $\frac{\partial L}{\partial\beta}$, est le vecteur des dérivées partielles première de la vraisemblance.
- $\frac{\partial^2 L}{\partial\beta\partial\beta'}$, est la matrice des dérivées partielles secondes de la vraisemblance.
- les itérations sont interrompues lorsque la différence entre deux vecteurs de solutions successifs sont négligeables.

Cette dernière matrice, dite Matrice hessienne, est intéressante car son inverse représente l'estimation de la matrice de variance co-variance de β. Elle sera mise en contribution dans les différents tests d'hypothèses pour évaluer la significativité des coefficients.

Lors de l'application, pour classer un nouvel individu ω, en considérant la fonction LOGIT, il faut alors s'appuyer sur la règle d'affectation :

$$Y(\omega) = 1 \Leftrightarrow \beta_0 + \beta_1 \times X_1(\omega) + ... + \beta_p \times X_p(\omega) > 0$$

Notons qu'il existe d'autres fonctions que LOGIT utilisées en régression logistique, notons entre autre :

- *probit* : il s'agit de la fonction inverse de la fonction de répartition d'une loi normale, mais son expression n'est pas explicite.
- *log-log* : $\ln(-\ln(1 - p(1)))$ mais notons que celle-ci est dissymétrique.

3.5 Arbres de décision

La construction d'arbres de décision remonte principalement aux travaux de Kass (1980) avec l'algorithme CHAID [Kas80], Breiman et al. (1984) avec l'algorithme CART [BFOS84], et Quinlan avec les algorithmes ID3 (1986) [Qui86] et C4.5 (1993) [Qui93]. La figure 3.5 présente un exemple d'arbre de décision construit à partir de l'algorithme C4.5 sur le jeu de données *Iris* [HB99].

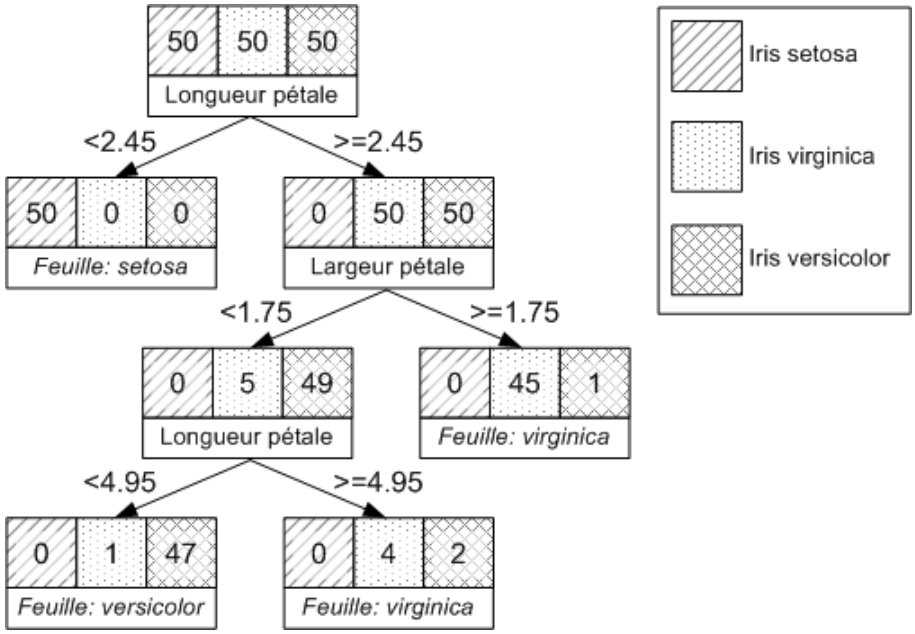


FIG. 3.5 – Exemple d'arbre de décision construit avec C4.5 sur le jeu de données Iris [HB99] à trois classes.

Schématiquement, il s'agit de déterminer un partitionnement des individus de la base d'apprentissage de telle sorte que les partitions obtenues permettent de classer le plus correctement possible un nouvel individu. Ainsi on essaie généralement de construire des partitions les plus homogènes possibles. Le partitionnement est représenté par un arbre de décision, à savoir un graphe sans cycle et connexe, où chaque noeud correspond à une question sur la valeur prise par une variable exogène. Chaque branche de l'arbre se

termine par une feuille à laquelle on affecte une des modalités de la variable endogène.

Partant d'un ensemble d'apprentissage $\Omega_a 0$, d'un arbre vide et en initialisant le noeud courant à la racine de l'arbre, les algorithmes d'apprentissage d'arbres de décision reposent sur trois étapes principales :

1. Le noeud courant est-il terminal selon un critère d'arrêt ?
2. Si oui, lui affecter une classe.
3. Si non, sélectionner une variable exogène et partitionner Ω_{a0} en sous ensembles $\Omega_{b0}, \Omega_{b1}, ..., \Omega_{bk}$ selon la variable sélectionnée en fonction d'un critère d'information. Puis construire les sous-arbres sur $\Omega_{b0}, \Omega_{b1}, ..., \Omega_{bk}$.

Les différentes implémentations se distinguent par les choix de critères utilisés à chacune de ces trois étapes : le critère d'arrêt, le critère d'affectation de classe et le critère de partitionnement.

Le critère d'arrêt détermine le moment où la récursion doit cesser et donc la croissance de l'arbre. Etre dans un noeud pur, c'est-à-dire lorsque tous les individus du noeud appartiennent à la même classe, est le critère d'arrêt trivial. Notons ici qu'il s'agit du critère d'arrêt utilisé lors de la construction d'arbres pour une forêt aléatoire. On peut citer d'autres critères d'arrêt classiques :

- L'absence d'apport informationnel (mesurée par le critère de partitionnement).
- Un nombre minimal d'individus autorisant la poursuite du partionnement.
- Une profondeur maximale, c'est-à-dire un nombre de noeuds maximal entre la racine et la feuille.

Lors de l'utilisation de ces deux derniers critères, il faut savoir qu'il est reconnu que le risque d'arrêter trop tôt la croissance d'un arbre est plus grand que d'arrêter trop tard. Breiman et al. [BFOS84] ont ainsi proposé la possibilité de réduire la taille d'un arbre après l'avoir construit, il s'agit du post-élagage. Cette méthode consiste dans un premier temps à produire des

arbres les plus purs possibles puis dans un second temps, à réduire l'arbre en utilisant un autre critère pour comparer ce dernier à des arbres de tailles inférieures afin d'obtenir un arbre plus performant en classement.

Le critère de partitionnement doit permettre de choisir parmi les différentes variables exogènes celle qui sera utilisée pour réaliser la partition sur le noeud courant non terminal. Les critères utilisés doivent mesurer parmi plusieurs partitions possibles celle qui est la plus porteuse d'information. Il faut donc que le critère utilisé caractérise la pureté (ou le gain en pureté) lors du passage du noeud parent aux noeuds (ou parfois feuilles) fils. Il existe un grand nombre de mesures de qualité de partitionnement. La plupart des études comparatives concluent que le choix de la mesure influence la taille de l'arbre mais très peu la qualité de la prédiction [Shi99], [LLS00]. De telles mesures prennent une valeur minimale lorsque le noeud est pur et maximale lorsque les individus présents sont équirépartis vis-à-vis de la variable endogène. Parmi les mesures les plus classiques mentionnons l'entropie de Shannon et l'indice de Gini, respectivement utilisés dans C4.5 [Qui93], et CART [BFOS84] :
Soit t le nombre de classes et p_i la proportion d'individus de la i^{eme} classe,
Entropie de Shannon :

$$H(P) = \sum_{i=1}^{t} -p_i \log_2(p_i)$$

Indice de Gini :

$$H(P) = \sum_{i=1}^{t} p_i(1-p_i) = 1 - \sum_{i=1}^{t} p_i^2$$

La figure 3.6 illustre ces deux mesures dans le cas d'un problème à deux classes. Wehenkel [Weh96] étudie les familles de fonctions entropiques et montre la forte similarité existant entre l'entropie de Shannon et l'indice de Gini. Notons que l'indice de Gini est le critère utilisé pour la construction des arbres d'une forêt aléatoire.

Lorsqu'un noeud est terminal, il reste à affecter à cette feuille une classe, ainsi lorsqu'un nouvel individu sera appliqué au modèle, s'il arrive dans cette

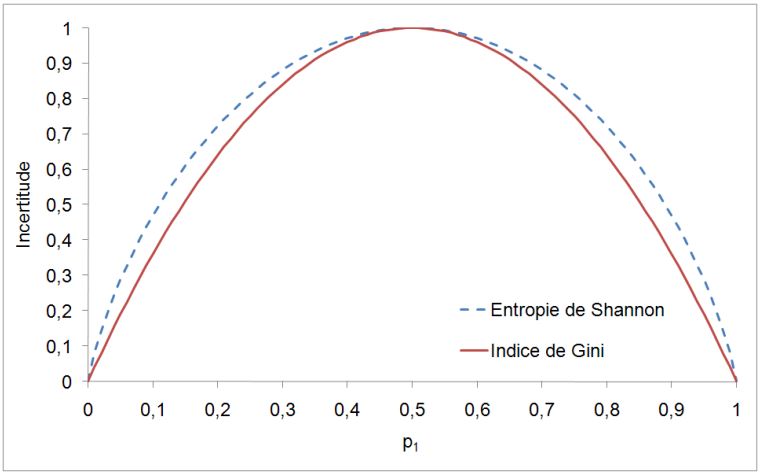


FIG. 3.6 – L'entropie de Shannon et l'indice de Gini dans un cas à deux classes.

feuille, la classe prédite pour cet individu sera la classe affectée à la feuille. Encore une fois, de nombreuses stratégies peuvent être appliquées. Lorsque la feuille est pure, on lui attribue l'unique classe représentée. Dans le cas contraire, la règle d'affectation de la classe peut différer selon les situations. En générale, la règle la plus courante est celle de la majorité, dans ce cas la classe la plus représentée est alors la classe affectée. Si les coûts de mauvaises classifications sont connus, on peut chercher à minimiser ces derniers. Dernièrement, Ritschard [Rit05] propose également d'utiliser l'intensité d'implication de Gras et al. [GAAB+96].

3.6 Agrégation de classifieurs

L'agrégation de classifieurs, ou "Méthode Ensemble", consiste à améliorer les performances d'apprentissage en combinant un grand nombre de classifieurs de base. Ces algorithmes sont basés sur des stratégies adaptatives (boosting [Fre90]) ou aléatoires (bagging [Bre96]). De nombreux articles comparatifs montrent leur efficacité sur des exemples de problèmes réels ([Gha00, VM02]). Le succès des Méthodes Ensembles tient au fait qu'ils ga-

rantissent une erreur plus faible que le meilleur des classifieurs qu'ils agrègent. En effet, comme le souligne déjà le théorème des jurés de Condorcet [Con85], si chaque membre d'un jury a une opinion indépendante sur le sujet du jugement, la probabilité que le jugement rendu par la majorité des votes soit correct augmente avec la taille du comité lorsque chaque membre a une probabilité d'avoir juste supérieure à 50%. En apprentissage, le problème est d'obtenir des classifieurs avec une opinion différente, c'est-à-dire des classifieurs qui ne commettent pas les mêmes erreurs lors de la prédiction. Ainsi, la notion de diversité décrit l'aptitude d'un ensemble à être formé de tels classifieurs ([BWHY05]).

3.6.1 Bagging

Le principe du bagging est de construire chaque classifieur de base à partir d'un échantillon bootstrap du jeux de données d'apprentissage. Un échantillon bootstrap est obtenu par tirage aléatoire avec remise de n individus du jeu de départ, n étant l'effectif de ce dernier. Le résultat est ensuite obtenu en moyennant les sorties de chaque classifieur dans le cas d'une variable endogène continue, ou par un vote à la majorité pour une variable endogène discrète.

Soit Ω_a un échantillon d'apprentissage de n individus $\omega \in \Omega$, Y une variable endogène à prédire et p variables exogènes $X = (X_1, X_2, ..., X_p)$. La construction d'un modèle de prédiction φ issu d'un bagging composé de C classifieurs de base φ'_i, $i \in (1, ..., C)$ se fait comme suit :

Pour $i = 0$ *à* C *faire*
Tirer un échantillon bootstrap Ω_b.
Construire $\varphi'_i(X)$ à partir de Ω_b.
Fin pour
Calculer $\varphi(X) = \frac{1}{C}\sum_{i=1}^{C} \varphi'_i(X)$ si Y est quantitative, ou faire un vote à la majorité entre les $\varphi'_i(X)$ si Y est qualitative.

L'un des avantages du bagging est qu'il peut s'accompagner d'une estimation "out-of-bag" de l'erreur de prévision. Il s'agit de moyenner les erreurs de prédiction de chaque classifieur de base, celles-ci étant évaluées sur les individus non tirés au sort lors du bootstrap relatif au classifieur. Cette mesure permet de juger de la qualité en généralisation du modèle ou de prévenir d'un sur-ajustement [Bes06].

3.6.2 Boosting

Le boosting adopte le même principe général que le bagging, c'est-à-dire la construction d'une famille de modèles qui sont ensuite agrégés par une moyenne des estimations ou par un vote. Cette construction de l'ensemble de classifieurs est cependant nettement différente. Lors d'un boosting chaque nouveau classifieur de base est une version adaptatée du précédent en donnant plus de poids, lors de l'estimation suivante, aux observations mal ajustées ou mal prédites. Intuitivement, cet algorithme concentre donc ses efforts sur les observations les plus difficiles à prédire, tandis que l'agrégation de l'ensemble des classifieurs permet d'éviter le sur-ajustement.

Les algorithmes de boosting existants diffèrent par leur méthodologie de pondération des erreurs pour l'obtention du classifieur suivant, le type de variable endogène étudié, ou encore les techniques d'agrégation en elles-mêmes. L'algorithme originel AdaBoost, pour *Adaptative Boosting* [Kau96], est l'un des plus utilisés, souvent avec un arbre de décision binaire comme classifieur de base.

3.6.3 Forêt aléatoire

Une forêt aléatoire, ou *Random Forest* selon son nom original anglais [Bre01], est une amélioration spécifique d'un bagging d'arbres binaires, par l'ajout d'une randomisation. Celle dernière consiste à limiter la recherche de la meilleure discrimination à k variables tirées au sort pour l'obtention

de chaque noeud. Cet ajout de hasard dans le choix des variables intervenant dans chaque arbre rend ceux-ci plus indépendants les uns des autres. Cette approche est d'autant plus pertinente et efficace pour des jeux de données hautement multidimensionnels, c'est-à-dire dans le cas où le nombre de variables exogènes est très élevé, où la randomisation combinée à la multiplication des arbres permet une meilleure exploration de l'espace de représentation.

En reprenant les notations utilisées pour le bagging (3.6.1) avec une variable endogène Y qualitative, le principe est le suivant :
Pour $i = 0$ *à* C *faire*
Tirer un échantillon bootstrap Ω_b.
Construire un arbre de décision binaire $\varphi'_i(X)$ de profondeur maximale à partir de Ω_b. Limiter lors de cette construction, la recherche de chaque noeud à k variables tirées à chaque fois au sort.
Fin pour
Faire un vote à la majorité entre les arbres $\varphi'_i(X)$.

Notons que plus k est petit, plus la diversité des arbres augmente, mais le risque de dégrader les performances de l'arbre en question aussi. Empiriquement, Léo Breiman propose d'utiliser $k = \sqrt{p}$, où p est le nombre de variables exogènes, comme paramétrage optimal [Bre01]. Nous avons retrouver cette valeur comme paramétrage optimal lors de nos tests sur les jeux de données industrielles de la société Fenics, et ceux de manière stable dès que le nombre d'arbres de la forêt est suffisamment grand. Les forêts aléatoires étant un bagging d'arbres aléatoires, elles bénécifient comme tout bagging de la possibilité de réaliser une estimation "out-of-bag".

Chapitre 4

Evaluation de modèles

4.1 Erreur en apprentissage et généralisation

L'objectif de l'apprentissage supervisé est naturellement de pouvoir utiliser les modèles construits sur de nouveaux individus. Or, calculer les différentes mesures d'évaluation sur les individus utilisés pour l'apprentissage donne souvent des valeurs bien trop optimistes (on parle alors de tests en "par coeur"), et empêche de connaître la capacité de généralisation du modèle (sa capacité à bien se comporter sur de nouveaux individus). L'idée est alors de partitionner les exemples disponibles suivant différents protocoles.

Apprentissage/test C'est le protocole le plus simple. Il consiste à séparer le jeu de données en deux parties : l'une sera utilisée pour construire le modèle ; l'autre pour le tester. Il faut juste définir la taille de chacun de ces deux échantillons. Généralement on utilise 70% des individus pour construire le modèle et 30% pour le tester. On parle alors de protocole "70/30". Le principal inconvénient de cette méthode est qu'il oblige à se passer de beaucoup d'individus pour la construction du modèle, et dans le même temps tous les individus ne sont pas exploités pour le test.

Validation croisée La validation croisée consiste à couper le jeu de départ en k subdivisions d'effectif équivalent par tirage aléatoire sans remise. Un modèle est ensuite construit sur $k-1$ subdivisions, et tester sur la subdivision restante. Ce procédé est répété pour que chaque subdivision se retrouve une unique fois en test. Ainsi chaque individu s'est retrouvé une fois en test sur un modèle où il n'a pas été utilisé pour la construction. La matrice de confusion obtenue contient donc tous les individus pour le calcul des différentes mesures. La figure 4.1 illustre ce procédé dans le cas d'une validation croisée à 5 subdivisions. Tous les tests de performances réalisé pour ce mémoire utilise cette méthode.

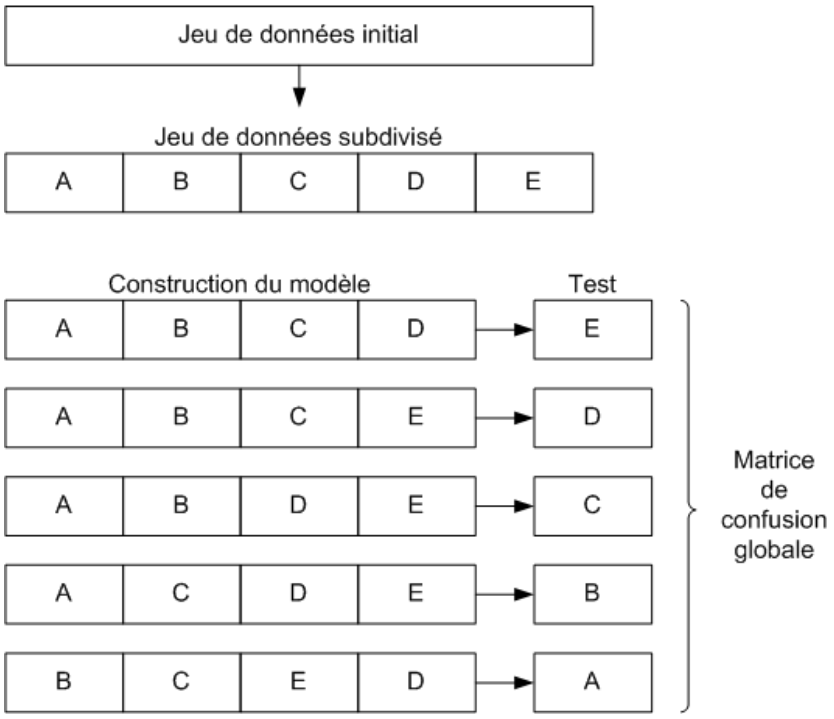


FIG. 4.1 – Principe d'une validation croisée à 5 subdivisions.

Plus le nombre de subdivisions est important, plus les différents modèles construits utilisent d'individus simultanément, et se rapprochent donc d'un modèle construit sur la totalité des individus. L'inconvénient devient alors le nombre de modèles à construire (égal au nombre de subdivisions), et les temps

de calcul associés. Notons que la configuration extrême de la validation croisée où chaque subdivision n'est constituée que d'un seul individu se nomme le *"leave-one-out"*.

4.2 Déséquilibre et notion de symétrie

Un jeu de données déséquilibrées est un problème d'apprentissage supervisé où les effectifs des différentes modalités d'une variable endogène discrète sont très différents. Il s'agit d'une problèmatique relativement récente apparue dès lors que le data mining est devenu une technologie amplement utilisée dans l'industrie, dans des exemples réels comme le diagnostic des maladies de la thyroïde [MA94], la gestion des défauts des boîtes de vitesses des hélicoptères [JMG95], la détection de fraudes téléphoniques [FP97], ou encore la recherche de gisements de pétrole sur des images satellites [KHM98], etc. Comme le notent Florian Verhein et Sanjay Chawla [VC07] "dans des applications comme le diagnostic médical ou la détection de fraudes, [les] jeux de données déséquilibrées sont la norme et non l'exception". Si le déséquilibre est un problème pour la production de modèles, nous allons voir que certaines des problématiques qui lui sont liées impactent directement l'évaluation des performances.

La plupart des algorithmes d'apprentissage supervisé sont basés sur deux hypothèses : (1) le critère à minimiser est le nombre d'erreurs et (2) le jeu de données d'apprentissage est un échantillon représentatif de la population sur laquelle le modèle sera appliqué. Or, dans le cas d'un jeu de données déséquilibré, il est difficile de répondre correctement à la première de ces hypothèses. Par exemple, si 99% des données appartiennent à une seule classe, il sera difficile de faire mieux que le 1% d'erreur obtenu en classant tous les individus dans cette classe. Weiss [Wei04, WP01] propose de distinguer plus précisément les différents problèmes des données déséquilibrées :

1. Métriques inappropriées : Que ce soit pour guider l'apprentissage, ou pour en évaluer les résultats, les mesures utilisées au cours du proces-

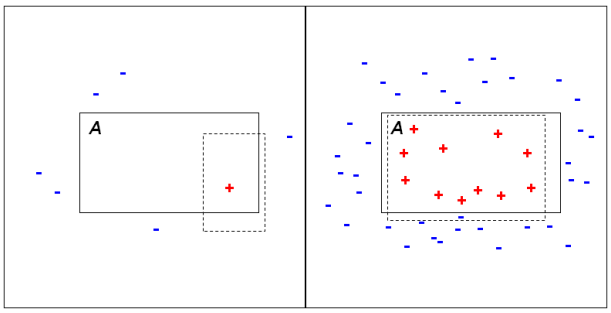


FIG. 4.2 – L'effet d'un manque "absolu" de données.

sus d'apprentissage ne sont pas adaptées aux classes déséquilibrées. En remplaçant le critère à optimiser, par exemple le taux d'erreur, par un critère plus pertinent, on doit pouvoir adapter simplement les algorithmes. Le rappel et la précision sont par exemple plus adéquats.

2. Manque "absolu" de données : Il s'agit du problème principal du déséquilibre : les données disponibles ne sont pas suffisantes pour définir clairement le concept. La figure 4.2 illustre ce principe : dans l'échantillon de la partie gauche les données disponibles concernant le concept A ne sont pas suffisantes pour définir ses frontières, estimées par les rectangles en pointillés. Le concept est beaucoup mieux défini sur l'échantillon de droite, pour lequel plus de données sont disponibles. Ce problème est à mettre en relation avec celui des "cas rares" (*Small disjuncts*) [Wei03].
3. Manque relatif de données : Les objets d'une classe ne sont pas rares au sens absolu, mais beaucoup moins représentés que ceux des autres classes. Le problème est donc le ratio classe majoritaire/classe minoritaire plus que le nombre d'individus disponibles pour apprendre le concept de la classe minoritaire.
4. Marge d'induction : Il s'agit de la marge appliquée à la règle apprise sur les données d'apprentissage pour pouvoir généraliser. On peut considérer la marge de généralité maximum : une fois qu'on a sélectionné un groupe d'individus comme appartenant à un même concept, on définit ce dernier grâce au nombre minimum de conditions qui mènent à ces individus. A l'inverse la marge de spécificité maximum va quant à

elle conserver toutes les règles possibles satisfaites par ces individus. Or la marge de généralité maximum est appropriée aux cas fréquents mais pas aux cas rares. Pourtant de nombreux systèmes d'induction préfèrent la généralité à la spécialisation, favorisant la classe la plus présente en cas d'incertitude.

5. Données bruitées : Le bruit a plus d'impact sur les classes rares que sur les classes fréquentes, tout simplement parce que peu d'individus bruités (mal étiquetés) suffisent pour brouiller le concept à apprendre. Le modèle devient incapable de distinguer le bruit de cas rares du concept. S'il est rendu plus spécifique, il apprendra correctement ces sous-concepts rares, mais également ceux qui sont réellement du bruit.

D'autres problématiques sont fortement liées à l'apprentissage à partir de jeux de données déséquilibrées. Parmi elles, l'asymétrie des coûts nous intéresse particulièrement car elle impacte directement l'évaluation des performances. Cette dernière est liée l'asymétrie des classes en termes d'importance, ou de coûts des erreurs. Ainsi dans l'exemple d'aide au diagnostic médical (voir 1.1), si faire une erreur sur la classe majoritaire (classer comme malade un individu sain) est coûteux en termes d'examens inutiles et de stress pour le patient, faire une erreur sur la classe minoritaire (ne pas détecter la maladie chez un patient) est bien plus grave. Cette différence en terme d'importance de chaque modalité n'est pas prise en compte ni par les systèmes d'apprentissage classiques, ni par les critères d'évaluation usuels.

Ce problème est lié à celui du déséquilibre, car bien souvent les classes rares sont les plus importantes, et les erreurs sur ces dernières sont plus coûteuses.

4.3 Indicateurs

4.3.1 Taxonomie des critères d'évaluation

Plusieurs critères d'évaluation de la qualité d'un modèle d'apprentissage supervisé sont établis dans la littérature. Il est possible de les répertorier en sept catégories principales :

1. Critères basés sur la précision, comme par exemple le taux de correction globale, les taux de précision et de rappel, la $F - mesure$, ou d'autres critères basés sur l'analyse de la matrice de confusion. Ce sont les critères principaux et les plus couramment utilisés.
2. Critères basés sur l'entropie, comme par exemple l'entropie croisée, le gain d'entropie [DMBMB06], ou la mesure de divergence dirigée.
3. Complexité du classifieur (par exemple longueur maximale, nombre de noeuds, nombre de feuilles dans le cas d'un arbre de décision).
4. Interprétabilité du modèle de classification. Ce critère est assez subjectif. Les techniques d'arbres de décision sont réputées pour leur interprétabilité. Un classifieur simple est souvent plus interprétable.
5. Vitesse : à la fois le temps nécessaire pour la construction du classifieur et pour classer un exemple. Dans un cadre industriel, il n'est pas rare que seule la vitesse pour classer un exemple (vitesse de l'application) compte.
6. Robustesse : la sensibilité de la méthode par rapport à des modifications mineures de la base d'apprentissage. Cette capacité permet de résister au bruit présent dans les données.
7. Capacité de passage à l'échelle.

Parmi ces critères, les 5 premiers concernent les modèles de classification. Les 3 derniers concernent les méthodes de construction de modèles. La vitesse concerne à la fois les méthodes de construction de modèles et les modèles eux-mêmes. Une liste complète des mesures avec leurs descriptions, ainsi que

l'étude empirique de ces mesures se trouvent, entre autres, dans les travaux de Caruana [CNM04, CNM06].

Notre objectif étant d'améliorer et de mettre en corrélation avec les besoins de l'utilisateur, les performances brutes du modèle, c'est-à-dire ses performances de classification issues de l'analyse de la matrice de confusion, nous ne nous intéresserons en détail qu'à la première catégorie. C'est également dans cette catégorie que s'inscrit la mesure que nous proposerons en chapitre 7.

4.3.2 Mesures de performance

la matrice de confusion contient toutes les informations sur la classification réelle et la classification prédite faite par un modèle de classification. Il s'agit d'une table de contingence confrontant les classes prédites par le modèle (en colonnes) et les classes réelles (en lignes) pour les individus du jeu de données. Dans le tableau 4.1, N_{ij} est le nombre d'individus de la classe C_i classés par le modèle dans la classe C_j. La performance d'un modèle de classification est évaluée en se basant sur les informations figurant dans cette matrice. En réduisant cette matrice de n^2 éléments à une seule valeur numérique pour évaluer la performance d'un classifieur, on perd la richesse d'information donnée par la matrice. Mais cela est nécessaire, car d'une part cela donne une information plus synthétique et plus interprétable, et d'autre part cela sert à la comparaison entre les classifieurs. Il est donc nécessaire de caractériser les mesures d'évaluation pour déterminer celles qui s'adaptent au mieux au problème de classification considéré.

Le taux de correction globale, noté *acc*, est le critère le plus utilisé. Il est défini comme le rapport entre le nombre d'individus correctement classifiés et l'effectif total :

$$acc = \frac{\sum_{i=1}^{i=n} N_{ii}}{\sum_{i=1}^{i=n} \sum_{j=1}^{j=n} N_{ij}}$$

Classe réelle \ **Classe prédite**	C_1	C_2	...	C_n
C_1	N_{11}	N_{12}	...	N_{1n}
C_2	N_{21}	N_{22}	...	N_{2n}
...	...	...	...	...
C_n	N_{n1}	N_{n2}	...	N_{nn}

TAB. 4.1 – Matrice de confusion issue de la prédiction d'un problème à n classes

On en déduit directement le taux d'erreur globale, noté err, défini par :

$$err = 1 - acc$$

L'objectif des méthodes de construction de modèles de classification est de maximiser le taux de correction globale (ou minimiser le taux d'erreur). L'utilisation de cet indice global, c'est-à-dire prenant en considération l'ensemble des classes, suppose que le coût de mauvaise classification est égal pour tous les individus. Cela rend impossible l'utilisation du taux de correction, ou d'erreur, dès lors que cela n'est plus vrai. C'est la cas dans notre problème de jeux de données déséquilibrées où l'oubli d'un individu de la classe d'intérêt (comme un cancer dans notre application industrielle) est beacoup plus grave, que l'insertion d'un individu de la classe majoritaire comme prédit de la classe minoritaire (présence d'une fausse alarme). Une analyse plus détaillée du taux de correction globale est présentée dans [PFR98, KB91].

Dans l'état de l'art, on considère souvent le cas à deux classes, car il s'agit d'un cas plus simple à expliquer. Dans la réalité, ce cas apparaît plus souvent que les autres. Dans la suite, nous présentons les mesures dans le cas à deux classes. Ces mesures, en général relatives à une classe (d'intérêt ou positive) contre l'ensemble des autres classes (dites négatives), peuvent être étendues pour les cas à trois classes ou plus. Nous utiliserons C_1 comme classe négative, et C_2 comme classe positive.

Le taux de rappel (ou taux de vrai positifs ou sensibilité) exprime la

probabilité de bien classer un individu de la classe positive :

$$r = \frac{N_{22}}{N_{21} + N_{22}}$$

Le taux de vrais négatifs (ou spécificité) exprime la probabilité de bien classer un individu de la classe négative :

$$\frac{N_{11}}{N_{11} + N_{12}}$$

Le taux de faux positifs exprime la probabilité de mal classer un individu de la classe négative :

$$\frac{N_{12}}{N_{11} + N_{12}}$$

Le taux de faux négatifs exprime la probabilité de mal classer un individu de la classe positive :

$$\frac{N_{21}}{N_{21} + N_{22}}$$

Ces quatre mesures sont en fait des probabilités conditionnées par les classes réelles des individus : le taux de vrais positifs et le taux de faux négatifs sont estimés sur les individus de la classe positive ; le taux de vrais négatifs et le taux de faux positifs sont estimés sur les individus de la classe négative. Les deux premières sont à maximiser et les deux dernières sont à minimiser.

Les quatre mesures suivantes sont également des probabilités conditionnées mais par les classes prédites des individus. Le taux de précision exprime la probabilité de bien classer un individu parmi ceux prédits comme positifs :

$$p = \frac{N_{22}}{N_{12} + N_{22}}$$

La valeur prédite négative exprime la probabilité de bien classer un individu parmi ceux prédits comme négatifs :

$$\frac{N_{11}}{N_{11} + N_{21}}$$

La probabilité de mal classer un individu parmi ceux prédits comme négatifs (*prediction-conditionned fallout*) :

$$\frac{N_{21}}{N_{11} + N_{21}}$$

La probabilité de mal classer un individu parmi ceux prédits comme positifs (*negative-conditioned miss*) :

$$\frac{N_{12}}{N_{12} + N_{22}}$$

Parmi les 4 mesures ci-dessus, les deux premières (le taux de précision et la valeur prédite négative) sont à maximiser et les deux dernières à minimiser. Les taux de précision et de rappel sont très utilisés en recherche d'information, même s'il ne juge de la qualité du modèle que vis-à-vis d'une seule classe.

La mesure la plus utilisée basée sur le taux de rappel (r) et le taux de précision (p) est la *F-measure* [RCB+79] définie par :

$$F\text{-}measure = \frac{2 \times r \times p}{r + p}$$

Elle combine les taux de rappel et précision sous forme d'une moyenne harmonique pour donner une mesure de qualité de la prédiction du classifieur sur la classe d'intérêt, mais elle ne permet pas de privilégier l'un des taux vis-à-vis de l'autre. La même importance est donc accordée aux faux positifs et aux faux négatifs, ce qui est rarement le cas en réalité où les faux-négatifs sont plus 'grave' dès lors que l'objectif prioritaire est la détection des individus d'intérêt. La F_β-*measure* corrige cette situation en ajoutant un paramètre β :

$$F_\beta\text{-}measure = \frac{(1 + \beta^2) \times r \times p}{\beta^2 \times r + p}$$

Celle-ci permet de jouer sur l'importance des deux taux au sein de la mesure mais conserve deux inconvénients :

1. Elle privilégie des valeurs équilibrées entre le taux de rappel et le taux de précision. Ainsi, des taux de rappel et précision faibles mais de hauteur équivalente conduisent malgré tout à une mesure importante.

2. Le paramètre β est très peu intuitif. La traduction des besoins de l'utilisateur, en terme de taux de rappel et de taux de précision, en un paramétrage de β est floue.

La moyenne géométrique g [KHM97] est également basée sur la notion de taux de rappel, mais l'utilise pour créer une mesure de performances globale sur l'ensemble des classes :

$$g = \sqrt[M]{\prod_{m=1}^{m=M} r_m}$$

où M est le nombre de classe et r_m le taux de rappel de la classe m. Il s'agit d'une mesure régulièrement utilisée en situation de déséquilibre puisqu'elle est indépendante des effectifs de chaque classe. Cependant, elle accorde une importance équivalente à chacune d'elle.

Le rapport de cotes (*odds-ratio*), est une grandeur statistique permettant de comparer deux facteurs de risques différents dans une population, dans notre contexte, il faut le maximiser. Il est défini comme suit :

$$\frac{\frac{N_{11}}{N_{12}}}{\frac{N_{21}}{N_{22}}} = \frac{N_{11} \times N_{22}}{N_{12} \times N_{21}}$$

Le coefficient Kappa mesure une corrélation entre deux variables statistiques. En apprentissage, il est utilisé pour évaluer l'accord entre la distribution naturelle et la distribution issue de la classification, ceci en prenant également en compte la distribution obtenable par chance. Il est considéré comme une amélioration du taux de correction globale. Comme la base contient $(N_{11} + N_{12})$ individus de la classe négative et $(N_{22} + N_{21})$ individus de la classe positive, lorsque l'on classifie $(N_{11} + N_{21})$ (respectivement $(N_{12} + N_{22})$) individus dans la classe négative (respectivement positive) de manière aléatoire, l'espérance du nombre d'individus correctement classifiés dans la classe négative (respectivement positive) est, N étant l'effectif total :

$$\frac{N_{11} + N_{12}}{N} \times (N_{11} + N_{21})$$

respectivement :

$$\frac{N_{21} + N_{22}}{N} \times (N_{12} + N_{22})$$

L'espérance du nombre d'exemples bien classifiés par chance est donc :

$$\frac{(N_{22} + N_{21}) \times (N_{22} + N_{12}) + (N_{11} + N_{12}) \times (N_{11} + N_{21})}{N}$$

On en déduit la contribution effective du classifieur :

$$(N_{11} + N_{22}) - \frac{(N_{22} + N_{21}) \times (N_{22} + N_{12}) + (N_{11} + N_{12}) \times (N_{11} + N_{21})}{N}$$

Le coefficient Kappa, introduit par J. Cohen [Coh60], évalue un taux de bonnes classifications et exclut la partie des bonnes classifications obtenue par chance :

$$Ka = \frac{(N_{11} + N_{22}) - \frac{(N_{22}+N_{21})\times(N_{22}+N_{12})+(N_{11}+N_{12})\times(N_{11}+N_{21})}{N}}{N - \frac{(N_{22}+N_{21})\times(N_{22}+N_{12})+(N_{11}+N_{12})\times(N_{11}+N_{21})}{N}}$$

Ce coefficient est toujours inférieur ou égal à 1 et on souhaite le maximiser. Un coefficient Kappa négatif signifie que le classifieur est pire qu'un classifieur aléatoire. La définition de cette mesure peut s'étendre facilement aux cas à plusieurs classes. Il peut également être utilisé pour la sélection d'attribut [LN05]. Cependant s'il peut être vu comme une amélioration du taux de correction globale, il en garde l'un des principaux défauts, à savoir que ce dernier suppose que le coût de mauvaise classification est égal pour tous les individus quelque soit leur classe.

Le coefficient de corrélation, aussi appelé le coefficient de corrélation de rang partiel de Kendall (*Kendall partial rank correlation*) est une autre forme d'amélioration du taux de correction globale :

$$Ke = \frac{N_{22} \times N_{11} - N_{12} \times N_{21}}{\sqrt{(N_{22} + N_{21}) \times (N_{11} + N_{12}) \times (N_{12} + N_{22}) \times (N_{11} + N_{21})}}$$

Ce coefficient prend ses valeurs entre -1 et 1. Une classification parfaite correspond à 1 et une classification aléatoire correspond à 0. Une valeur négative signifie une classification pire que la classification aléatoire. Celui-ci ne permet toujours pas la prise en compte d'une asymétrie de coût.

Le $Lift$ [HBPJ96] est souvent employé dans l'analyse marketing. Elle mesure combien de fois le classifieur en question est meilleur sur une sous-population par rapport à un classifieur aléatoire, cela pour une classe donnée :

$$Lift = \frac{taux\ de\ rappel\ sur\ une\ sous-population}{taux\ d'individus\ positif\ dans\ la\ population\ totale}$$

Cette mesure est intéressante en marketing, par exemple quand on souhaite cibler une publicité sur un nombre limité de personnes. Le $Lift$ permet d'évaluer la sous-population qui maximisera la qualité du ciblage.

L'erreur carrée (*root mean squared error* ou $RMSE$) est souvent utilisée en régression (où la variable endogène prend ses valeurs sur un segment continu). Cette mesure est cependant applicable en classification binaire où les 2 classes sont ramenées aux valeurs 0 et 1. Elle est définie par :

$$RMSE = \sqrt{\frac{1}{N}\sum(ClassePredite(individu) - ClasseReelle(individu))^2}$$

Cette mesure globale, issu du domaine des probabilités, conserve néanmoins les inconvénients dûs à la non prise en considération d'une éventuelle asymétrie des coûts en fonction des classes.

Chapitre 5

Espace des individus

5.1 Introduction

L'apprentissage à partir de données déséquilibrées nécessite des stratégies adaptées pour obtenir une classification correcte de la classe minoritaire. L'évantail des solutions proposées vont de l'échantillonnage à la construction d'un modèle de prédiction spécifique de la classe d'intérêt (*one class learning*) en passant par l'introduction d'un biais dans le critère optimisé par l'apprentissage pour prendre en compte le déséquilibre [PMM+94, KHM98, BSGR03]. Pour notre application (aide au diagnostic du cancer du sein), le modèle de prédiction doit se satisfaire aux contraintes de qualité inhérentes au domaine médical. Dans ce chapitre, nous passons en revue des techniques d'échantillonnage utilisées pour contourner les problèmes liés aux données déséquilibrées, puis nous proposons une stratégie fondée sur l'échantillonnage qui intègre les besoins de l'utilisateur (ici en terme de taux de rappel pour répondre à l'asymétrie des coûts) dans l'orientation des performances du classifieur.

5.2 Etat de l'art

5.2.1 Sur et sous-échantillonnage

Le ré-échantillonnage des données est une idée fréquemment exploitée pour enrayer l'impact du déséquilibre sur l'apprentissage. Dans nombre de cas, l'objectif est de rétablir l'équilibre entre les effectifs, soit en réduisant celui de la classe majoritaire (sous-échantillonnage) soit en augmentant celui de la classe minoritaire (sur-échantillonnage).

Dans le cas du sous-échantillonnage aléatoire, aucune attention n'est prêtée aux individus de la classe majoritaire sélectionnés pour l'apprentissage. Ses principaux avantages résident ainsi dans sa simplicité de mise en oeuvre et sa rapidité d'exécution [Jap00]. Cependant, il est souvent argué que le sous-échantillonnage induit une perte d'informations. Plusieurs auteurs ont donc cherché à diriger la sélection des individus. Dans ce sens, [KM97] enlèvent des individus redondants parmi les individus majoritaires (qui ralentissent l'apprentissage sans en augmenter la qualité) et les individus à la frontière de décision entre les classes (qui dégradent l'apprentissage). Les premiers sont identifés via la constitution d'un sous-ensemble consistant des données d'apprentissage (un sous-ensemble C de S est consistant s'l permet la prédiction de S avec la règle 1-NN [Har68]), alors que les seconds sont détectés au travers des liens de Tomek (*Tomek links*, [Tom76]) : il existe un lien entre les individus x et y si il n'existe pas d'individu z plus près de x (resp. y) que ne l'est y (resp. x). Une autre solution consiste à appliquer un algorithme d'édition à l'effectif de la classe majoritaire. [BVSF04] utilisent pour ce faire l'édition de Wilson (*Wilson Editing*, [Wil72]) qui consiste à appliquer un classifieur de type k-NN aux données d'apprentissage et à en supprimer les individus mal étiquetés. Cependant, comme la règle k-NN est également soumise au problème du déséquilibre, ils biaisent la mesure de distance en la pondérant en fonction de la classe de l'exemple considéré. Ainsi, la distance à des individus de classe minoritaire diminue par rapport à la distance à des individus de classe majoritaire.

La solution opposée au sous-échantillonnage est le sur-échantillonnage. Dans sa version aléatoire, il s'agit de reproduire des individus de la classe minoritaire tirés au sort pour atteindre des effectifs équilibrés. Si la technique est simple à mettre en oeuvre, elle court le risque d'entrainer un sur-apprentissage des données en forçant le classifieur à s'intéresser à des régions très étroites de l'espace. Pour éviter cet écueil, [CBHK02] proposent SMOTE (*Synthetic Minority Oversampling Technique*), une technique de sur-échantillonnage qui génère des individus synthétiques afin d'étendre les limites de la classe minoritaire. Elle procède de la manière suivante. Les k plus proches voisins de chaque individu positif X sont récupérés. Parmi eux sont tirés au sort suffisamment d'individus pour atteindre le taux de sur-échantillonnage désiré. Un nouvel individu N est alors généré pour chacun de ces voisins V. Son $i^{\text{ème}}$ attribut prend pour valeur : $N_i = X_i + (V_i - X_i) \times \text{Rand}(0,1)$ où $\text{Rand}(0,1)$ représente un nombre aléatoire uniforme entre 0 et 1.

Plusieurs auteurs ont étudié l'efficacité de ces approches. Dernièrement, [HKN07] en réalisent une comparaison exhaustive. Ils concluent à la supériorité du sous-échantillonnage lorsque les données sont raisonnablement déséquilibrées ($\geq 10\%$) et à celle du sur-échantillonnage dans le cas contraire (conclusion partagée avec [BVSF04]). Par ailleurs, cette étude plaide en faveur des méthodes simples d'échantillonnage puisqu'elle ne note aucune amélioration significative des performances pour les méthodes dirigées, méthodes qui, de plus, sont coûteuses (en accord avec [Jap00]).

5.2.2 Echantillonnage et ensemble

Dans les exemples décrits ci-dessus, l'échantillonnage ne vise qu'à contourner le problème du déséquilibre pour rétablir une situation "normale". Pourtant, l'échantillonnage peut aussi servir à améliorer l'apprentissage, comme le montre les travaux réalisés sur les ensembles.

L'échantillonnage des individus, lors de la construction d'un ensemble, permet de créer de la diversité entre chaque classifieur de base, et cela afin

d'améliorer les résultats de l'ensemble grâce à des "opinions" individuelles plus indépendantes. Des études ont cherché à profiter de l'approche ensembliste pour gérer le problème du déséquilibre. Barandela et al. [BSGR03] proposent d'apprendre les classifieurs d'un ensemble sur des échantillons équilibrés constitués de tous les individus de la classe minoritaire et d'un échantillon aléatoire d'individus de la classe majoritaire. Pour ces auteurs, l'idée est que l'équilibre entre les classes permet d'optimiser la prédiction. Cependant, certains travaux montrent qu'il n'en est pas toujours ainsi [DK01, WP03]. Pour ajuster la distribution des classes au problème, une première solution consiste à en essayer plusieurs pour choisir la plus adaptée [PAL04]. Ainsi, ces auteurs proposent un processus en 2 étapes : (1) choix de la meilleur distribution puis (2) choix du meilleur classifieur pour le bagging. Si elle est efficace, cette stratégie est coûteuse en individus et en temps puisque l'évaluation conjointe de la distribution et des classifieurs demande à la fois de réserver des individus et de construire des classifieurs qui ne serviront pas à la prédiction. Une stratégie plus fine consiste à adapter le boosting au problème du déséquilibre. Dans leur méthode *DataBoost-IM*, [GV04] séparent les individus difficiles à apprendre (dits graines, individus pondérés au cours du boosting) suivant leur classe d'appartenance. Pour chaque groupe de graines (et suivant la distribution entre ces groupes), de nouveaux individus typiques de la classe correspondante sont générés et intégrés à l'échantillon à apprendre. Cette stratégie permet ainsi d'adapter automatiquement la distribution entre les classes au problème en cours tout en bénéficiant de l'approche ensembliste.

5.3 FUNSS : une approche guidée par le coût

Les travaux existants nous ont amené à faire plusieurs constats. D'abord, lorsque le déséquilibre est élevé, le sur-échantillonnage est plus adapté que le sous-échantillonnage. Cependant, il est nécessaire de veiller aux régions de l'espace qui seront privilégiées pour ne pas entraîner de sur-apprentissage. Inversement le sous-échantillonnage n'est efficient que si le déséquilibre est limité, car sinon la perte d'informations vis-à-vis de la classe majoritaire est

trop importante. Ensuite, la meilleure distribution entre les classes dépend du problème : un échantillonnage équilibré et aléatoire n'optimise pas forcément le taux d'erreur. Enfin, les méthodes ensemble sont un moyen efficace de tirer profit de l'échantillonnage. A ces constats il est également intéressant de remarquer que les stratégies d'échantillonnage sont rarement reliées aux besoins de l'utilisateur, en particulier sur le compromis rappel/précision pour la classe d'intérêt. Or, notre domaine d'application est justement un domaine où il est intéressant d'intégrer ces préférences pour optimiser la prédiction. C'est pourquoi nous proposons FUNSS (*Fitting User Needs Sampling Strategy*), une approche originale pour traiter le déséquilibre entre les classes sans chercher à uniformiser les effectifs.

5.3.1 Principe

L'idée de FUNSS est de traduire le compromis rappel/précision en terme de marge d'induction entre les individus de chaque classe. Les individus de la classe minoritaire (ou individus positifs) sont entourés par une quantité importante d'individus des classes majoritaires (ou individus négatifs) qui empèchent le classifieur de les apprendre correctemment et entraîne un faible taux de rappel pour la classe minoritaire. Pour augmenter le taux de rappel, une solution consiste à choisir des individus négatifs éloignés des individus positifs (c'est-à-dire à augmenter la marge d'induction des règles les visant). Et à l'inverse, pour augmenter le taux de précision, il suffit de garder les individus négatifs proches des individus positifs. Nous allons présenter FUNSS en modifiant l'échantillonnage réalisé au cours des forêts aléatoires en un échantillonnage dirigé, notons que FUNSS peut s'appliquer avec n'importe quel classifieur de base dès lors qu'un bagging (3.6.1) est réalisé.

A chaque tirage aléatoire avec remise dans l'échantillon d'apprentissage, le processus est le suivant : Si l'individu est positif, il est directement intégré dans le nouvel échantillon ; Dans le cas contraire, un groupe de k individus négatifs est tiré au hasard ainsi qu'un individu positif. L'individu négatif du groupe qui est soit le plus proche (appelé par la suite mode "proche") soit le

plus éloigné (mode "éloigné") de l'individu positif, est intégré dans le nouvel échantillon. Ainsi, les individus négatifs sont sélectionnés en fonction de leur proximité par rapport aux individus positifs. Si cette proximité est faible, la classe positive sera plus facilement distinguée de la classe négative induisant un rappel plus élevé (et vice-versa). Chaque échantillon de la forêt aléatoire est donc l'occasion d'augmenter ou de diminuer le rappel pour atteindre une valeur fixée par l'utilisateur. Pour cela, le rappel de la forêt est estimé à chaque nouvel arbre à l'aide des individus out-of-bag. S'il est en dessous du rappel désiré, l'échantillonnage suivant sélectionne des individus négatifs éloignés des individus positifs. Dans le cas contraire, les individus négatifs proches sont favorisés (voir figure 5.1).

Le calcul des distances entre individus est un point important de cette stratégie. Cependant, dans les espaces de grandes dimensions, la distance euclidienne perd son pouvoir discriminant. Ainsi, la distance d'un point à son plus proche voisin tend à être égale à la distance de ce même point à son voisin le plus éloigné quand le nombre de dimensions tend vers l'infini (sous les conditions d'indépendance entre ces dimensions, [BGRS99]). Pour contourner ces problèmes, une distance de rang est utilisée. Pour déterminer l'individu le plus proche d'une cible, les individus sont ordonnés pour chaque attribut sur leur proximité à cette cible. La distance est alors la somme des rangs d'un individu : plus cette somme est élevée, plus l'individu est loin de la cible. Cette distance conduit à des distributions identiques sur chaque attribut. La figure 5.2 présente un exemple de sélection d'un individu de la classe majoritaire lors d'un échantillonnage.

5.3.2 Expérimentations

La figure 5.3 a pour but d'illustrer l'effet des deux modes, proche ou éloigné, de sélection des individus lors de l'échantillonnage. Pour réaliser ces graphiques, une analyse en composante principale (ACP) a été réalisée sur la totalité du jeu de données Satimage [HB99] dont la composition est détaillée table 5.1.

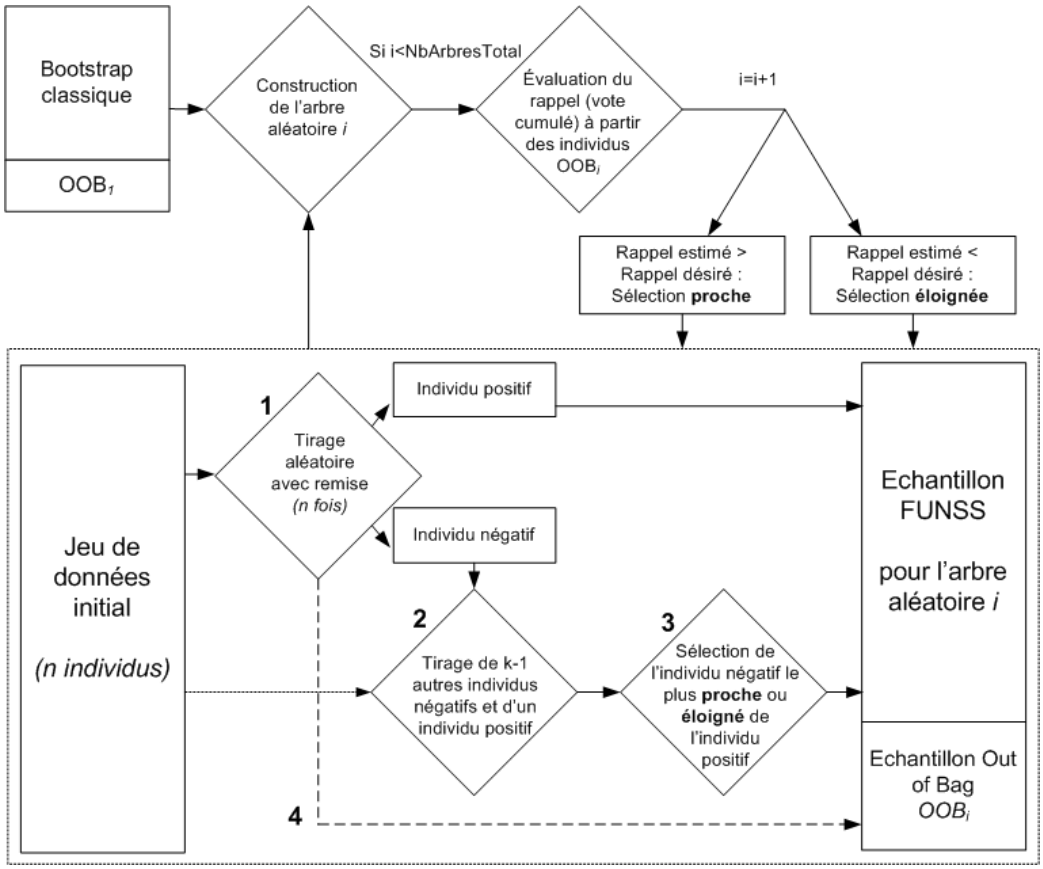
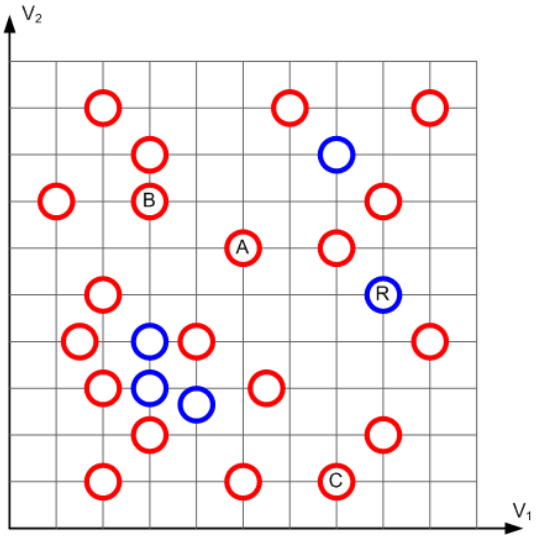


FIG. 5.1 – Principe de construction d'une forêt aléatoire utilisant l'échantillonnage FUNSS.

5 individus minoritaires en bleu, R est tiré en référent.
20 individus majoritaires en rouge, A B et C sont les trois concurrents tiré au sort.
A est 2e en proximité de R selon V1 et 1er selon V2.
La distance d entre A et R est donc :
$d(A, R) = 2 + 1 = 3$
De même pour B et C :
$d(B, R) = 3 + 2 = 5$
$d(C, R) = 1 + 3 = 4$



A serait sélectionné en mode « proche ».
B serait sélectionné en mode « éloigné ».

FIG. 5.2 – Exemple de sélection d'un individu de la classe majoritaire lors d'un échantillonnage FUNSS.

	Effectif	Variables exogènes	Effectif positif	Fréquence classe positive
Satimage	6435	36	626	9,73%
MammoClusters	5977	378	1779	29,76%

TAB. 5.1 – Composition des jeux de données déséquilibrées Satimage et MammoClusters.

Les deux premiers axes de cette ACP ont ensuite été conservés comme plan de projection pour la visualisation des différents échantillons créés. Ces derniers sont aux nombres de trois : (A) échantillon FUNSS en mode éloigné, son but est de permettre au modèle de gagner en rappel sur la classe positive. Des groupes de 10 individus négatifs ont été utilisés lors de l'étape de sélection. (B) échantillon bootstrap classique. (C) échantillon FUNSS en mode proche, son but est de permettre au modèle de gagner en précision

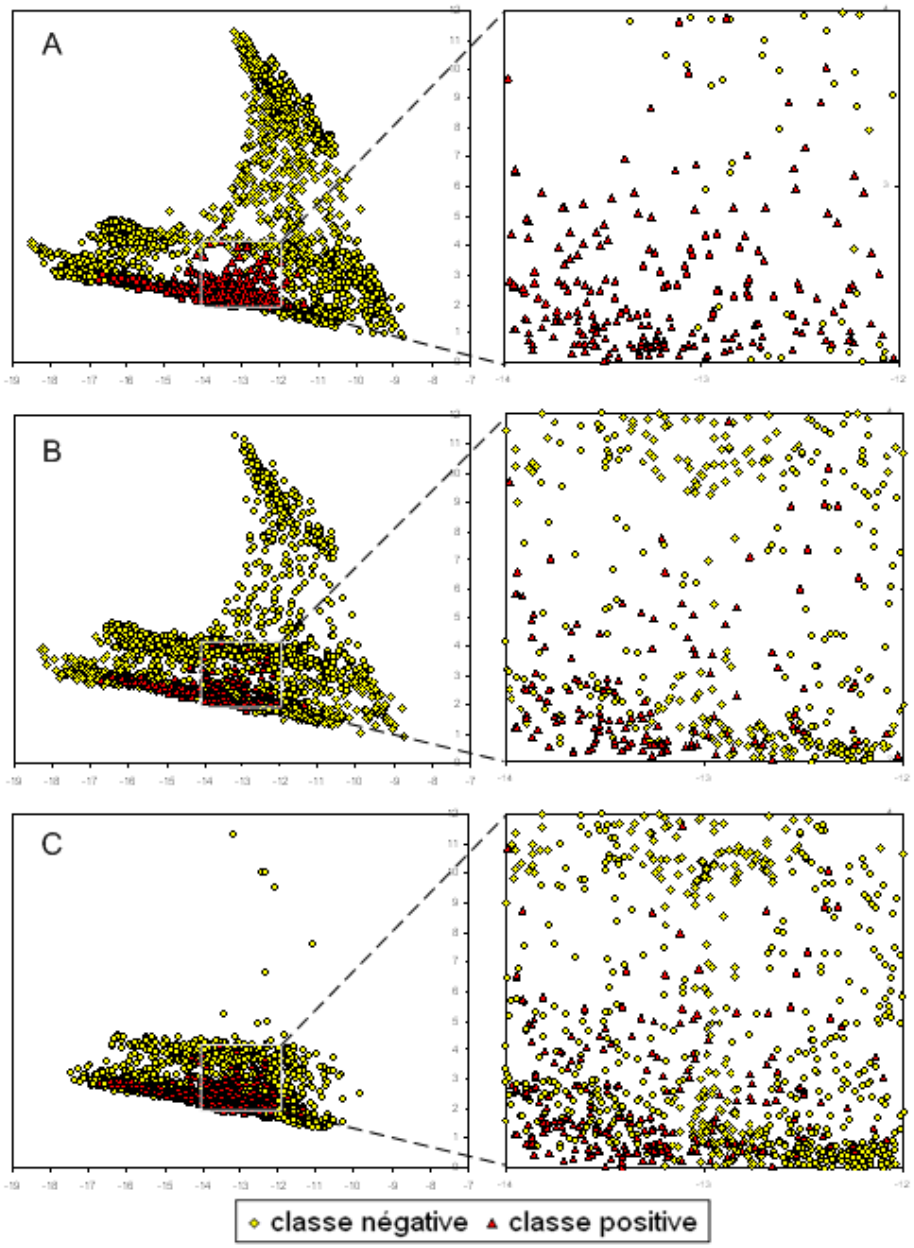


FIG. 5.3 – Visualisation dans le plan formé par les 2 premiers axes ACP de 3 échantillons issus du jeu Satimage : (A) FUNSS mode "éloigné", (B) bootstrap, (C) FUNSS mode "proche".

sur la classe positive. Ce dernier a été construit avec les mêmes paramètres que l'échantillon A. Pour chacune de ces visualisations un agrandissement sur une zone mixte (contenant à la fois des individus positifs et négatifs) est présenté.

Il n'est pas possible de commenter ces illustrations en terme de marge d'induction, entre autre à cause du phénomène de projection. Cependant on notera les différences en terme de rapport d'effectifs dans la zone mixte. Le bootstrap classique (B) présente un rapport proche de l'équilibre, c'est d'ailleurs en ce sens que la zone a été choisie. Les échantillons FUNSS présentent des rapports inverses l'un de l'autre : (A) la volonté de sélectionner des individus négatifs "éloignés" fait décroitre leur nombre dans la zone mixte, ce qui doit logiquement faciliter la détection des individus positifs dans cette zone pour l'arbre construit sur cet échantillon et donc faire augmenter le rappel de la forêt pour la classe positive. (B) inversement la volonté de sélectionner des individus négatifs "proches" augmentent leur nombre en zone mixte et force le classifieur à être plus précis dans sa détection des individus positifs quitte à perdre en capacité de rappel sur cette même classe.

L'évolution du rappel estimé de la forêt pour la classe positive après chaque construction d'arbre est présentée en figure 5.4. Ce rappel est calculé pour la classe positive, à partir des individus out-of-bag (c'est-à-dire des individus de l'échantillon d'apprentissage non présents dans l'échantillon de construction de l'arbre). Les données utilisées pour le graphique sont issues d'une validation croisée composée de 5 subdivisions sur le jeu Satimage. La forêt aléatoire utilisée comporte 50 arbres avec une randomisation sur 6 variables. L'échantillonnage FUNSS est paramétré de la manière suivante : objectif d'un rappel de 80% pour la classe positive et groupes de 10 individus négatifs lors de la sélection. Les points correspondants au premier arbre ont été tronqués pour une meilleure lisibilité du reste du graphique, leurs valeurs en rappel pour la classe positive étaient toutes comprises entre 45 et 55%.

On note que l'oscillation autour de la valeur de rappel souhaitée est de plus en plus faible au cours de la construction de l'arbre, un nouveau vote de la part d'un arbre ayant de moins en moins d'impact sur la prédiction de

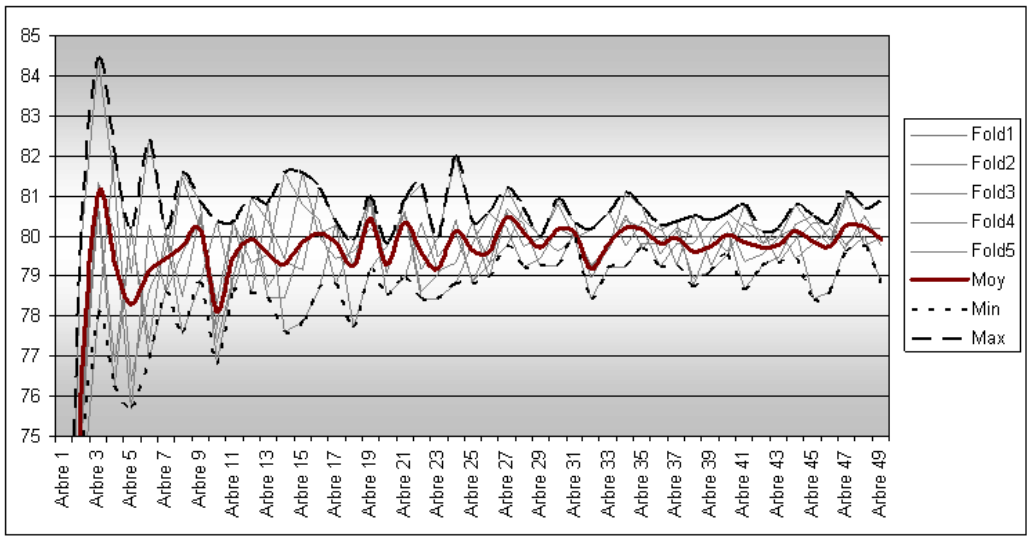


FIG. 5.4 – Evolution du rappel estimé lors de la construction d'une forêt aléatoire utilisant l'échantillonnage FUNSS80. Résultats issus d'une 5-CrossValidation sur le jeu Satimage.

la forêt au fur et à mesure que le nombre d'arbres déjà construits augmente.

Des résultats en terme de performance sont présentés en table 5.2. Il s'agit de tests réalisés sur le jeu de données Satimage. Tous les résultats sont issus de validations croisées composées de 5 subdivisions, pour des forêts aléatoires de 50 arbres, utilisant 6 variables pour la randomisation. La version de SMOTE ([CBHK02], voir 5.2.1) testée ne comporte pas de sous-échantillonnage de la classe négative. L'algorithme BRF (Balanced Random Forest) construit une forêt aléatoire à partir de bootstraps équilibrés [CLB04]. Les temps indiqués proviennent de l'utilisation de la plateforme de data mining interne à la société Fenics (LearnIt, voir 1.1), les indices de temps sont calculés sur la base de 1 pour une forêt aléatoire classique. La notation FUNSSXX signifie que l'algorithme FUNSS a été paramétré avec une barre de rappel désiré pour la classe positive à atteindre de $XX\%$. Les autres paramètres sont les mêmes que ceux utilisés pour la figure 5.4.

Ces résultats montrent qu'aucun algorithme n'est supérieur aux autres sur

Algorithme	R classe positive	R classe négative	P classe positive	Correction globale	Temps (s)	Indice temps
RF	52,1	98,9	83,2	94,3	828	1
BRF	71,6	96,4	68,6	94,0	1782	2,2
SMOTE900	78,4	93,8	57,6	92,3	6528	7,9
FUNSS70	70,1	96,3	67,2	93,8	2077	2,5
FUNSS80	79,6	92,1	52,0	90,9	2406	2,9
FUNSS90	88,0	85,5	39,6	85,7	2251	2,7

TAB. 5.2 – Résultats (en %) de différents algorithmes obtenus en validation croisée (5 subdivisions) sur le jeu Satimage.(R : Rappel ; P : Précision)

l'ensemble des critères retenus. Cependant différentes remarques à propos des résultats de l'algorithme FUNSS peuvent être faites :

1. FUNSS montre la possibilité d'apprendre de manière correcte une modalité minoritaire sans modifier a priori la distribution des effectifs selon la variable endogène.
2. Ses temps de calcul sont sensiblement inférieurs à une technique avancée telle que SMOTE, même si cela ne rentre pas dans nos objectifs.
3. Le rappel de la classe positive obtenu en validation croisée est en adéquation directe avec le souhait émis par l'utilisateur via son paramétrage, ce qui évite de procéder à plusieurs itérations d'apprentissage pour obtenir le type de performances voulues.

5.4 LARSS : vers une adaptation localisée

5.4.1 Adaptations

Dans notre présentation de FUNSS, le but était de construire des classifieurs de base (1) soit spécialisés vers des performances de rappel élevé pour la classe minoritaire, (2) soit vers des performances de précision élevée pour cette même classe. Dans ce but, les individus de la classe minoritaire

étaient utilisés lors de l'échantillonnage (1) soit tous comme des pôles répulsifs lors qu'ils servaient de référent pour le choix de l'individu majoritaire à conserver, (2) soit tous comme des pôles attractifs. Ensuite la composition du bagging entre les classifieurs de base de chaque type évoluait pour obtenir les performances attendues par l'utilisateur.

Nous présentons ici une évolution de l'approche FUNSS, nommée LARSS (*Local Attraction/Repulsion Sampling Strategy*). Le but n'est plus d'obtenir une performance souhaitée par l'utilisateur a priori, mais d'optimiser par échantillonnage l'information exploitable du jeu d'apprentissage initial. Ce que nous entendons par optimisation de l'information exploitable, est l'obtention simultanée d'une précision élevée dans les régions de l'espace de représentation où les individus positifs sont facilement trouvés, et un rappel élevé dans les régions où les erreurs sur la classe minoritaire sont fréquents.

Cette adaptation tente d'intégrer la problématique de variabilité intra-classe, dans la technique d'échantillonnage proposée. Pour ce faire la propriété de pôlarité attractive ou répulsive lors du choix d'un individu négatif dans l'échantillonnage n'est plus globale à l'ensemble des individus positifs mais devient propre à chacun d'entre eux sous la forme d'un coefficient d'attraction α_i. Celui-ci est positif lorsqu'on cherche des individus négatifs proches (pôle d'attraction) et négatif dans le cas contraire (pôle répulsif), il est également borné entre $[-q; q]$ où q est un paramètre utilisateur. Pour assurer une meilleure cohérence dans le choix des individus négatifs, trois référents de la classe minoritaire sont tirés au sort à la place d'un seul dans l'échantillonnage FUNSS.

Le choix d'un individu négatif parmi k concurrents tirés au sort peut se formaliser ainsi :
Soit $\omega_i, i \in \{1; 2; 3\}$, 1 individu parmi 3 référents de la classe minoritaire ;
α_i, le coefficient d'attraction de l'individu ω_i ;
$\omega_j, j \in \{1; ...; k\}$, 1 individu parmi k concurrents de la classe majoritaire ;
$d(\omega_i, \omega_j)$, la somme sur toutes les variables exogènes des rangs de classement de ω_j parmi les k concurrents, du plus proche au plus éloigné de ω_i ;
$F(\omega_j)$, la fonction d'évaluation d'un individu de la classe majoritaire ;

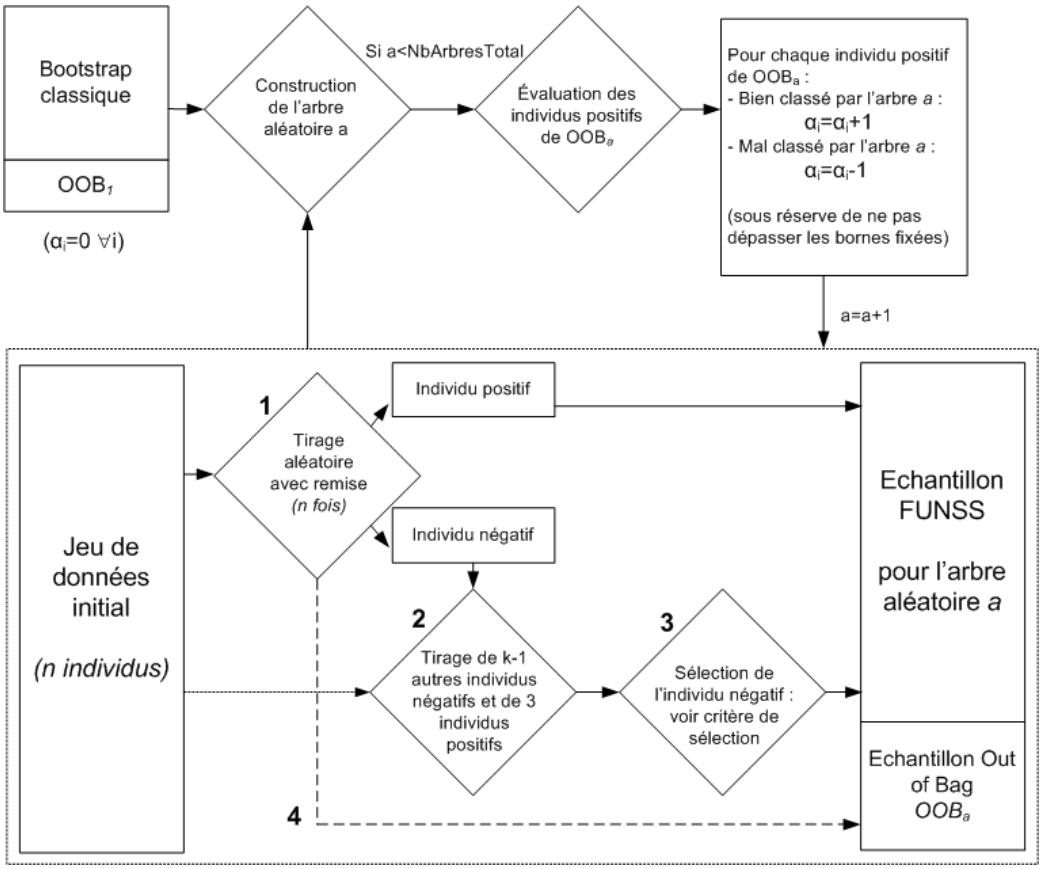


FIG. 5.5 – Principe de construction d'une forêt aléatoire utilisant l'échantillonnage LARSS.

J, l'indice de l'individu de la classe majoritaire sélectionné.

La sélection est alors :
$J = j|F(\omega_j) = \min_{j=1}^{k}(F(\omega_j))$, avec $F(\omega_j) = \sum_{i=1}^{3} \alpha_i \times d(\omega_i, \omega_j)$.

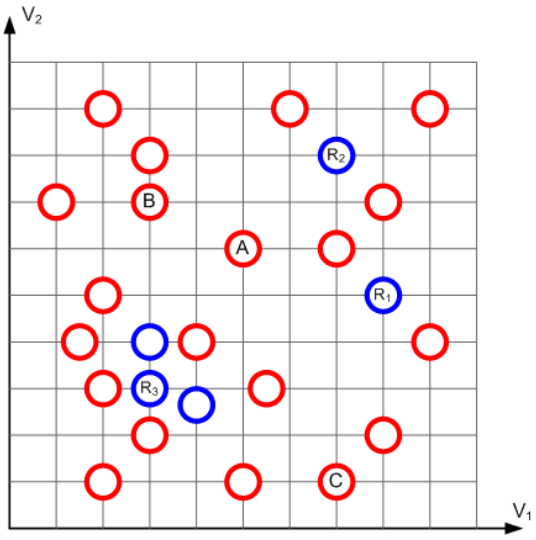
Dans le cas où plusieurs individus négatifs répondent au critère de sélection, un tirage aléatoire parmi eux est effectué. Initialement tous les coefficients d'attraction sont nuls. Ils évoluent à chaque fois que l'individu concerné fait parti de l'échantillon out-of-bag : l'individu est testé à travers l'arbre construit, s'il est correctement classé, son coefficient α_i est augmenté d'une unité (on renforce son effet d'attraction), s'il est mal classé, son α_i est diminué d'une unité (on renforce son effet de répulsion), sous réserve de ne pas dépasser les bornes. Une vision générale du processus de LARSS est donnée en figure 5.5, un exemple de sélection d'un individu de la classe majoritaire est présenté en figure 5.6.

5.4.2 Expérimentations

La figure 5.7 repose exactement sur le même principe et les mêmes paramétrages que la figure 5.3 (voir section 5.3.2). Elle permet de visualiser dans un plan de projection, formé des 2 premiers axes issus d'une analyse par composante principale, les individus de différents échantillonnage opérés sur le jeu de données Satimage. En (A) on retrouve un bootstrap classique et en (B) l'échantillon réalisé pour le 50^e arbre d'une forêt aléatoire construit sur le principe LARSS.

Contrairement à l'expérimentation équivalente sur FUNSS, on observe quasiment aucune différence entre les 2 échantillons, que ce soit sur la totalité de plan ou sur l'agrandissement en zone "mixte". Des différences existent, puisque les performances d'apprentissage changent (voir ci-après), mais celles-ci doivent être suffisamment bien réparties et diverses sur l'ensemble de l'espace de représentation pour disparaître lors de la projection en 2 dimensions. Les arbres ne se spécialisent pas alternativement sur le taux de rappel ou le

R1 R2 et R3 sont les trois référents de la classe minoritaire bleu tirés au sort.
Nous sommes au 10e arbre :
R1 a été 3 fois "Out-of-bag" lors des arbres précédents, mal classé 3 fois.
$\alpha_{R1} = -3$
R2, 2 fois Out-of-bag, mal classé 2 fois.
$\alpha_{R2} = -2$
R3, 4 fois Out-of-bag, bien classé 3 fois.
$\alpha_{R3} = 2$



A B et C sont 3 concurrents tirés au sort de la classe majoritaire.
$d(A, R1) = 2 + 1 = 3$
$d(A, R2) = 2 + 2 = 4$
$d(A, R3) = 2 + 2 = 4$
$F(A) = (-3 \times 3) + (-2 \times 4) + (2 \times 4) = -9$

$d(B, R1) = 3 + 2 = 5$
$d(B, R2) = 3 + 1 = 4$
$d(B, R3) = 1 + 3 = 4$
$F(B) = (-3 \times 5) + (-2 \times 4) + (2 \times 4) = -15$

$d(C, R1) = 1 + 3 = 4$
$d(C, R2) = 1 + 3 = 4$
$d(C, R3) = 3 + 1 = 4$
$F(C) = (-3 \times 4) + (-2 \times 4) + (2 \times 4) = -12$

L'individu B est sélectionné.

FIG. 5.6 – Exemple de sélection d'un individu de la classe majoritaire lors d'un échantillonnage LARSS.

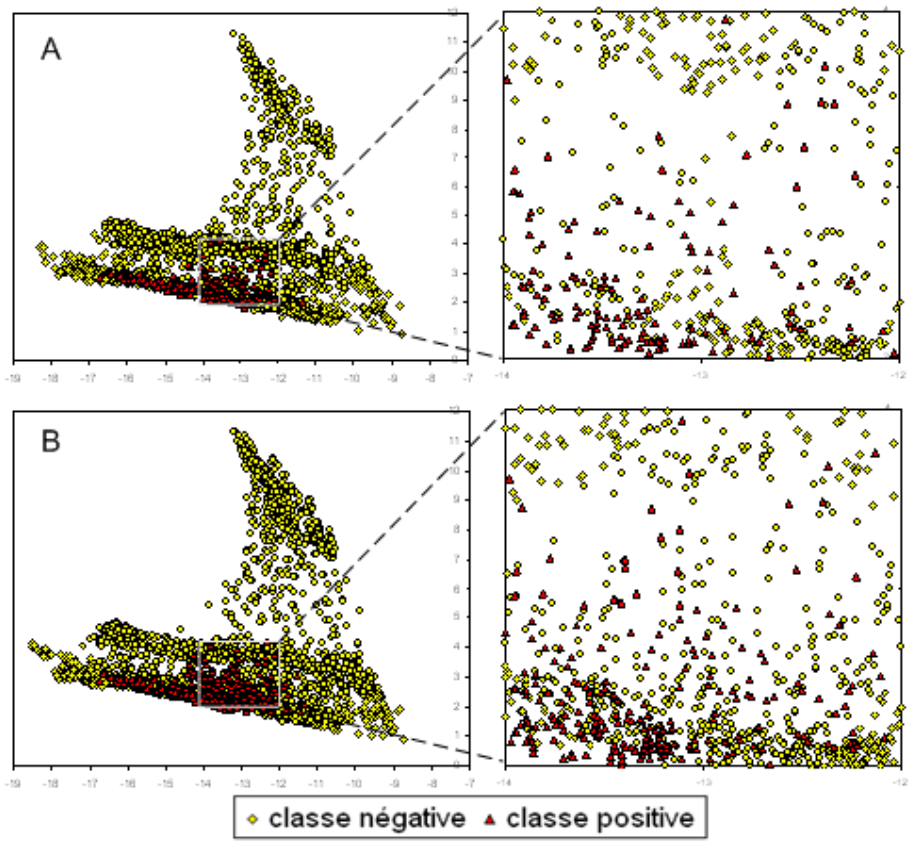


FIG. 5.7 – Visualisation dans le plan formé par les 2 premiers axes ACP de 2 échantillons issus du jeu Satimage : (A) Bootstrap, (B) FUNSS localisé (50^e arbre).

taux de précision de la classe minoritaire, mais évoluent dans leur globalité.

Les tables 5.3 et 5.4 présentent les résultats des tests réalisés en validation croisée de 5 subdivisions, respectivement sur les jeux de données Satimage et MammoClusters dont les compositions sont détaillées en table 5.1. Le jeu MammoClusters est issu des bases de données de la société Fenics pour la mise au point des modèles de détection des cancers de type foyers de microcalcifications. Ce dernier a cependant été réduit en terme d'individus et de variables exogènes, les résultats n'illustrent donc en rien les performances du produit Fenics.

Algorithme	R classe positive	R classe négative	P classe positive	Correction globale
Forêt aléatoire	52,1	98,9	83,2	94,3
LARSS	45,1	99,5	90,1	94,2
Forêt aléatoire Opt-R70	69,7	96,0	66,8	93,4
Balanced Random Forest	71,6	96,4	68,6	94,0
LARSS Opt-R70	69,8	97,4	72,1	94,7
Forêt aléatoire Opt-R80	80,8	93,7	56,2	92,3
SMOTE900	78,4	93,8	57,6	92,3
LARSS Opt-R80	**81,0**	**94,2**	**59,5**	**92,9**
Forêt aléatoire Opt-R90	89,8	88,1	45,4	88,2
LARSS Opt-R90	**89,8**	**90,2**	**49,5**	**90,2**

TAB. 5.3 – Résultats (en %) de différents algorithmes obtenus en validation croisée (5 subdivisions) sur le jeu Satimage.(R : Rappel ; P : Précision)

Les forêts aléatoires bâties sur le principe LARSS sont principalement comparées aux forêts aléatoires classiques et aux forêts aléatoires construites à partir de bootstraps équilibrés (BRF). Les algorithmes présentant la notation "Opt-RXX" utilisent l'optimisation des vecteurs de votes d'une forêt, qui dans un cas à 2 classes, consiste à faire varier le nombre de votes positifs à obtenir pour classer un individu comme positif. Cela permet de choisir le type de performance voulu entre taux de rappel et taux de précision de la classe minoritaire. Tous les détails de ce procédé sont présentés en section 7.3. La notation "Opt-RXX" signifie que le seuil sur les votes a été fixé pour

obtenir un taux de rappel de la classe minoritaire au plus proche possible de "XX%". Les forêts utilisées sont constituées de 50 arbres, 6 et 20 variables exogènes sont tirées au sort lors de la randomisation respectivement pour le jeu Satimage et le jeu MammoClusters. LARSS est paramétré avec des groupes de 10 individus négatifs concurrents et des coefficients d'attraction définis sur $[-5; 5]$.

Algorithme	R classe positive	R classe négative	P classe positive	Correction globale
Forêt aléatoire	74,2	95,4	87,2	89,1
Balanced Random Forest	81,4	92,2	81,6	89,0
LARSS	75,0	95,6	87,9	89,5
Forêt aléatoire Opt-R80	80,1	92,2	81,5	88,5
BRF Opt-R80	79,0	93,8	84,4	89,3
LARSS Opt-R80	**80,1**	**94,0**	**84,9**	**89,9**
Forêt aléatoire Opt-R90	90,1	82,3	68,5	84,6
BRF Opt-R90	90,1	82,8	68,7	84,9
LARSS Opt-R90	**90,2**	**83,5**	**69,7**	**85,5**

TAB. 5.4 – Résultats (en %) de différents algorithmes obtenus en validation croisée (5 subdivisions) sur le jeu MammoClusters.(R : Rappel ; P : Précision)

On remarque qu'en conservant une règle de vote à la majorité lors de l'agrégation, LARSS obtient des taux de rappel de la classe minoritaire inférieurs à une BRF, mais avec un taux de correction globale supérieur. On notera par contre qu'à taux de rappel équivalent pour la classe minoritaire, les forêts aléatoires construites à l'aide de LARSS obtiennent de meilleures performances sur l'ensemble des autres indicateurs.

5.5 Conclusion

Nous avons vu dans ce chapitre que l'une des méthodes les plus utilisées pour palier les problèmes générés par le déséquilibre est l'échantillonnage. Les techniques usuelles ont pour but de rééquilibrer les effectifs entre chaque

classe pour ensuite appliquer un apprentissage supervisé classique. Pour ce faire :

1. soit ces dernières sur-échantillonnent la classe minoritaire, de manière aléatoire ou en créant des individus synthétiques pour éviter le sur-apprentissage.
2. soit celles-ci sous-échantillonnent la ou les classes majoritaires, de manière aléatoire ou guidée. Cependant le sous-échantillonnage est rapidement limité lorsque le déséquilibre est de plus en plus marqué, car dans ces situations extrêmes il entraine une trop grande perte d'informations.

Il est parfois difficile de se prononcer sur le choix entre une technique de sur ou de sous échantillonnage pour un problème de déséquilibre donné. De plus ces méthodes tentent de se ramener vers un cas "classique" équilibré et ne prennent pas en considération les besoins spécifiques en terme de performance souvent liés à la nature même du déséquilibre.

Les nouvelles méthodes d'échantillonnage que nous proposons, FUNSS (*Fitting User Needs Sampling Strategy*) et LARSS (*Local Attraction/Repulsion Sampling Strategy*), tente de tirer partie des propriétés des méthodes ensemble, et plus précisément du bagging, pour apporter une alternative sur ces deux points. Tout d'abord FUNSS et LARSS ne changent pas a priori la distribution des classes, puisqu'elles respectent la composition proposée par un bootstrap. FUNSS et LARSS s'adaptent de cette manière à n'importe quel déséquilibre, que celui-ci soit marqué ou non. FUNSS guide la sélection des individus de la classe majoritaire en utilisant des propriétés de voisinage pour permettre un apprentissage correct malgré le déséquilibre. Ce procédé permet également d'orienter le modèle vers un type de performances souhaitées, et combiné aux forêts aléatoires, de respecter les besoins de l'utilisateur.

Les tests réalisés montrent que remplacer le bootstrap classique par la méthode LARSS, lors de la construction d'une forêt aléatoire, permet d'obtenir des performances supérieures dans une situation de déséquilibre. LARSS obtient également de meilleurs résultats que d'autres méthodes d'échantillonnage comme les BRF ou encore SMOTE.

L'apprentissage supervisé a pour base de travail un tableau individus/valeurs. C'est pourquoi après avoir présenté dans ce chapitre les méthodes existantes et nos contributions touchant à la sélection des individus, pour répondre à notre problématique, nous allons voir dans le chapitre suivant celles touchant aux valeurs, et donc à l'espace de représentation.

Chapitre 6

Espace de représentation

6.1 Introduction

Après avoir vu comment il est possible de jouer sur l'espace des individus en modifiant l'échantillonnage pour l'adapter aux problématiques des jeux de données déséquilibrées, nous nous intéressons dans ce chapitre à l'espace de représentation. La qualité d'un apprentissage est entre autres choses liée à la présence de variables discriminantes. Or, dans le cas d'une qualité d'apprentissage insuffisante, il est nécessaire de trouver un moyen qui, à partir de l'information disponible, permet de re-décrire les données d'entrée du problème d'apprentissage considéré en obtenant de nouvelles variables discriminantes. Les méthodes de construction de variables résolvent ce problème.

La construction de variables permet la création de nouvelles variables synthétiques. Ces variables synthétiques sont issues de la découverte de relations entre variables initiales. Les méthodes de construction de variables entraînent une augmentation de la taille de l'espace de représentation des données, dans la mesure où de nouvelles variables sont construites. Cependant, aucune information extérieure aux données initiales n'est ajoutée lors du processus de construction. Plusieurs auteurs ont défini la construction de variable selon leur point de vue, citons entre autres :

- Murphy, pour qui il s'agit de toute forme d'induction générant de nouvelles descriptions non présentées dans les données d'entrées initiales [MP91].
- Mitchell, qui l'a définie comme le procédé d'accroissement de l'espace de représentation, basé sur les connaissances du domaine [Mit97].
- Rendell, qui l'énonce comme la création de concepts utiles n'existant pas dans la description initiale des données [Ren88].

Après le rappel des principales méthodes existantes pour la construction de variables, nous proposerons dans ce chapitre une nouvelle méthode de construction de variables nommée FuFeFa (Fuzzy Feature Factory). Celle-ci tente de construire des variables spécifiques à la discrimination d'individus au sein de jeux de données déséquilibrées en s'appuyant sur la découverte de règles d'association prédictives.

6.2 Etat de l'art

6.2.1 Taxonomie

Il existe différentes taxonomies des méthodes de construction de variables. Pour présenter celles-ci, nous utiliserons la classification en quatre groupes proposée par Legrand [Leg04], elle-même basée sur les taxonomies de Bloedern [BM98] et Fawcett [Faw93]. Ces quatres catégories sont :

1. Les méthodes de construction par analyse topologique des arbres.
2. Les méthodes de construction par analyse et exploration des données.
3. Les méthodes de construction basée sur l'utilisation des connaissances du domaine ou d'un expert.
4. Les méthodes multi-stratégiques.

La table 6.1 synthétise cette classification. Cette liste n'est pas exhaustive et a pour but de présenter les différents grands types de méthodes.

topologie des arbres	exploration des données	connaissances du domaine	multi-stratégiques
FRINGE [Pag89]	BACON [LBS83]	MIRO [DRC89]	INDUCE-1 [ML77]
CITRE [Mat90]	STAGGER [SG86]		
	FCE [Car92]		

TAB. 6.1 – Classification des méthodes de construction de variables.

6.2.2 Méthodes par analyse topologique des arbres

Les méthodes de ce type déterminent de nouvelles variables par analyse des règles issues d'arbres d'induction. Nous illustrons ce type d'analyse par deux méthodes : CITRE [Mat90] et FRINGE [Pag89].

CITRE est un système basé sur les arbres de décision. Il effectue de la construction de variables en sélectionnant des relations dans les branches d'un arbre d'induction. Un arbre est construit. CITRE sélectionne des paires de relations booléennes à partir des noeuds des branches de l'arbre, comme par exemple la relation booléenne : *couleur='rouge' et taille='grand'* où *couleur* et *taille* sont deux variables initiales. La sélection des relations booléennes se fait grâce à l'une des méthodes suivantes :

- Root : Sélection des relations dans les deux premiers noeuds de chaque branche.
- Fringe : Sélection des relations dans les deux derniers noeuds de chaque branche.
- Adjacente : Toutes les paires adjacentes le long de chaque branche.
- All : Toutes les combinaisons de paires de variables le long de chaque branche.

A partir de ces relations booléennes, CITRE forme de nouvelles variables booléennes : pour l'exemple précédent, la variable créée sera de la forme suivante : si un individu possède à la fois la modalité *'rouge'* pour la variable *couleur* et la modalité *'grand'* pour la variable *taille* alors la valeur de la nouvelle variable pour cet individu sera 1, sinon 0.

FRINGE s'applique initialement sur des problèmes à deux classes avec des variables exogènes booléennes. Cependant, la présence de variables qualitatives ne pose pas de problème. En effet, ces dernières subissent alors un codage disjonctif complet. L'algorithme initial construit de nouvelles variables constituées de conjonctions de propositions des deux derniers noeuds précédant les feuilles, menant à une conclusion positive, c'est-à-dire visant la classe d'intérêt. Plusieurs raisons expliquent le choix de cette stratégie :

- Les noeuds terminaux (feuilles) sont moins sûrs, puisqu'ils couvrent très peu d'individus.
- La répétition de séquences de sous-arbres a lieu, le plus souvent, dans la partie basse du modèle.

D'autres études ont permis, par la suite, d'enrichir la liste des formes détectées avec le système FRINGE. Les travaux [YBR91] ont permis d'y ajouter les disjonctions, et les travaux [OV93], la forme XOR. L'algorithme de base ne change pas, ces perfectionnements touchant uniquement au pouvoir de représentation des nouvelles variables construites.

6.2.3 Méthodes par analyse et exploration des données

Ces méthodes analysent et explorent les données, les relations existantes entre variables exogènes, individus, et variable endogène afin de déterminer de nouvelles variables. Il existe de nombreuses méthodes de ce type. Nous ne citerons qu'un petit nombre de ces méthodes.

Tout d'abord l'algorithme **STAGGER** génère de nouvelles variables à l'aide de combinaisons booléennes de variables numériques [SG86]. Les travaux qui ont suivi, [Sch87] y ont ajouté le partitionnement de variables numériques. Les nouvelles variables sont formées en appliquant les opérateurs booléens ET, OU et NON aux variables existantes selon un procédé composé de plusieurs heuristiques. Ainsi, STAGGER peut apprendre des combinaisons linéaires de variables.

La méthode **BACON** base pour sa part sa procédure de construction sur

les interdépendances existantes entre les variables numériques [LBS83].

Enfin **FCE** prend comme point de départ un ensemble d'espaces de représentation chacun composé par un tirage aléatoire sur les variables exogènes. Puis il génère un nouvel espace de représentation par produit des espaces générés : sa taille est donc supérieure à celle des espaces initiaux [Car92].

6.2.4 Méthodes basée sur l'utilisation des connaissances du domaine

Ces méthodes utilisent les connaissances du domaine ou les connaissances fournies par un expert dans le but de construire de nouvelles variables. Citons par exemple la méthode **MIRO** qui utilise les connaissances du domaine sous la forme d'un ensemble de règles spécifiées par un expert afin de construire de nouvelles variables. Celles-ci sont binaires, un individu prend la valeur 1 s'il respecte la règle, 0 sinon [DRC89].

6.2.5 Méthodes multi-stratégiques

L'approche multi-stratégique est une des méthodologies les plus importantes en construction de variables. Elle consiste souvent en l'emploi simultané de méthodes de différents types parmi ceux précédemment cités. Il existe cependant des méthodes proprement multi-stratégiques comme l'algorithme INDUCE-1 [ML77].

INDUCE-1 est une méthode dirigée à la fois par les données et par les connaissances du domaine. Elle utilise une variété de règles et de procédures pour générer de nouvelles variables, nommées méta-variables. Ceci s'effectue grâce à une description structurelle des exemples d'apprentissage, qui correspond à la partie dirigée par les connaissances du domaine, associée à la détermination des dépendances qualitatives entre les variables, qui correspond à la partie dirigée par les données.

6.3 FuFeFa : Fuzzy Feature Factory

Notre méthode, nommée FuFeFa pour *"Fuzzy Feature Factory"*[1], se rapproche des méthodes de construction de variables par analyse de la topologie des arbres, même si celle-ci n'analyse pas des branches d'arbres ou des noeuds, mais se base sur la découverte de règles d'association prédictives. Le but de cette méthode est de construire de nouvelles variables discriminantes spécifiques à la nature déséquilibrée du jeu de données d'apprentissage. En effet nous prenons comme hypothèse que lorsqu'un jeu de données est déséquilibré, il existe (1) de large plage de l'espace de représentation ne contenant quasiment aucun individu positif (c'est-à-dire de la classe minoritaire), mais également (2) des sous-espaces plus étroits denses en individus positifs.

Le principe de FuFeFa est de découvrir ces sous-espaces à l'aide de règles d'association prédictives, puis de les traduire en nouvelles variables pour l'apprentissage. FuFeFa se décompose en trois étapes détaillées ci-après :

1. Découverte de règles d'association prédictives pour chaque classe.
2. Relâchement des bornes de chaque item de chaque règle.
3. Création d'une nouvelle variable par règle découverte.

6.3.1 Règles d'association

Les règles d'associations [HHM66, AIS93] sont utilisées pour la découverte d'informations. Elles se présentent sous la forme *si Antécédent alors Conséquent* où antécédent et conséquent sont des propositions logiques composées d'items. Prenons par exemple la règle d'association illustrative suivante :

si aire<20 et périmètre>35 alors etirement='fort'

[1] Le terme "Fuzzy" n'a ici aucun lien avec la "Fuzzy Logic" ou logique floue, mais ce rapporte au principe de relâchement des bornes décrit en 6.3.2

Ici, aire, périmètre et étirement sont trois variables. L'antécédent de la règle se compose de deux items. Le conséquent d'un seul item. Chaque item se décompose en une variable, un opérateur logique binaire (ici, '<','>','='), et soit une borne lorsqu'il s'agit d'une variable quantitative, soit une modalité lorsqu'il s'agit d'une variable qualitative.

Deux principales mesures caractérisent une règle d'association : le *support* et la *confiance*.
Soit n l'effectif total d'une population d'apprentissage Ω_a.
Soit n_A le nombre d'individus satisfaisants l'antécédent de la règle.
Soit n_{AC} le nombre d'individus satisfaisant l'antécédent et le conséquent de la règle.
Le support et la confiance sont définies par :

$$support = \frac{n_A}{n}$$

$$confiance = \frac{n_{AC}}{n_A}$$

On peut donc assimiler la confiance au taux de vérification de la règle créée et le support au nombre d'individus concernés par la règle.

Dans le cadre de la méthode FuFeFa, les règles d'association utilisées sont des règles d'association prédictives. Les règles d'association prédictives [BWY98] ont comme particularité d'avoir comme conséquent un unique item portant sur la variable endogène. Elles permettent donc une classification directe des individus sous réserve de faire partie du support de la règle. Notons que l'algorithme FuFeFa ne travaillera que sur les variables exogènes quantitatives pour les antécédents, car il a été créé en premier lieu pour des jeux de données issu de l'imagerie où la totalité des variables exogènes sont quantitatives.

L'algorithme de découverte des règles d'association prédictives utilisé par FuFeFa a été mis au point par la société Fenics et ne sera pas détaillé. Ce dernier est optimisé pour les problèmes à grande dimensionnalité. Il est paramétré de manière à obtenir des règles conjonctives visant chaque classe de la variable endogène, avec des supports et des confiances élevées vis-à-vis des

effectifs et des fréquences de chaque classe (voir expérimentations en 6.4).

6.3.2 Relâchement des bornes des items

Soit I un item d'un antécédent de l'une des règles découvertes. I est défini par X_j une variable exogène, $>$ un opérateur binaire, et b une valeur de borne. On associe à I une fonction de satisfaction des individus $f_I(\omega_i)$ où $f_I(\omega_i) = 1$ si l'individu ω_i satisfait l'item I, et $f_I(\omega_i) = 0$ dans le cas contraire. On note n_I le nombre d'individus pour lesquels $f_I(\omega_i) = 1$. La figure 6.1 illustre ces notions.

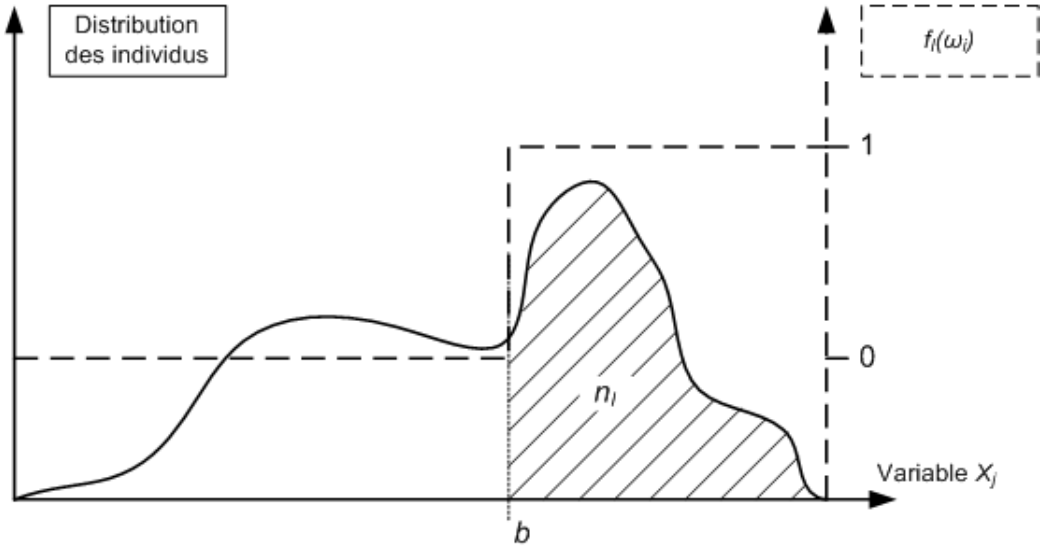


FIG. 6.1 – Réponse de satisfaction à un item I et distribution des individus selon une variable quantitative X_j

Le but de l'étape de relâchement des bornes est de transformer $f_I(\omega_i)$ qui présente une discontinuité en b, en une fonction continue $g_I(\omega_i)$ prenant en considération la proximité (intérieure ou extérieure) à la borne. Pour ce faire, l'utilisateur détermine deux paramètres, t_{int} et t_{ext}, respectivement taux d'intériorisation et d'extériorisation. Cela permet de déterminer deux nouvelles

bornes b_{int} et b_{ext} respectant :
Le nombre d'individus prenant leur valeur sur X_j entre b et b_{int} et ayant $f_I(\omega_i) = 1$ est égal à $n_I \times t_{int}$.
Le nombre d'individus prenant leur valeur sur X_j entre b et b_{ext} et ayant $f_I(\omega_i) = 0$ est égal à $n_I \times t_{ext}$.

La discontinuité de f_I en b est alors remplacée par le segment $[(b_{int}, 1); (b_{ext}, 0)]$ pour obtenir g_I. La figure 6.2 présente un exemple de fonction g_I.

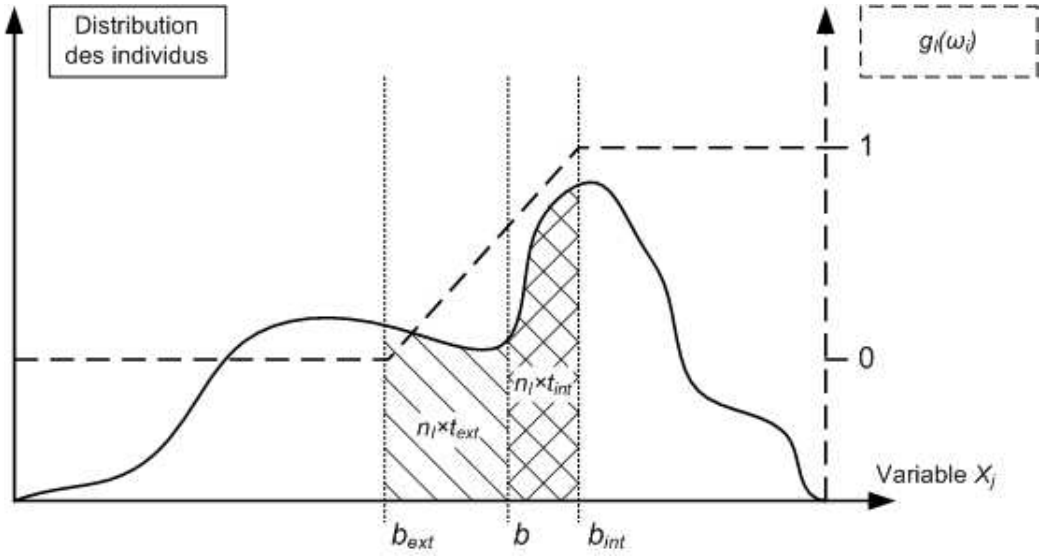


FIG. 6.2 – Fonction $g_I(\omega_i)$ associée à un item I et distribution des individus selon une variable quantitative X_j

6.3.3 Création des variables et utilisation en forêt aléatoire

Pour chaque règle d'association prédictive découverte, une variable est construite comme suit :
Soit R une règle découverte dont l'antécédent est une conjonction de k items

I_h, $h \in \{1; ...; k\}$.
Soit X_R la variable associée à R et $X_R(\omega_i)$ la valeur de l'individu ω_i selon cette variable.

$$X_R(\omega_i) = \prod_{h=1}^{h=k} g_{I_h}(\omega_i)$$

La variable construite est le produit des fonctions de satisfaction aux bornes relâchées des différents items de l'antécédent de la règle d'association prédictive.

Pour illustrer et expérimenter notre méthode, nous allons choisi de l'utiliser avec des forêts aléatoires. Celle-ci ne jouant que sur l'espace de représentation, elle peut s'utiliser conjointement avec n'importe quelle méthode d'apprentissage de classes. Un apprentissage par forêt aléatoire utilisant la méthode de création de variables FuFeFa (noté FFFRF pour FuFeFa Random Forest) se décompose en trois étapes :

1. Découverte de règles d'association prédictives visant tour à tour chaque classe de la variable endogène et respectant les contraintes de support et de confiance fixées par l'utilisateur.
2. Création d'une nouvelle variable par règle découverte.
3. Apprentissage par forêt aléatoire sur le nouvel espace de représentation composé des variables initiales et des variables construites par FuFeFa.

6.4 Expérimentations

Les tests présentés dans cette section ont été réalisés sur le jeu de données Satimage [HB99] déjà présenté en 5.3.2 et le jeu de données MammoMasses issu des bases de données de la société Fenics, mais fortement diminué en nombre d'individus et en nombre de variables pour éviter tout lien entre les résultats présentés et les performances des systèmes d'aide au diagnostic Fenics. Le jeu MammoMasses comporte 15278 individus étiquetés selon deux classes : "cancer" (429 individus) et "non cancer" (14849 individus). La

classe minoritaire, labelisée "cancer", représente 2,8% du jeu total, il s'agit d'un déséquilibré très élevé. Chaque individu est caractérisé par 941 variables exogènes. Les tests ont été réalisés en validation croisée de 5 subdivisions, les forêts sont composées de 50 arbres et respectivement 6 et 31 variables en randomisation pour les jeux Satimage et MammoMasses.

Le paramétrage d'une FFFRF est peu intuitif et se décompose en deux parties. Tout d'abord les paramètres consernant la découverte des règles d'association prédictives envers chaque classe : notre but étant de découvrir soit (1) de larges régions de l'espace de représentation ne contenant quasiment aucun individu positif (c'est-à-dire de la classe minoritaire), soit (2) des régions plus étroites denses en individus positifs (voir 6.3), les règles représentant ces régions doivent respecter des supports et des confiances minimaux. Cela permet également de s'assurer de la qualité en généralisation des règles, mais nécessite de la part de l'utilisateur une phase de vérification de faisabilité de découverte de règles en fonction des contraintes de support et de confiance qu'il fixe. Ensuite, les autres paramètres consernent l'étape de relâchement des bornes, il s'agit des taux d'intériorisation et d'extériorisation. Si ceux-ci sont trop petits, très peu d'individus prendront des valeurs strictement entre 0 et 1 sur la variable construite, celle-ci perdant alors de son intérêt. Si au contraire ils sont trop grands, la notion de borne intéressante s'efface au profit d'un classement sur la variable considérée. Le paramétrage utilisé pour les expérimentations est détaillé en table 6.2 et résulte de tests préalables. On notera que tous les supports minimaux utilisés représentent 20% de l'effectif de la classe visée par les règles.

Les tables 6.3 et 6.4 présentent les résultats obtenus sur chacun des deux jeux de données par des forêts aléatoires soit classiques, soit utilisant la construction de variable FuFeFa. De la même manière qu'en 5.4.2, les algorithmes présentant la notation "Opt-RXX" utilisent l'optimisation des vecteurs de votes d'une forêt. Rappelons que dans un cas à 2 classes, cela consiste à faire varier le nombre de votes à obtenir pour classer un individu comme appartenant à la classe minoritaire. Cela permet de choisir le type de performance voulu entre taux de rappel et taux de précision de la classe minoritaire. Tous les détails de ce procédé sont présentés en section 7.3. La

Paramètre	Test Satimage	Test MammoMasses
Règles visant la classe majoritaire :		
Support minimal	0,18	0,19
Confiance minimale	0,95	0,99
Nombre d'items maximal	4	4
Règles visant la classe minoritaire :		
Support minimal	0,02	0,005
Confiance minimale	0,6	0,5
Nombre d'items maximal	4	4
Taux d'intériorisation	0,3	0,3
Taux d'extériorisation	0,3	0,3

TAB. 6.2 – Paramètres utilisés

notation "Opt-RXX" signifie que le seuil sur les votes a été fixé pour obtenir un taux de rappel de la classe minoritaire au plus proche possible de "XX%".

Pour le test sur le jeu Satimage, selon les subdivisions de la validation croisée et selon la classe visée, entre 5 et 6 règles d'association prédictives ont été découvertes. Cela signifie qu'entre 10 et 12 variables construites se sont ajoutées aux variables initiales pour la construction des FFFRF. Dans le cas du jeu MammoMasses, entre 7 et 8 règles ont été trouvées par subdivision et par classe, d'où l'ajout d'entre 14 et 16 variables.

Les résultats sur le jeu Satimage montrent un gain nul ou faible en terme de performance. Par contre sur le jeu MammoMasses, présentant un déséquilibre beaucoup plus marqué (classe minoritaire de 2,8% de l'effectif global), l'amélioration est notable. En particulier, on remarque que pour des forêts optimisées de manière à obtenir des taux de rappel élevés pour la classe minoritaire, les FFFRF se comportent mieux que les forêts aléatoires classiques.

Algorithme	R classe minoritaire	R classe majoritaire	P classe minoritaire	Correction globale
Forêt aléatoire	52,1	98,9	83,2	94,3
FFFRF	52,7	98,8	83,0	94,3
Forêt aléatoire Opt-R70	69,7	96,0	66,8	93,4
FFFRF Opt-R70	70,5	95,9	67,0	93,4
Forêt aléatoire Opt-R80	80,8	93,7	56,2	92,3
FFFRF Opt-R80	80,7	94,0	56,8	92,6
Forêt aléatoire Opt-R90	89,8	88,1	45,4	88,2
FFFRF Opt-R90	90,8	88,0	45,5	88,3

TAB. 6.3 – Résultats (en %) de différentes forêts aléatoires obtenus en validation croisée (5 subdivisions) sur le jeu Satimage.(R : Rappel ; P : Précision)

6.5 Conclusion

Ce chapitre nous a permis d'appréhender différentes méthodes d'enrichissement de l'espace de représentation. Ces méthodes de construction de variables vont de l'analyse topologique d'arbres à l'utilisation de techniques d'exploration des données en passant par la traduction de connaissances d'expert. Ces différentes méthodes peuvent bien sûr être utilisées conjointement. Cependant aucune n'est spécifique à la problématique des jeux de données déséquilibrées.

C'est pourquoi nous avons proposé la méthode FuFeFa (*Fuzzy Feature Factory*). En partant de l'hypothèse que lorsqu'un jeu de données est déséquilibré, il existe de vastes sous-espaces vides, ou presque vides, d'individus de la classe minoritaire, et dans le même temps des sous-espaces plus restreints, mais denses en individus de la classe minoritaire (assimilable en partie à des sous concepts de la variable endogène), nous pensons qu'intégrer sous forme de variables exogènes une notion de proximité ou d'éloignement à ces localités peut aider à améliorer l'apprentissage.

Pour réaliser ceci, FuFeFa se compose de trois étapes :

Algorithme	R classe minoritaire	R classe majoritaire	P classe minoritaire	Correction globale
Forêt aléatoire	22,3	99,8	84,2	97,7
FFFRF	26,6	99,8	83,9	97,8
Forêt aléatoire Opt-R50	50,1	98,5	50,2	97,1
FFFRF Opt-R50	50,1	98,7	51,3	97,3
Forêt aléatoire Opt-R70	70,5	94,1	25,3	93,4
FFFRF Opt-R70	**71,0**	**94,4**	**27,5**	**93,7**
Forêt aléatoire Opt-R90	91,8	61,6	6,8	66,4
FFFRF Opt-R90	**92,9**	**69,4**	**11,4**	**70,1**

TAB. 6.4 – Résultats (en %) de différentes forêts aléatoires obtenus en validation croisée (5 subdivisions) sur le jeu MammoMasses.(R : Rappel ; P : Précision)

1. Découverte des sous espaces particuliers à l'aide de règles d'association prédictives.
2. Traduction de la notion de proximité par relâchement puis produit des valeurs des bornes des items des règles découvertes.
3. Ajout à l'espace de représentation d'une nouvelle variable par règle aux bornes relâchées.

Les tests réalisés nous poussent à poser l'hypothèse que la méthode trouve un intérêt certain lorsque le déséquilibre est important. Sinon l'information ajoutée est trouvée directement par l'algorithme d'apprentissage dans les variables initiales. Celle-ci reste à être confirmée par un futur plan de tests.

La principale difficulté de cette méthode réside dans la découverte des règles d'association prédictives respectant les contraintes fortes en support et en confiance traduisant nos hypothèses. Nous travaillons actuellement sur une nouvelle adaptation de notre technique, plus proche encore des méthodes par analyse topologique des arbres et davantage intégrée à la construction d'une forêt aléatoire, qui tend à résoudre ce point. Très sommairement, cette technique nommée G2S (pour *Gradual Shaping Space*) modifie la construc-

tion d'une forêt aléatoire comme suit :

1. Construction d'un arbre.
2. Sélection des "meilleures" feuilles en terme de support et de confiance par classe.
3. Traduction des branches amenant à ces feuilles sous forme de règles d'association prédictives.
4. Transformation de ces règles en variables, de la même manière que FuFeFa.
5. Ajout à l'espace de représentation et construction de l'arbre suivant.

L'étape de sélection de feuilles, voire de portion de branches, est très importante et nécessite d'être appronfondie.

Notons enfin que nous utilisons aussi FuFeFa pour réduire la dimensionnalité de l'espace de représentation. Il s'agit simplement de construire les variables issues de FuFeFa et de limiter l'espace de représentation uniquement à ces variables construites. Les premiers tests réalisés en apprentissage sur cet espace réduit montrent en moyenne des pertes de taux de correction globale inférieures à 1%. La réduction d'espace de représentation peut être intéressante pour la construction de graphes de voisinage et la recherche de similarités.

Nous avons vu en chapitre 5 les modifications possibles sur l'espace des individus, et dans ce chapitre, celles sur l'espace de représentation, pour répondre à notre double objectif d'apprentissage sur des jeux déséquilibrés et d'obtention de performances en adéquation avec les besoins de l'utilisateur. Le chapitre suivant touche à la dernière étape de la construction d'une méthode ensemble (voir chapitre 3.6), à savoir l'agrégation des classifieurs de base, cette étape impactant directement le type de performance obtenu. C'est pourquoi nous proposons de modifier et d'optimiser cette étape. Cette optimisation nécessite la définition d'une nouvelle mesure de performance intégrant les besoins de l'utilisateur.

Chapitre 7

Mesure de performance

7.1 Introduction

L'évaluation des performances d'un modèle constitue l'étape finale de tout processus d'apprentissage supervisé. Elle est le retour nécessaire à l'utilisateur pour le guider dans la poursuite de sa fouille de données. Ces mesures sont généralement symétriques. De façon pratique, on entend par symétrique le fait que chaque classe, et chaque type d'erreur relative à une même classe, se voient attribuer une importance similaire. Or dans le cas des jeux de données déséquilibrées (très répandues en milieu industriel) cela n'est que très rarement le cas. Dans ce type de problème l'objectif principal est d'identifier les instances représentatives de la classe minoritaire.

L'évaluation des performances des modèles résultant d'un apprentissage à partir d'un jeu de données déséquilibrées doit prendre en considération l'aspect non symétrique de l'importance des classes, sans se limiter à un taux de correction global. Une évaluation locale, c'est-à-dire par classe, doit alors être conduite, quite à devoir fusionner ensuite ces critères en une mesure unique. Le taux de rappel et le taux de précision sont les deux indicateurs de base des performances d'un modèle vis-à-vis d'une classe, même s'il est possible de créer d'autres critères utilisant le dénombrement de chaque type

d'erreurs (insertion ou omission).

Nous allons fait dans le chapitre 4 un tour d'horizon de différentes mesures de performances des modèles d'apprentissage, ainsi que leur lien avec la notion de symétrie, ce qui limite l'intérêt de la plupart dans un contexte de jeux de données déséquilibrées. Nous proposerons dans ce chapitre un nouveau critère d'évaluation appelé PRAGMA (pour *Precision and RecAll rates Guided Model Assessment*). Celui-ci permet d'évaluer les performances des modèles en prenant en considération les aspects spécifiques à l'apprentissage sur des jeux de données déséquilibrées. Puis pour illustrer l'intérêt de ce critère, nous proposons une optimisation des forêts aléatoires utilisant PRAGMA pour intégrer les préférences de l'utilisateur dans la construction du modèle.

7.2 PRAGMA : Precision and RecAll rates Guided Model Assessment

PRAGMA utilise deux principes : la notion d'importance d'une classe et la notion de préférence entre taux de rappel et taux de précision pour chaque classe.

Tout d'abord l'importance d'une classe est représentée par un coefficient θ_i fixé par l'utilisateur et utilisé en fin d'évaluation.

Ensuite pour chaque classe, nous évaluons le modèle en fonction de son taux de rappel (r_i) et de son taux de précision (p_i). Cette fonction $f(r_i, p_i)$, que nous cherchons à minimaliser par analogie avec le nombre d'erreurs d'un modèle, doit avoir les propriétés suivantes :

(1) $f(0,0) = 1$, on fixe la valeur de la pire situation ($r_i = 0$ et $p_i = 0$).

(2) $f(1,1) = 0$, on fixe la valeur de la meilleure situation ($r_i = 1$ et $p_i = 1$).

(3) $\frac{df(r,p)}{dr} < 0, r \in [0;1]$, à taux de précision égal, la mesure doit diminuer lorsque le taux de rappel augmente.

(4) $\frac{df(r,p)}{dp} < 0, p \in [0;1]$, à taux de rappel égal, la mesure doit diminuer lorsque le taux de précision augmente.

Une telle fonction peut avoir l'écriture suivante : $f(r_i, p_i) = 1 - 0.5(r_i + p_i)$

Pour prendre en compte les souhaits de l'utilisateur en terme de préférence entre le taux de rappel et le taux de précision, nous décidons de pondérérer à la fois r_i et p_i :
$f(r_i, p_i) = 1 - 0.5(\lambda \times r_i + \Omega \times p_i) = 1 + \alpha \times r_i + \beta \times p_i$

Le ratio α/β détermine la préférence entre le rappel et la précision (plus celui-ci est grand (supérieur à 1), plus le rappel est préféré ; plus celui-ci est petit (inférieur à 1), plus la précision est préférée ; s'il est égal à 1, cela signifie qu'aucune distinction n'est faite entre le rappel et la précision).

Pour déterminer, de manière instinctive et compréhensible, ces deux paramètres, l'utilisateur doit définir deux situations extrêmes qu'il juge de qualité équivalente. En pratique, ces deux situations sont : (a) celle où le taux de rappel est parfait ($r_i = 1$) et (b) celle où le taux de précision est parfait ($p_i = 1$). Il implique donc à l'utilisateur de définir deux valeurs x et y tel que $f(1, x) = f(y, 1)$. Choisir ces deux valeurs peut être considéré comme répondre aux deux questions suivantes :

- Quel compromis êtes-vous prêt à faire vis-à-vis de la précision pour avoir un taux de rappel de 1 ? (répondre à cette question permet de définir x, avec $0 \leq x < 1$) (a)
- Quel compromis êtes-vous prêt à faire vis-à-vis du rappel pour avoir un taux de précision de 1 ? (répondre à cette question permet de définir y, avec $0 \leq y < 1$) (b)

Avec cette dernière contrainte (5) $f(1, x) = f(y, 1)$, nous pouvons déter-

miner les paramètres α et β :

$$\alpha = \frac{-1}{1 + \frac{(1-y)}{(1-x)}} \quad et \quad \beta = \frac{1}{1 + \frac{(1-y)}{(1-x)}} - 1$$

La fonction f utilisée pour évaluer localement un modèle selon son rappel et sa précision sur une modalité est la suivante :

$$f(r_i, p_i) = \frac{-1}{1 + \frac{(1-y)}{(1-x)}} \times r_i + (\frac{1}{1 + \frac{(1-y)}{(1-x)}} - 1) \times p_i + 1$$

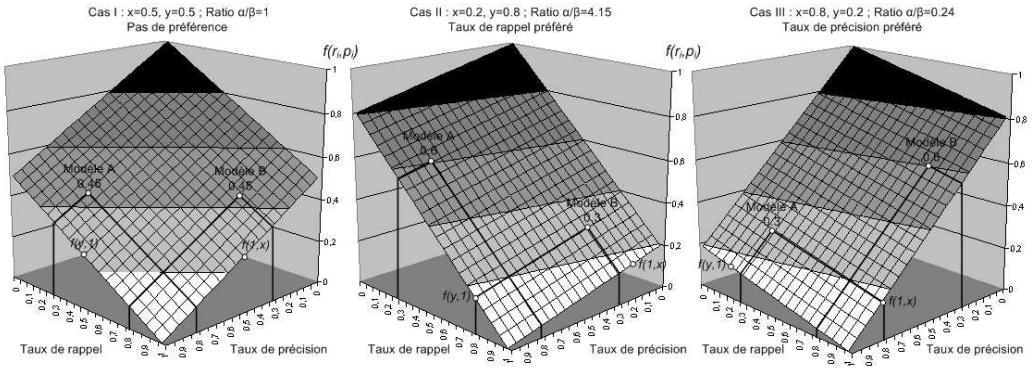


FIG. 7.1 – 2 modèles, A($r = 0.3; p = 0.8$) et B($r = 0.8; p = 0.3$), sont localement évalués à l'aide de 3 différentes $f(r_i, p_i)$: Cas I (symétrique) A et B sont équivalents ; Cas II (rappel préféré) B est meilleur ; Cas III (précision préférée) A est meilleur.

Cette fonction correspond à l'équation d'un plan où l'axe défini par les points $(0, 0, 1)$ et $(1, 1, 0)$ est fixe, et où le ratio α/β détermine l'orientation du plan autour de cet axe (la préférence entre les taux de rappel et de précision) comme illustré en figure 7.1.

Ces évaluations locales (propres à chaque classe) sont ensuite combinées

lors de l'évaluation finale à l'aide d'une moyenne pondérée où les coefficients d'importance de chaque classe constituent les pondérations. Avec n étant le nombre de classe de la variable endogène, on obtient :

$$PRAGMA = \frac{1}{\sum_{i=1}^{i=n} \theta_i} \sum_{i=1}^{i=n} \theta_i \times f_i(r_i, p_i)$$

7.3 Exemple d'optimisation des forêts aléatoires

Lors de l'application d'une forêt aléatoire, la prédiction pour un individu est obtenue en comptabilisant les prédictions de chaque arbre pour l'individu (chaque arbre vote pour une classe) puis en choisissant la classe ayant reçu le plus de voix parmi tous les arbres de la forêt (vote à la majorité).

Les performances d'une forêt aléatoire sont sensiblement supérieures à celles d'un arbre seul tel que C4.5 [LZG04]. Elle est également plus robuste au bruit et présente de meilleures facultés de généralisation [Bre01]. Cependant, celle-ci n'est pas spécifiquement adaptée aux jeux de données déséquilibrées, et ses deux paramètres (le nombre d'arbres et le nombre k de variables à tirer au sort lors de la randomisation) ne permettent pas à l'utilisateur de spécifier ses préférences en termes de taux de rappel et de précision selon chaque classe.

L'évolution que nous proposons ici consiste à remplacer l'étape du vote classique à la majorité par une nouvelle stratégie de vote pondéré où la recherche automatique des poids optimaux se fait à l'aide de PRAGMA.

7.3.1 Stratégie de vote

Notre stratégie de vote consiste à donner plus ou moins d'importance aux voix attribuées par les arbres. Une pondération par classe est déterminée (soit par l'utilisateur, soit automatiquement), laquelle multiplie le nombre

de voix reçues par l'individu pour cette classe. Ainsi la classe assignée à un objet n'est pas toujours celle dont il a reçu le plus de voix, mais celle dont le nombre de voix multiplié par son poids est le plus grand. Ceci permet d'augmenter les taux de rappel des classes minoritaires en leur affectant des pondérations fortes, ou plus généralement de jouer sur les taux de rappel et de précision de chaque classe en modifiant leur pondération.

La figure 7.2 illustre la notion de distribution de votes d'une forêt aléatoire. Une forêt aléatoire de 20 arbres a été construite pour chacun des 2 jeux utilisés. En abscisse, on trouve le nombre de votes reçus par un individu pour une classe donnée, entre 0 et 20 voix. Un graphique du même type pourrait être construit pour chaque classe. En ordonnée, on trouve le nombre d'individus ayant reçu ce nombre de voix, chaque couleur correspondant aux individus d'une classe. On voit par exemple sur le graphique A, la distribution de votes pour la classe 'S' du jeu de reconnaissance de lettres manuscrites Letters [HB99]. Les individus les mieux votés sont effectivement ceux de la classe 'S' (en vert), il apparait néanmoins que parmi les autres individus, ceux les plus souvent votés comme étant des 'S' sont les 'Z' (en rouge). Sur le graphique B, concernant le jeu Mammo (voir 7.4), la distribution présentée est celle pour la classe 'Cancers'. Ce jeu ne possédant que deux classes, la distribution pour la classe 'NonCancers' s'obtiendrait par une symétrie d'axe vertical. On remarque entre autre qu'un large segment en terme de nombre de votes (de 7 à 16) présente des individus des deux classes. La forêt commettra donc des erreurs de prédiction. Avec un vote à la majorité (équivaut à un seuil de décision de 10 votes), les taux de rappel et de précision de la classe 'Cancers' seront affectés par les erreurs commises. Un seuil à 17 votes permettrait d'obtenir un taux de précision parfait, un seuil à 7 votes permettrait d'obtenir un taux de rappel parfait.

7.3.2 Recherche automatique

Il peut être assez difficile de trouver manuellement les pondérations ajustant au mieux les résultats du modèle aux besoins de l'utilisateur. Si pour

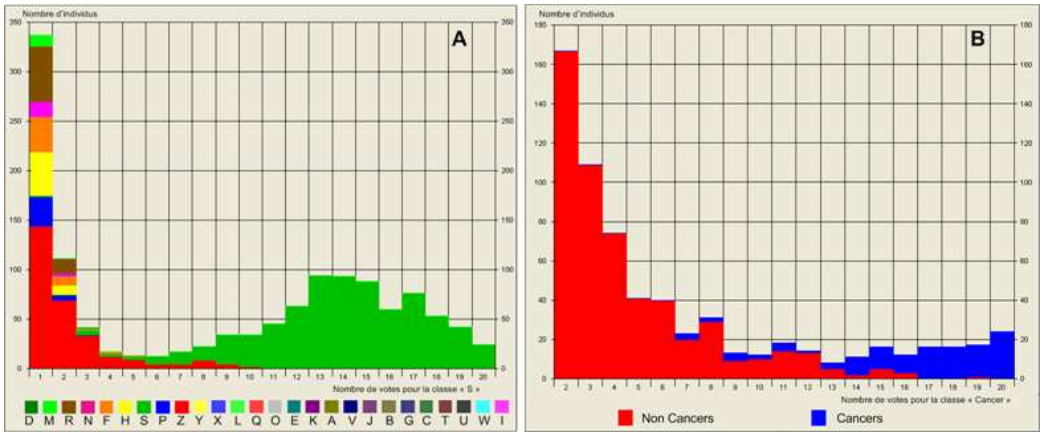


FIG. 7.2 – Exemples de distribution de votes pour : (A) le jeu de données Letters [HB99] (les objets sans vote pour la modalité 'S' ne sont pas représentés) ; (B) le jeu de données Mammo (voir 7.4)(les objets ayant moins de 2 votes pour la modalité 'Cancer' ne sont pas représentés).

un problème à deux modalités tout peut se ramener à un déplacement de la frontière, en terme de nombre de votes, entre les deux classes, dès qu'il y a trois classes ou plus, le nombre de possibilités de paramétrage, c'est-à-dire de ratios entre chaque couple de pondérations, devient bien plus conséquent, et il apparaît nécessaire de rendre automatique la recherche des pondérations.

L'algorithme utilisé pour automatiser la recherche des pondérations est construit autour d'un recuit simulé [KGJV83] cherchant à optimiser la mesure PRAGMA paramétrée selon les souhaits de l'utilisateur. Ce procédé est parfaitement adapté et efficace pour ce type d'optimisation. Le surcoût calculatoire (comparé à une forêt aléatoire classique) est extrêmement faible. En effet, la forêt n'est construite qu'une fois, seul le résultat du vote après pondération est mis à jour pour permettre l'évaluation par PRAGMA. De plus, en conservant pour chaque individu le nombre de votes non pondérés qu'il a reçu pour chaque classe, il suffit de mettre à jour uniquement la matrice de confusion après pondération du vote pour évaluer le modèle et passer à l'itération suivante.

7.4 Expérimentations

Nous présentons dans cette section les résultats obtenus par pondération automatique des votes d'une forêt aléatoire. Deux types de situations ont été envisagés :

1. Utilisation de l'optimisation sur des jeux de données équilibrées. Notre but ici, en tant qu'utilisateur, est de favoriser un maximum le taux de rappel de certaines classes jugées 'prioritaires'. Les tests sont réalisés sur les jeux de données de référence Autos et Letters [HB99] dont les variables endogènes possèdent respectivement 6 et 26 modalités.
2. Utilisation de l'optimisation des jeux de données déséquilibrées à 2 modalités. Notre but dans cette situation est de favoriser un maximum le taux de rappel de la classe minoritaire. Les tests sont réalisés sur les jeux Hypothyroïd et Satimage [HB99] réduits à 2 classes (minoritaire ; fusion des autres classes), ainsi que sur le jeu Mammo, issu de la mise au point du système d'aide au diagnostic de la société Fenics.

7.4.1 Jeux de données équilibrés

Nous supposerons ici que l'utilisateur cherche à maximiser les taux de rappel des classes '_3' et '_2' pour le jeu Autos, et les taux de rappel des voyelles pour le jeu Letters. Ceci se traduit par le paramétrage de la fonction PRAGMA suivant : coefficient d'importance 10 et couple $(x; y) = (10; 90)$ pour les classes prioritaires, coefficient d'importance 1 et couple $(x; y) = (80; 80)$ pour les autres classes. Nous utilisons des forêts aléatoires de 20 arbres, avec respectivement 5 et 4 variables pour la randomisation.

Les résultats présentés en tables 7.1 et 7.2 sont issus d'une validation croisée de 10 subdivisions. La figure 7.3 montre les résultats détaillés sur le jeu Letters. Notez que la classe '__2' du jeu Autos ne contient que 3 objets, les résultats propres à cette classe sont peu significatifs. Les différentes moyennes réalisées sont toujours pondérées par les effectifs des différentes classes.

	'_3'	'_2'	'_1'	'_0'	'__1'	'__2'	MP	MA	MG
RF Class. R	81.4	68.7	74.1	85.1	81.8	33.3	74.6	79.4	78.0
RF Class. P	84.6	84.6	72.7	77.0	81.8	50.0	84.6	75.6	78.2
RF Opt. R	92.6	71.9	74.1	83.6	68.2	66.7	81.4	77.4	78.5
RF Opt. P	80.6	82.1	76.9	75.7	83.3	100.0	81.5	77.8	78.8
Evolution R	**+11.2**	**+3.2**	0.0	-1.5	-13.6	+33.4	**+6.8**	**-2.0**	**+0.5**
Evolution P	-4.0	-2.5	+4.2	-1.3	+1.5	+50.0	**-3.1**	**+2.2**	**+0.6**

TAB. 7.1 – Résultats pour Autos : le taux de correction global de la forêt aléatoire classique (RF Class.) est de 78.0%, celui de la forêt aléatoire optimisée (RF Opt.) est de 78.5%, soit une amélioration +0.5pts. Légende : R Rappel ; P Précision ; MP Moyenne des classes Prioritaires ; MA Moyenne des Autres classes ; MG Moyenne Globale.

Ces différents résultats montrent la capacité de l'optimisation à retranscrire les volontés de l'utilisateur. Pour les deux jeux de données, les taux de rappel des classes ciblées ont augmenté. Il en résulte également de part les liens entre les différents indicateurs issus de la matrice de confusion :

1. Une baisse du taux de précision pour ces mêmes classes.
2. Une baisse du taux de rappel et une augmentation du taux de précision (en moyenne) pour les classes où aucune préférence n'avait été spécifiée.
3. Ces changements n'entraînent pas forcément une diminution du taux de correction global. Celui-ci peut augmenter ou diminuer selon les jeux de données et le paramétrage de la mesure PRAGMA (ici augmentation du taux correction global pour Autos et diminution sur Letters).

7.4.2 Jeux de données déséquilibrées

L'objectif est de détecter un maximum d'objets de la classe minoritaire (taux de rappel élevé) sans présenter trop de faux positifs (taux de précision correct). Nos tests sont réalisés sur 3 jeux de données, dont les compositions sont présentés en table 7.3. Il s'agit des jeux Hypothyroïd et Satimage [HB99] réduits à deux classes en fusionnant les classes non minoritaires et Mammo

	Moy Consonnes	Moy Voyelles	Moy Globale
RF Class. Rappel	88.0	88.5	88.1
RF Class. Précision	87.9	90.8	88.5
RF Opt. Rappel	84.8	95.0	87.1
RF Opt. Précision	90.9	78.9	88.1
Evolution Rappel	**-3.2**	**+6.5**	**-1.0**
Evolution Précision	**+3.0**	**-11.9**	**-0.4**

TAB. 7.2 – Résultats pour Letters : le taux de correction global de la forêt aléatoire classique (RF Class.) est de 88.1%, celui de la forêt aléatoire optimisée (RF Opt.) est de 87.2%, soit une perte -0.9pts.

issu de la mise au point d'un système d'aide au diagnostic du cancer du sein. Notons que ce dernier a été réduit en terme de variables et d'individus pour ne pas dévoiler des résultats industriels confidentiels.

Jeux de données	Effectif	Nombre de variables exogènes	Fréquence de la classe minoritaire
Hypothyroïd	3772	27	7.71%
Satimage	6435	36	9.73%
Mammo	3528	134	5.00%

TAB. 7.3 – Composition des jeux de données déséquilibrées utilisés.

La volonté de maximiser le taux de rappel de la classe minoritaire se traduit par le paramétrage de PRAGMA suivant : coefficient d'importance 10 et couple $(x; y) = (10; 90)$ pour la classe minoritaire, coefficient d'importance 1 et couple $(x; y) = (80; 80)$ pour la classe majoritaire. Nous présentons table 7.4 les résultats obtenus en validation croisée de 10 subdivisions, avec C4.5 (témoin de référence des difficultés pouvant présenter les jeux de données), une forêt aléatoire classique, et une forêt aléatoire optimisée par pondération des votes à l'aide de la mesure PRAGMA. Les forêts sont composées de 20 arbres, avec respectivement 5, 6 et 15 variables utilisées lors de la randomisation pour les 3 jeux de données.

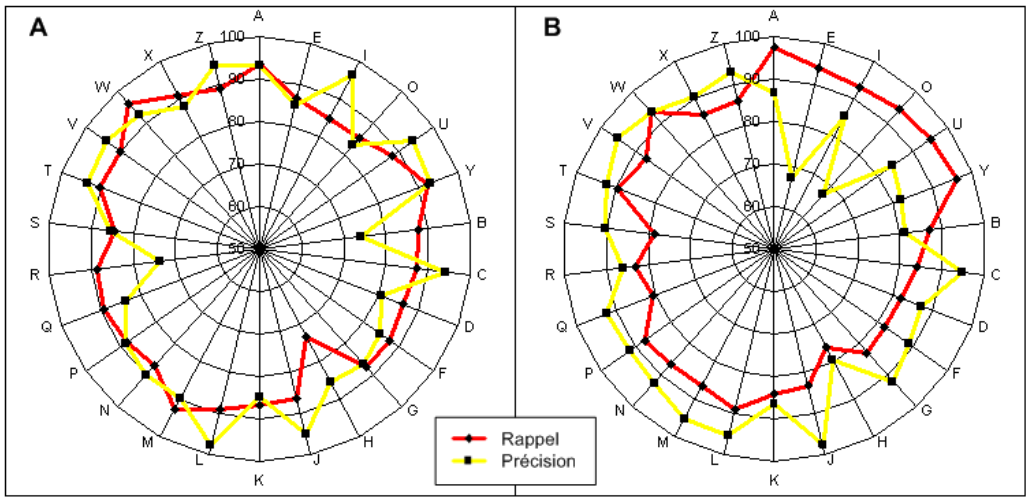


FIG. 7.3 – Résultats détaillés pour Letters : (A) Forêt aléatoire classique ; (B) Forêt aléatoire optimisée, le rappel et la précision "s'organisent" selon les préférences de l'utilisateur.

		Hypothyroïd	Satimage	Mammo
C4.5	Rappel classe minoritaire	96.9	55.0	8.3
	Précision classe minoritaire	95.2	59.0	66.7
	Taux de correction globale	**99.4**	91.9	98.1
RF Class.	Rappel classe minoritaire	95.1	50.9	29.2
	Précision classe minoritaire	94.2	**83.2**	**84.5**
	Taux de correction globale	99.1	94.2	98.5
RF Opt.	Rappel classe minoritaire	**99.5**	**62.0**	**43.5**
	Précision classe minoritaire	**97.7**	73.9	82.0
	Taux de correction globale	99.2	**94.3**	**98.7**

TAB. 7.4 – Résultats (en %) obtenus en 10-CrossValidation pour Hypothyroïd, Satimage et Mammo.

Les taux du rappel des classes minoritaires les plus élevés sont systématiquement obtenus par la forêt aléatoire optimisée, ceci sans provoquer de trop fortes baisses des taux de précision. La mesure PRAGMA guide en cela

parfaitement le modèle vers les performances souhaitées par l'utilisateur. On remarque également que selon les cas l'optimisation permet également parfois d'améliorer le taux de précision de la classe minoritaire ou le taux de correction globale.

Des résultats détaillés obtenus en validation croisée de 10 subdivisions sur le jeu Mammo sont présentés en figure 7.4. Ils permettent une meilleure description de l'effet de la pondération des votes et de l'utilisation de la mesure PRAGMA sur les performances du modèle. Quatre indices (taux de rappel, taux de précision, nombre d'erreurs, mesure PRAGMA) sont évalués pour différentes valeurs du ratio R : pondération des votes pour la classe 'Cancer' / pondération des votes de la classe 'Non Cancer'.

On remarque que les plus fortes variations pour les taux de rappel et de précision se produisent pour la classe 'Cancer' de par son effectif faible. Le graphe du nombre d'erreurs présente deux caractéristiques notables :

1. Celui-ci est asymétrique, car une baisse légère du taux de rappel sur la classe majoritaire due à une forte pondération de la classe minoritaire crée logiquement plus d'erreurs qu'une faible variation du taux de rappel de la classe minoritaire
2. Pour les ratios $2 \leq R \leq 5$ le nombre d'erreurs total varie très peu alors que la nature des erreurs change (voir les graphes des taux de rappel et de précision). Une sorte de transfert d'erreurs se produit :
   $R \leq 2$: les objets mal classés appartiennent majoritairement à la classe 'Cancer'
   $3 \leq R \leq 4$: les proportions d'objets mal classés pour chacune des 2 classes sont similaires
   $5 \leq R$: les objets mal classés appartiennent majoritairement à la classe 'Non Cancer'

La lecture du graphe de la mesure PRAGMA permet de faire différentes observations :

1. L'asymétrie est inversée, montrant ainsi que les variations du taux de rappel de la classe 'Cancer' constituent la principale influence de la

mesure (ceci s'expliquant par le fort coefficient d'importance et le paramétrage orienté vers le taux de rappel pour la classe 'Cancer')

2. Pour les ratio $2 \leq R \leq 5$ le plateau observé pour le graphe du nombre d'erreurs disparaît au profil de variations plus importantes. Ceci montre que la nature des erreurs est prise en considération par la mesure PRAGMA pour laquelle une erreur de classement d'individus de la classe 'Cancer' fait davantage augmenter la mesure que celle d'individus de la classe 'Non Cancer'.

Les souhaits de l'utilisateur de rendre la classe 'Cancer' plus importante et de favoriser son taux de rappel vis-à-vis de son taux de précision sont ainsi visibles à travers la lecture du graphe de la mesure PRAGMA.

7.5 Conclusion

Ce chapitre nous a permis de présenter l'étape d'agrégation des arbres en forêt, où nous proposons un remplacement du vote à la majorité par un vote pondéré. En jouant sur les pondérations il est ainsi possible de jouer sur l'équilibre entre taux de rappel et taux de précision sur chaque classe, pour obtenir des performances en accord avec les souhaits de l'utilisateur. Si cette modification revient à changer le seuil du nombre de votes à obtenir pour classer un individu dans la classe d'intérêt pour le cas d'un jeu à deux modalités, l'automatisation du choix des pondérations devient nécessaire dans les autres cas. C'est dans ce cadre que la mesure PRAGMA est utilisée comme fonction à minimiser lors de l'optimisation. Celle-ci permet la prise en compte des besoins de l'utilisateur en terme de préférence entre le taux de rappel et le taux de précision sur chaque classe, ainsi que l'importance qu'il accorde à chacune d'elles. Les tests réalisés montrent l'efficacité de la méthode.

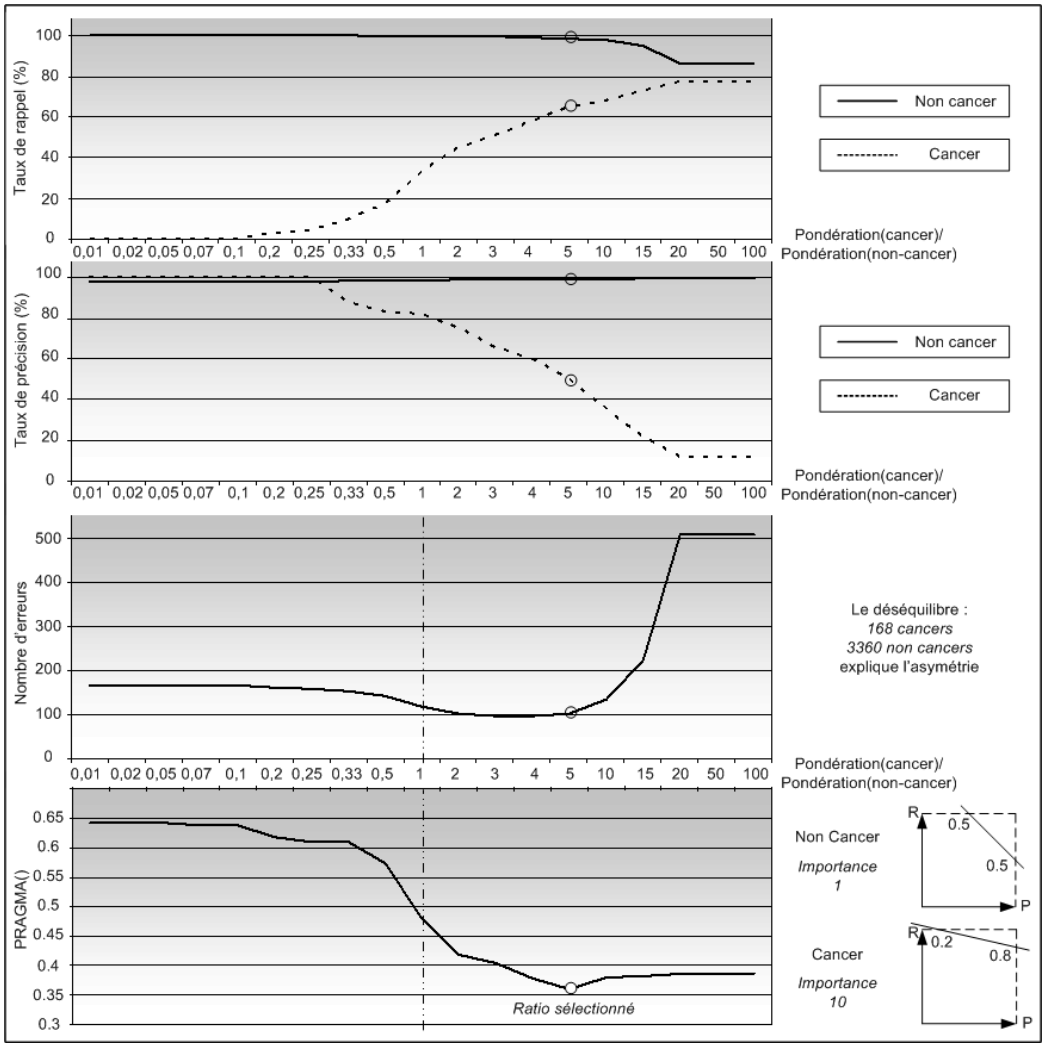


FIG. 7.4 – Résultats détaillés pour le jeu de données Mammo.

Chapitre 8

Conclusion

8.1 Bilan

Dans nombre de problèmes industriels d'apprentissage supervisé les classes sont déséquilibrées, c'est-à-dire que la classe d'intérêt est beaucoup moins représentée que le cas général. C'est notamment le cas dans le domaine médical, lorsqu'on cherche à induire des modèles permettant de prédire la présence d'une maladie chez un patient.

Le double objectif que nous nous sommes fixé dans cette thèse est d'une part l'adaptation des statégies d'échantillonnage et de construction de variables au problème des jeux de données déséquilibrées, et d'autre part la prise en considération de besoins spécifiques de l'utilisateur tels que l'obtention d'un taux de rappel élevé pour la classe minoritaire. Le défi sous-jacent à ces travaux est un projet industriel : la mise au point d'un système d'aide au diagnostic du cancer du sein à partir de mammographies numériques. Pour atteindre ces objectifs nous avons abordé ces questions à travers trois étapes du processus d'apprentissage : l'étape d'échantillonnage (chapitre 5), l'amélioration de l'espace de représentation (chapitre 6), et l'agrégation des classifieurs d'une méthode ensemble pour l'optimisation d'une mesure de performance du modèle (chapitre 7).

Nous avons commencé par présenter les différents éléments de base que sont l'apprentissage supervisé, les méthodes d'apprentissage de classes, l'agrégation de classifieurs, ou encore l'évaluation de modèles. Considérant que certains principes de l'apprentissage supervisés pouvaient être redéfini dans le cadre d'une utilisation pour des jeux déséquilibrés et l'obtention de performances spécifiques, nous avons proposé trois modifications. Nous avons étayé ces dernières d'expérimentations basées sur l'algorithme des forêts aléatoires, mais toutes nos contributions sont généralisables à l'ensemble des classifieurs d'apprentissage supervisé. Seule la première d'entre elles, touchant à l'échantillonnage, nécessite la mise en place d'un bagging.

Tout d'abord en ce qui concerne l'espace des individus (chapitre 5), nous avons présenté deux techniques d'échantillonnage destinées à remplacer le bootstrap classiquement utilisé dans la construction d'un bagging. FUNSS (*Fitting User Needs Sampling Strategy*), la première d'entre-elles, guide la sélection des individus de la classe majoritaire pour spécialiser chaque classifieur de base soit vers un but de taux de rappel de la classe minoritaire élevé, soit vers vers un but de taux de précision de la classe minoritaire élevé. Puis le choix entre ces deux types de classifieurs, lors de leur construction successive, permet l'obtention d'un modèle ayant des performances en adéquation avec les attentes de l'utilisateur. La deuxième, LARSS (*Local Attraction/Repulsion Sampling Strategy*), permet la mise en place progressive au cours de la création du bagging, d'un échantillonnage optimisé. Ce dernier tente de minimiser le nombre d'individus de la classe majoritaire autour des individus de la classe minoritaire que les classifieurs précédents ne réussissent pas à détecter ; et inversement d'augmenter la précision du modèle, en favorisant la présence d'individus de la classe majoritaire autour des individus de la classe minoritaire bien classés. Les tests réalisés montrent les gains de performances obtenus grâce à cette dernière méthode.

Notre seconde contribution touche à l'amélioration de l'espace de représentation (chapitre 6) où notre but est de construire de nouvelles variables exogènes spécifiques à la nature déséquilibrée du jeu de données. Pour ce faire, nous proposons la méthode FuFeFa (*Fuzzy Feature Factory*) qui se décompose en trois étapes :

1. Découverte de règles d'association prédictives de support et de confiance élevés, pour trouver des sous-espaces aux distributions intéressantes.
2. Relâchement des bornes des items, pour intégrer une notion de proximité ou d'éloignement aux frontières de sous-espace considérés, et gagner en capacité de généralisation.
3. Intégration de ces indications sous forme de nouvelles variables exogènes ajoutées à celle initiales pour améliorer les performances d'apprentissage.

Cette méthode semble trouver son intérêt pour les jeux très déséquilibrés et des paramétrages traduisant une volonté de taux de rappel élevé pour la classe minoritaire. Pour des jeux de données moins déséquilibrés, l'information donnée par les nouvelles variables construites est quasiment entièrement retrouvée directement par le classifieur dans les variables initiales.

Enfin, nous proposons une mesure de performance des modèles d'apprentissage supervisé appelée PRAGMA (*Precision and RecAll rates Guided Model Assessment*). Celle-ci est entièrement paramétrable par l'utilisateur pour lui permettre de définir de manière intuitive ses besoins spécifiques selon chaque classe. Cette évaluation quantitative peut entre autre permettre d'optimiser différents éléments d'un processus d'apprentissage supervisé. Nous avons choisi comme exemple l'étape d'agrégation des arbres aléatoires en forêt en remplaçant le vote classique à la majorité par un vote pondéré. Dans cet exemple d'optimisation, les votes pour chaque classe sont multipliés par une pondération avant d'effectuer l'attribution de la classe prédite. Les pondérations sont choisies pour minimiser la mesure PRAGMA et ainsi obtenir le compromis le plus en adéquation avec les attentes de l'utilisateur.

Durant cette thèse nous avons été amenés à traiter d'autres problèmes, qui ne figurent pas dans ce mémoire. Nous avons notamment participé à la mise au point de méthodes de segmentation d'images, pour la détection des cancers du sein. Concernant l'apprentissage, nous nous sommes également penchés sur des questions de labellisation des objets segmentés, ou encore d'amélioration incrémentale des modèles en fonction de l'augmentation du

volume de données.

Pour conclure, rappelons que ces travaux ont été effectués dans le cadre d'une convention CIFRE (Convention Industrielle de Formation par la Recherche). Environ 90% du temps a ainsi été passé en entreprise. Parallèlement aux travaux de recherche présentés dans ce mémoire la conception et le développement de plusieurs logiciels ont été menés, comme le logiciel Smart Look, système d'aide au diagnostic de Fenics, ou surtout LearnIT, notre plateforme interne de data mining, intégrant entre autre l'ensemble des méthodes présentées. Cette opportunité de pouvoir mener un travail de recherche en entreprise, et particulièrement au sein d'une jeune entreprise dynamique et innovante a été une expérience extrêmement enrichissante.

8.2 Perspectives

L'ensemble des travaux menés durant cette thèse nous laisse penser qu'il est difficile aujourd'hui de trouver de nouvelles méthodes d'apprentissage supervisé permettant d'améliorer fortement et de manière globale, c'est-à-dire pour l'ensemble des couples rappel/précision, les performances. Cependant des adaptations permettent d'augmenter spécifiquement celles-ci vis-à-vis d'un besoin précis, comme cela est notre cas dans la mise au point d'un système d'aide au diagnostic, où un taux de rappel élevé pour la classe "cancer" est absolument prioritaire.

Dans un cadre industriel, nos perspectives sont de continuer à accroître les performances de nos modèles de prédiction. Pour cela, nous axons désormais notre travail sur trois fondamentaux en amont de l'apprentissage :

1. L'augmentation du volume de données d'apprentissage : celle-ci nécessite désormais l'utilisation de techniques incrémentales d'amélioration des modèles. Nous travaillons sur ces techniques.
2. L'amélioration de la qualité des individus et de leurs descripteurs fournis à l'apprentissage : il est ici question de la qualité de segmentation des zones extraites d'une mammographie, et donc plus du domaine de

l'imagerie. Nous cherchons également de nouvelles propriétés aidant la discrimination par le modèle, ceci par un dialogue continu avec les radiologues sur leur méthodologie de lecture d'une mammographie.

3. L'amélioration de la qualité de labélisation : celle-ci est très liée à la qualité de la segmentation et au travail réalisé avec les radiologues.

Enfin, nous travaillons désormais sur des mesures de similarité entre données structurées utilisant nos modèles de forêts aléatoires. Ce travail a pour but la mise en place d'un logiciel de recherche par le contenu sur les dossiers mammographiques.

Chapitre 9

Annexe : Données utilisées

Les jeux de données utilisés dans ce travail proviennent de deux sources distinctes. Les jeux Satimage, Letters, Autos et Hypothyroïd sont issus de la collection de l'Université de Californie à IRVINE (http ://kdd.ics.uci.edu/) [HB99]. Le choix d'utiliser ces jeux de données est motivé ici par la volonté de proposer des évaluations expérimentales reproductibles par d'autres chercheurs. Nous reprenons dans les prochaines sections les descriptions de ces jeux de données telles qu'elles sont données dans le repertoire de l'UCI.

Les autres jeux utilisés sont issus de l'activité de la société Fenics, et plus précisemment de la mise au point d'un système d'aide au diagnostic du cancer du sein. Nous ne pouvons donner plus de détails sur ces jeux de données que ceux déjà fournis lors des précédents chapitres où ils sont utilisés.

9.1 Jeu de données SATIMAGE

PURPOSE The database consists of the multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. The aim is to predict this classification, given the multi-spectral values. In the sample database, the

class of a pixel is coded as a number.

PROBLEM TYPE Classification

AVAILABLE This database was generated from Landsat Multi-Spectral Scanner image data. These and other forms of remotely sensed imagery can be purchased at a price from relevant governmental authorities. The data is usually in binary form, and distributed on magnetic tape(s).

SOURCE The small sample database was provided by :
Ashwin Srinivasan
Department of Statistics and Modelling Science
University of Strathclyde
Glasgow, Scotland, UK

ORIGIN The original Landsat data for this database was generated from data purchased from NASA by the Australian Centre for Remote Sensing, and used for research at :
The Centre for Remote Sensing
University of New South Wales
Kensington, PO Box 1, NSW 2033, Australia.

The sample database was generated taking a small section (82 rows and 100 columns) from the original data. The binary values were converted to their present ASCII form by Ashwin Srinivasan. The classification for each pixel was performed on the basis of an actual site visit by Ms. Karen Hall, when working for Professor John A. Richards, at the Centre for Remote Sensing at the University of New South Wales, Australia. Conversion to 3x3 neighbourhoods and splitting into test and training sets was done by Alistair Sutherland.

HISTORY The Landsat satellite data is one of the many sources of information available for a scene. The interpretation of a scene by integrating spatial data of diverse types and resolutions including multispectral and radar data, maps indicating topography, land use etc. is expected to assume significant importance with the onset of an era characterised by integrative approaches to remote sensing (for example, NASA's Earth Observing System commencing this decade). Existing statistical methods are ill-equipped for handling such diverse data types. Note that this is not true for Landsat MSS data considered in isolation (as in this sample database). This data satisfies the important requirements of being numerical and at a single resolution, and standard maximum-likelihood classification performs very well. Consequently, for this data, it should be interesting to compare the performance of other methods against the statistical approach.

DESCRIPTION One frame of Landsat MSS imagery consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and two are in the (near) infra-red. Each pixel is a 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about 80m x 80m. Each image contains 2340 x 3380 such pixels.

The database is a (tiny) sub-area of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3x3 square neighbourhood of pixels completely contained within the 82x100 sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the 9 pixels in the 3x3 neighbourhood and a number indicating the classification label of the central pixel.

The number is a code for the following classes (Number-Class) :
1-red soil
2-cotton crop
3-grey soil
4-damp grey soil

5-soil with vegetation stubble
6-mixture class (all types present)
7-very damp grey soil
There are no examples with class 6 in this dataset.

The data is given in random order and certain lines of data have been removed so you cannot reconstruct the original image from this dataset.

In each line of data the four spectral values for the top-left pixel are given first followed by the four spectral values for the top-middle pixel and then those for the top-right pixel, and so on with the pixels read out in sequence left-to-right and top-to-bottom. Thus, the four spectral values for the central pixel are given by attributes 17,18,19 and 20. If you like you can use only these four attributes, while ignoring the others. This avoids the problem which arises when a 3x3 neighbourhood straddles a boundary.

NUMBER OF EXAMPLES 6435

NUMBER OF ATTRIBUTES 36 (= 4 spectral bands x 9 pixels in neighbourhood)

ATTRIBUTES The attributes are numerical, in the range 0 to 255.

CLASS There are 6 decision classes : 1,2,3,4,5 and 7. There are no examples with class 6 in this dataset. They have all been removed because of doubts about the validity of this class.

AUTHOR Ashwin Srinivasan
Department of Statistics and Data Modeling
University of Strathclyde

Glasgow, Scotland, UK
ross@uk.ac.turing

9.2 Jeu de données LETTERS

SOURCE INFORMATION Letter Image Recognition Data
Creator : David J. Slate
Odesta Corporation ; 1890 Maple Ave ; Suite 115 ; Evanston, IL 60201
Donor : David J. Slate (dave@math.nwu.edu) (708) 491-3867
Date : January, 1991

PAST USAGE P. W. Frey and D. J. Slate (Machine Learning Vol 6, 2 March 1991) : "Letter Recognition Using Holland-style Adaptive Classifiers".

The research for this article investigated the ability of several variations of Holland-style adaptive classifier systems to learn to correctly guess the letter categories associated with vectors of 16 integer attributes extracted from raster scan images of the letters. The best accuracy obtained was a little over 80%. It would be interesting to see how well other methods do with the same data.

RELEVANT INFORMATION The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. We typically train on the first 16000 items and then use the resulting model to predict the letter category for the remaining 4000. See the article cited above for more details.

NUMBER OF INSTANCES 20000

NUMBER OF ATTRIBUTES 17 (Letter category and 16 numeric features)
1. lettr capital letter (26 values from A to Z)
2. x-box horizontal position of box (integer)
3. y-box vertical position of box (integer)
4. width width of box (integer)
5. high height of box (integer)
6. onpix total on pixels (integer)
7. x-bar mean x of on pixels in box (integer)
8. y-bar mean y of on pixels in box (integer)
9. x2bar mean x variance (integer)
10. y2bar mean y variance (integer)
11. xybar mean x y correlation (integer)
12. x2ybr mean of x * x * y (integer)
13. xy2br mean of x * y * y (integer)
14. x-ege mean edge count left to right (integer)
15. xegvy correlation of x-ege with y (integer)
16. y-ege mean edge count bottom to top (integer)
17. yegvx correlation of y-ege with x (integer)

CLASS DISTRIBUTION 789 A, 766 B, 736 C, 805 D, 768 E, 775 F, 773 G, 734 H, 755 I, 747 J, 739 K, 761 L, 792 M, 783 N, 753 O, 803 P, 783 Q, 758 R, 748 S, 796 T, 813 U, 764 V, 752 W, 787 X, 786 Y, 734 Z.

9.3 Jeu de données AUTOS

SOURCE INFORMATION 1985 Auto Imports Database
Creator/Donor : Jeffrey C. Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu)

Date : 19 May 1987
Sources :
1) 1985 Model Import Car and Truck Specifications, 1985 Ward's Automotive Yearbook.
2) Personal Auto Manuals, Insurance Services Office, 160 Water Street, New York, NY 10038
3) Insurance Collision Report, Insurance Institute for Highway Safety, Watergate 600, Washington, DC 20037

PAST USAGE Kibler, D., Aha, D. W., & Albert, M. (1989). Instance-based prediction of real-valued attributes. Computational Intelligence, 5, 51–57.
Predicted price of car using all numeric and Boolean attributes
Method : an instance-based learning (IBL) algorithm derived from a localized k-nearest neighbor algorithm. Compared with a linear regression prediction...so all instances with missing attribute values were discarded. This resulted with a training set of 159 instances, which was also used as a test set (minus the actual instance during testing).
Results : Percent Average Deviation Error of Prediction from Actual
11.84% for the IBL algorithm
14.12% for the resulting linear regression equation

RELEVANT INFORMATION This data set consists of three types of entities : (a) the specification of an auto in terms of various characteristics, (b) its assigned insurance risk rating, (c) its normalized losses in use as compared to other cars. The second rating corresponds to the degree to which the auto is more risky than its price indicates. Cars are initially assigned a risk factor symbol associated with its price. Then, if it is more risky (or less), this symbol is adjusted by moving it up (or down) the scale. Actuarians call this process "symboling". A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe.

The third factor is the relative average loss payment per insured vehicle year. This value is normalized for all autos within a particular size classification (two-door small, station wagons, sports/speciality, etc...), and represents the average loss per car per year. Note : Several of the attributes in the database could be used as a "class" attribute.

NUMBER OF INSTANCES 205

NUMBER OF ATTRIBUTES 26 total (15 continuous/1 integer/10 nominal)

ATTRIBUTE INFORMATION (Attribute : Attribute Range)
1. symboling : -3, -2, -1, 0, 1, 2.
2. normalized-losses : continuous from 65 to 256.
3. make : alfa-romero, audi, bmw, chevrolet, dodge, honda, isuzu, jaguar, mazda, mercedes-benz, mercury, mitsubishi, nissan, peugot, plymouth, porsche, renault, saab, subaru, toyota, volkswagen, volvo.
4. fuel-type : diesel, gas.
5. aspiration : std, turbo.
6. num-of-doors : four, two.
7. body-style : hardtop, wagon, sedan, hatchback, convertible.
8. drive-wheels : 4wd, fwd, rwd.
9. engine-location : front, rear.
10. wheel-base : continuous from 86.6 120.9.
11. length : continuous from 141.1 to 208.1.
12. width : continuous from 60.3 to 72.3.
13. height : continuous from 47.8 to 59.8.
14. curb-weight : continuous from 1488 to 4066.
15. engine-type : dohc, dohcv, l, ohc, ohcf, ohcv, rotor.
16. num-of-cylinders : eight, five, four, six, three, twelve, two.
17. engine-size : continuous from 61 to 326.
18. fuel-system : 1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi.

19. bore : continuous from 2.54 to 3.94.
20. stroke : continuous from 2.07 to 4.17.
21. compression-ratio : continuous from 7 to 23.
22. horsepower : continuous from 48 to 288.
23. peak-rpm : continuous from 4150 to 6600.
24. city-mpg : continuous from 13 to 49.
25. highway-mpg : continuous from 16 to 54.
26. price : continuous from 5118 to 45400.

9.4 Jeu de données HYPOTHYROID

AUTHOR Ross Quinlan

DATA SET INFORMATION From Garavan Institute
Documentation : as given by Ross Quinlan
6 databases from the Garavan Institute in Sydney, Australia
Approximately the following for each database :
2800 training (data) instances and 972 test instances
29 or so attributes, either Boolean or continuously-valued
No missing values

ATTRIBUTE INFORMATION :
age : continuous.
sex : M,F.
on_thyroxine : f,t.
query_on_thyroxine : f,t.
on_antithyroid_medication : f,t.
thyroid_surgery : f,t.
query_hypothyroid : f,t.

query_hyperthyroid : f,t.
pregnant : f,t.
sick : f,t.
tumor : f,t.
lithium : f,t.
goitre : f,t.
TSH_measured : f,t.
TSH : continuous.
T3_measured : f,t.
T3 : continuous.
TT4_measured : f,t.
TT4 : continuous.
T4U_measured : f,t.
T4U : continuous.
FTI_measured : f,t.
FTI : continuous.
TBG_measured : f,t.
TBG : continuous.

RELEVANT PAPERS :

Quinlan,J.R., Compton,P.J., Horn,K.A., Lazurus,L. (1986). Inductive knowledge acquisition : A case study. In Proceedings of the Second Australian Conference on Applications of Expert Systems. Sydney, Australia.
Quinlan,J.R. (1986). Induction of decision trees. Machine Learning, 1, 81–106.

Bibliographie

[AIS93] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *ACM SIGMOD Internationnal Conference on Management of Data*, pages 207–216, 1993.

[Bes06] Philippe Besse. *Data mining II. Modélisation Statistique & Apprentissage.* UPS. Université Paul Sabatier, Toulouse 3. LSP. Laboratoire de statistique et probabilités. France, Octobre 2006.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, CA, 1984.

[BGRS99] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful ? *Lecture Notes in Computer Science*, 1540 :217–235, 1999.

[BM98] E. Bloedorn and R. Michalski. *Data-driven constructive induction*, volume 13 of *Trans. on Intelligent systems.* 1998.

[Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2) :123–140, 1996.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1) :5–32, 2001.

[BSGR03] Ricardo Barandela, José Salvador Sánchez, Vicente García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3) :849–851, 2003.

[BVSF04] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri. The imbalanced training sample problem : under or over sampling ? In A. Fred, T. Caelli, R.P.W. Duin, A. Campilho, and D. de Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 3138 of *Lecture Notes in Computer Science*, pages 806–814, 2004.

[BWHY05] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods : a survey and categorisation. *Information Fusion*, 6(1) :5–20, 2005.

[BWY98] Liu Bing, Hsu Wynne, and Ma Yiming. *Knowledge Discovery and Data Mining*, chapter Integrating classification and association rule mining, pages 80–86. 1998.

[Car92] C. Carpineto. Trading off consistency and efficiency in version-space induction. In *Proceedings of Ninth International Machine Learning Conference*, Aberdeen, Scotland, 1992.

[CBHK02] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote : Synthetic minority oversampling technique. *Journal of Artificial Intelligence and Research*, 16 :321–357, 2002.

[CLB04] C. Chen, A. Liaw, and L. Breiman. Using random forest to learn imbalanced data. 2004.

[CNM04] R. Caruana and A. Niculescu-Mizil. Data mining in metric space : An empirical analysis of supervised learning performance criteria. In *In Proceedings of the first Workshop on ROC Analysis in AI, ROCAI'04*, pages 9–18, 2004.

[CNM06] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23th International Conference on Machine Learning, ICML'06*, pages 161–168, 2006.

[Coh60] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement.*, 20 :37–46, 1960.

[Con85] N.C Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.* Imprimerie Royale, Paris :France, 1785.

[DK01] G. Dupret and M. Koda. Bootstrap re-sampling for unbalanced data in supervised learning. *European Journal of Operational Research*, 134(1) :141–156, 2001.

[DMBMB06] T. H. Dang, C. Marsala, B. Bouchon-Meunier, and A. Boucher. Discrimination-based criteria for the evaluation of classifiers. In *Proceedings of the 7th International Conference on Flexible Query Answering Systems, FQAS'06*, volume 4027 of *LNAI*, pages 552–563, Milan, Italie, 2006.

[DRC89] G. Drastal, S. Raatz, and G. Czako. Induction in an abstract space. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, 1989.

[Faw93] T. Fawcett. Feature discovery for inductive learning. Technical report, Department of Computer and Information Science University of Massachusetts, 1993.

[Fis36] R. A. Fisher. *Annals of Eugenics*, chapter The use of multiple measurements in taxonomic problems., pages 179–188. London, 1936.

[FP97] Tom Fawcett and Foster J. Provost. Adaptive fraud detection. *Data Min. Knowl. Discov.*, 1(3) :291–316, 1997.

[FPSS96] Usama M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The kdd process for extracting usefull knowledge from volumes data. *Communication of the ACM*, 39(11) :27–34, 1996.

[Fre90] Y. Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the Workshop on Computational Learning Theory.* Morgan Kaufmann Publishers, 1990.

[GAAB+96] R. Gras, S. Ag. Almouloud, M. Bailleuil, A. Larher, M. Polo, H. Ratsimba-Rajohn, and A. Totohasina. *L'implication Statistique, Nouvelle Méthode Exploratoire de Données. Application à la Didactique, Travaux et Thèses.* La Pensée Sauvage., 1996.

[Gha00] B. Ghattas. Agrégation d'arbres de classification. *Revue de Statistique Appliquée*, 48(2) :85–98, 2000.

[GV04] H. Guo and H.L. Viktor. Learning from imbalanced data sets with boosting and data generation : the databoost-im approach. *SIGKDD Explorations*, 6(1) :30–39, 2004,.

[Har68] P. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14 :515–516, 1968.

[HB99] S. Hettich and S D Bay. The uci kdd archive, 1999.

[HBPJ96] S. Huet, A. Bouvier, M. Poursat, and E. Jolivet. Statistical tools for nonlinear regression : a practical guide with s-plus examples. *Springer series in statistics.*, 1996.

[HHM66] P. Hajek, I. Havel, and Chytil M. *Computing*, chapter The guha method of automatic hypotheses determination, pages (1) :293–308. 1966.

[HKN07] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *ICML '07 : Proceedings of the 24th international conference on Machine learning*, pages 935–942, New York, NY, USA, 2007. ACM Press.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer, 2001.

[Jap00] Nathalie Japkowicz. The class imbalance problem : Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, volume 1, pages 111–117, 2000.

[JMG95] Nathalie Japkowicz, Catherine Myers, and Mark A. Gluck. A novelty detection approach to classification. In *IJCAI*, pages 518–523, 1995.

[Kas80] G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Journal of Applied Statistics*, 29(2) :119–127, 1980.

[Kau96] Morgan Kaufmann, editor. *Experiments with a New Boosting Algorithm*, 1996.

[KB91] I. Kononenko and I. Bratko. Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6 :67–80, 1991.

[KGJV83] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. In *Science*, volume 220. 1983.

[KHM97] Miroslav Kubat, Robert C. Holte, and Stan Matwin. Learning when negative examples abound. In Maarten van Someren and Gerhard Widmer, editors, *ECML*, volume 1224 of *Lecture Notes in Computer Science*, pages 146–153. Springer, 1997.

[KHM98] Miroslav Kubat, Robert C. Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2-3) :195–215, 1998.

[KM97] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets : One-sided selection. In Douglas H. Fisher, editor, *ICML*, pages 179–186. Morgan Kaufmann, 1997.

[LBS83] P. Langley, G. L. Bradshaw, and H. Simon. *Machine learning : An artificial intelligence approach*, chapter Rediscovering chemistry with the Bacon system, pages pp. 307–330. Michalski, J. C. R. and Mitchell T., 1983.

[Leg04] G. Legrand. *Approche méthodologique de sélection et construction de variables pour l'amélioration du processus d'extraction de connaissances à partir de grandes bases de données.* PhD thesis, Université Lumière Lyon 2, 2004. Thèse de doctorat.

[LLS00] T.S. Lim, W.Y. Loh, and Y.S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40 :203–228, 2000.

[LN05] G. Legrand and N. Nicoloyannis. Data preprocessing and kappa coefficient. In Springer, editor, *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mi-*

ning, and Granular Computing, RSFDGrC'05, volume 3641 of *LNCS*, pages 176–184, Regina, Canada, 2005.

[LZG04] F. Leon, M.H. Zaharia, and D. Gâlea. Performance analysis of categorization algorithms. In *Proceedings of the 8th International Symposium on Automatic Control and Computer Science*, number ISBN 973-621-086-3, 2004.

[MA94] P. M. Murphy and D. W Aha. Uci repository of machine learning databases, 1994.

[Mat90] C. J. Matheus. Adding domain knowledge to sbl thorough feature construction. In *Proceedings of the Eighth National Conference on Articial Intelligence*, 1990.

[Mit97] T. M. Mitchell. Machine learning. In *ICML'97*, New York, 1997.

[ML77] R. S. Michalski and J. B. Larson. *ACM SIGART Newsletter*, volume vol. 63, chapter Inductive Inference of VL Decision Rules, pages pp. 38–44. 1977.

[MP91] P. M. Murphy and M. J. Pazzani. Id2-of-id3 : constructive induction of m-of-n concepts for discriminators in decision trees. In *Proceedings of the Eighth International Workshop on Machine Learning*, San Mateo, CA, 1991.

[OV93] A. L. Oliveira and A. S. Vincentelli. *Advances in neural information processing system*, chapter Learning complex boolean functions : Algorithms and applications. 1993.

[Pag89] G. Pagallo. Learning dnf by decision trees. In *Proceedings of the eleventh International Joint Conference on Artificial Intelligence*, 1989.

[PAL04] Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection : classification of skewed data. *SIGKDD Explor. Newsl.*, 6(1) :50–59, 2004.

[PFR98] F. J. Provost, T. Fawcett, and Kohavi R. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the 15th International Conference on Machine*

Learning, ICML'98, pages 445–453, San Francisco, CA, USA, 1998.

[PMM+94] Michael J. Pazzani, Christopher J. Merz, Patrick M. Murphy, Kamal Ali, Timothy Hume, and Clifford Brunk. Reducing misclassification costs. In *ICML*, pages 217–225, 1994.

[Qui86] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1 :81–106, 1986.

[Qui93] J. R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.

[RCB+79] Van Rijsbergen, Chawla, Bowyer, Hall, and Kegelmeyer. *Information Retrieval*. London, Butterworth, 1979.

[Ren88] L. A. Rendell. Learning hard concepts. In *Proceedings of The Third European Working Session on Learning*, London, 1988.

[Rit05] G. Ritschard. De l'usage de la statistique implicative dans les arbres de classification. pages 305–315, 2005.

[Sap90] G. Saporta. *Probabilités, analyse des données et statistique*. 1990.

[Sch87] J. S. Schlimmer. Incremental adjustment of representations in learning. In *Proceedings of the 4th International Conference on Machine Learning*, 1987.

[SG86] J. C. Schlimmer and R. H. Granger. *Machine Learning*, volume 1, chapter Incremental learning from noisy data, pages pp. 317–354. 1986.

[Shi99] Y.-S. Shih. Families of splitting criteria for classification trees. *Statistics and Computing*, 9(4) :309–315, novembre 1999.

[Ten07] M. Tenenhaus. *Statistique : Méthodes pour décrire, expliquer et prévoir*. 2007.

[Tom76] I. Tomek. Two modifications of cnn. *IEEE Transactions on Systems, Man and Cybernetics*, 6 :769–772, 1976.

[Vap95] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA, 1995.

[VC07] Florian Verhein and Sanjay Chawla. Using significant, positively associated and relatively class correlated rules for associative classification of imbalanced datasets. In *ICDM*, pages 679–684. IEEE Computer Society, 2007.

[VM02] G. Valentini and F. Masulli. Ensembles of learning machines. In M. Marinaro and R. Tagliaferri, editors, *Neural Nets WIRN Vietri-02*, LNCS. Springer-Verlag, 2002.

[Weh96] Louis Wehenkel. On uncertainty measures used for decision tree induction. In *Proceedings of the International Congress on Information Processing and Management of Uncertainty in Knowledge based Systems, IPMU96*, pages 413–418, Granada, 1996.

[Wei03] Gary Mitchell Weiss. *The effect of small disjuncts and class distribution on decision tree learning*. PhD thesis, Rutgers University, New Brunswick, NJ, USA, 2003. Director-Haym Hirsh.

[Wei04] Gary M. Weiss. Mining with rarity : a unifying framework. *SIGKDD Explorations*, 6(1) :7–19, 2004.

[Wil72] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, 2 :408–420, 1972.

[WP01] Gary M. Weiss and Foster Provost. The effect of class distribution on classifier learning. Technical report, Department of Computer Science, Rutgers University, January 2001.

[WP03] Gary M. Weiss and Foster J. Provost. Learning when training data are costly : The effect of class distribution on tree induction. *J. Artif. Intell. Res. (JAIR)*, 19 :315–354, 2003.

[YBR91] D. Yang, G. Blix, and E. Rendell. A scheme for construction and comparison of empirical methods. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 1991.

[ZR02] D.A. Zighed and R. Rakotomalala. *Data Mining*, volume H3 744 of *Techniques de l'ingénieur*, pages 1–26. Editions Techniques de l'Ingénieur, 2002.