—

GRIMAAG

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
Université Montpellier II

Phénomènes de Complexité et Concentration en Classification ; Applications en Apprentissage Automatique, Data Mining, et Analyse d'Images

MÉMOIRE

présenté et soutenu publiquement le 16/12/2002

pour l'obtention de l'

Habilitation à Diriger les Recherches
de l'Université Montpellier II

(Spécialité Informatique)

par

Richard Nock

Composition du jury

Rapporteurs :	François Denis	Professeur	Université de Provence	Marseille
	Jean-Pierre Nadal	DR CNRS	Ecole Normale Supérieure	Paris
	Lorenza Saitta	Professeur	Université du Piémont Oriental	Italie
Examinateurs :	Ahmed Ainouche	Professeur	Université Antilles-Guyane	Martinique
	Colin de la Higuera	Professeur	Université Jean Monnet	St-Etienne
	Olivier Gascuel	DR CNRS	LIRMM	Montpellier
	Michel Habib	Professeur	LIRMM & Univ. Montpellier II	Montpellier

Habilitation préparée au sein du
Groupe de Recherche en Informatique et Mathématiques Appliquées des Antilles-Guyane

Mis en page avec la classe thloria.

Remerciements

Un travail d'HDR est un travail d'équipe, au sens logistique du terme. A ce titre, j'ai une dette proprement inestimable envers les personnes qui m'ont permis de faire mes travaux de recherche dans les meilleurs conditions, ici, sur le campus de Schoelcher.

Au départ, il y a eu Serge Lapiquonne, qui m'a proposé cet environnement de travail, dans lequel tant et tant de choses se sont passées. La Martinique, ça n'est pas seulement de l'eau, du sable, du sel et du soleil. Derrière cette carte postale édulcorée et la photo contractuelle de ma courte aventure tropicale, il y a une aventure humaine. Cette dernière, par contre, sort largement des sentiers battus, ceux des poncifs usuels, sur cette Belle île. C'est une bonne chose. Après, ce sont aussi d'autres personnes qui m'ont apporté leur soutien, leur volonté, leur compétence, leur amitié ou tout simplement leurs encouragement. Celles-ci comptent en premier lieu les membres de notre centre de ressources en informatique de la Martinique, et leur directeur, Olivier Portecop.

Les remerciements sont aussi le seul moment d'un tel mémoire où l'on peut dérouler une histoire sans avoir à en répertorier des théorèmes, des preuves, des articles, même si, en définitive, tout comme dans des articles, beaucoup d'histoires commencent par des problèmes ! Par bonheur, se calquent sur mon histoire Antillaise certains des encouragements les plus sincères, voire les plus involontaires, qu'il soit donné de recevoir. Ces courts moments de vie, ce sont autant d'instants gravés, d'où surgissent au gré de mon histoire post-métropolitaine, des visages, des attitudes, des fou-rires, parfois aussi des moments de tristesse ou des regrets, des circonstances, des prénoms : Thrycia, Suzy, Marie-Jo, Lise, José, Louis-Félix, Line, Myriam, Florence, Pascal, Régine, Michaëlle, Max-Henri, Luc, Michel.

Chemin faisant le long de ces quatre années de travail que je déroule encore une fois, je repense à deux de mes collègues les plus importants : Ahmed Ainouche et Abdellatif Moudafi. Leur comportement, tant humain que professionnel, fut exemplaire et précieux sur bien des points, si bien qu'à travers leurs conseils patients et avisés, ce furent parfois plus que des collègues qui m'ont aidé à surmonter certaines difficultés de mon expérience, bien petite par rapport à la leur. Puisse cette contribution personnelle à la recherche en être un remerciement sincère.

Si ce mémoire d'habilitation a pu prendre forme, c'est aussi grâce à une personne qui n'a jamais cessé de m'encourager, malgré la distance : Michel Habib. Avec lui, ce furent aussi quelques autres personnalités ou anciens du LIRMM dont les encouragements n'ont jamais failli : Christophe, Colin, Gilles, Pascal, Thérèse. Trois chercheurs de renom ont également accepté de juger ce travail : François Denis, Jean-Pierre Nadal et Lorenza Saitta. Je les en remercie chaleureusement, car la tâche est, je crois, très loin d'être négligeable, surtout lorsque comme eux, on s'en acquitte avec autant de professionnalisme et de rigueur.

J'ai essayé de mettre un peu du meilleur de moi-même dans ce mémoire. C'est pourquoi je le dédie aux quelques personnes qui m'ont aidé à le donner. Quelques-unes, trop nombreuses, sont aujourd'hui disparues. Deux d'entres-elles me manquent tout particulièrement : ma mère et ma grand-mère. Ce mémoire, c'est aussi un peu du souvenir du temps qui m'en sépare, qui me rappelle que ce mémoire ne s'est pas construit sur du vide ou sur un hasard de l'existence. J'ai eu la chance de partager le chemin de quelques personnes de grande valeur. Outre les membres de ma famille (Agnès, Malika et David en tête), j'ai fait la rencontre de trois personnes sans qui, j'en suis persuadé, je n'aurais jamais eu, indirectement, la chance de faire cette habilitation, puisque je n'aurais peut-être jamais eu la volonté de débuter une thèse : Andrée Durbec, Anne Lomascolo et Gérald Mongin. A leurs différentes manières pour chacun, leur soutien fût, j'en suis sûr, indissociable de la réussite dans mes études supérieures, et il y aura encore pendant longtemps derrière ma volonté, un peu de leur influence.

Merci.

A ma Mère.
A ma Grand-Mère.

Table des matières

Introduction

"Life is just a long random walk. Things are created because the circumstances happen to be right. More often than not, creations [...] are accidental". Ces phrases sont les toutes premières du livre de Devroye, Györfi et Lugosi, "A Probabilistic Theory of Pattern Recognition" [Devroye et al., 1996]. Le contenu de ce mémoire d'Habilitation tout entier en est l'illustration. J'éviterai d'ailleurs de m'étendre sur son titre : faire plus court eut été une gageure, je crois. Il ne manquerait sans doute que quelques ratons laveurs pour en faire un inventaire à la Prévert, mais ma thèse fût, à sa manière, déjà un inventaire, même si le manuscrit n'y fait aucunement référence : une partie de mes travaux de 1998, n'y figure en effet pas. A la liste des travaux que j'ai entrepris depuis l'obtention de ma thèse, s'ajoute donc dans cette Habilitation une partie de ceux qui ont occupé mon temps pendant la fin de ma thèse elle-même. Comment chercher à caractériser les problèmes auxquels je me suis intéressé ?

En fait, il s'agit systématiquement de problèmes de classification. Qu'il s'agisse en effet de classer des exemples, d'interpréter des formules, de réduire des jeux de données, ou bien de trouver des objets dans une image, il y a toujours, sous-jacent, le problème du partitionnement d'un ensemble d'objets en un nombre de "classes" plus ou moins réduit. Cette passion pour la classification au sens très large du terme est survenue alors que je me préparais à faire ma spécialisation de troisième année d'Ingénieur Agronome. Je devais opter pour un DEA de Biostatistiques, mais ce fut un conseil très avisé de Gilles Caraux qui m'orienta vers un DEA d'Informatique, celui du Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM). J'avais opté, avant mon DEA, pour des modules optionnels approfondis en statistiques, aussi un tel choix d'orientation pourrait sembler étrange, mais le DEA d'Informatique fut l'occasion de découvrir de nouveaux aspects de la classification, et en particulier, un champ d'action s'étendant bien au delà des problèmes que je connaissais : analyse d'images, apprentissage automatique, Data Mining, etc. . J'eus par la suite la chance, durant mon service militaire, de travailler dans un centre d'expertise de la DGA (alors sous la direction du médecin Général Jean-Paul Papin) qui se focalisait entr'autre sur des problèmes d'évaluation des individus, un autre domaine d'investigation idéal pour la classification. Depuis, mon intérêt pour cette discipline, au sens le plus large, n'a cessé de croître, avec la reconnaissance d'un de ses volets industriels les plus importants, issu en partie de l'apprentissage automatique : le Data Mining. La métaphore la plus répandue pour présenter ce domaine est celle de la mine d'or, le chercheur étant celui qui construit les bulldozers, les grues permettant de fouiller cette mine de données digitales pour en trouver les "pépites", l'information qu'elle contient. A la différence des mines d'or traditionnelles cependant, le Data Mining fait face à un phénomène tout à fait original : la montagne à fouiller est susceptible de doubler de taille tous les neuf mois, et ceci depuis dix ans maintenant. Quand on regarde l'évolution de cette discipline depuis les premiers projets réels de Data Mining (qui datent environ de cette époque), on s'aperçoit qu'il y a un gouffre s'accroissant entre les quantités de données qui peuvent être générées, et nos compétences pour les analyser. Les besoins en outils de plus en plus puissants pour analyser et comprendre vont donc certainement s'accroître encore d'avantage. Le Data Mining n'est, bien évidemment, pas le seul domaine qui soit mis à rude épreuve par l'augmentation de capacité (de stockage ou de traitement) des ordinateurs. Un autre domaine se trouve être celui du traitement d'images, et surtout de la segmentation d'images, domaine qui a occupé une partie significative de mon temps de recherche. Au travers des premiers travaux que j'ai effectué avec Christophe Fiorio, ce qui m'a attiré dans ce domaine, et m'a poussé à en faire un de mes sujets de recherche, c'est aussi, d'une certaine manière, la disponibilité croissante d'une quantité d'information industrielle et facile à saisir, le besoin de pouvoir analyser cette montagne d'information de plus en plus rapidement pour en déterminer les objets constituants, tout en essayant de conserver une qualité maximale au résultat. Sur bien des points, la segmentation d'images partage ses problématiques avec le Data Mining.

Je me suis focalisé autour de deux grands types d'outils pour étudier tous ces problèmes de classification. Le premier, ce fut l'analyse de la complexité de résolution dans le pire des cas, autrement dit, en étant quelque peu rude, la NP-Complétude et ce qui en découle (l'étude de la difficulté d'approximation, et des liens entre NP et des classes de complexité généralisant la classe P, soit par la complexité en temps, soit par l'aspect de randomisation). Lorsque j'ai commencé cette Habilitation, beaucoup de problèmes d'apprentissage pour lesquels on conjecturait des résultats négatifs de complexité n'étaient pas encore résolus (beaucoup ne le sont d'ailleurs toujours pas). La théorie de la complexité structurelle, c'était sans doute un moyen original de s'attaquer à l'apprentissage automatique, *i.e.* un domaine dont la vocation est, après tout, de prouver l'existence d'algorithmes d'apprentissage efficaces, donc des bornes supérieures, et moins d'exhiber des bornes inférieures les plus grandes possibles ! C'était en tout cas un moyen plus facile que d'autres pour entrer dans ce domaine, surtout vues mes performances pitoyables en programmation au début de ma thèse (Christian Capelle s'en souvient sans doute encore). C'était même d'autant plus intéressant que les résultats qui allaient être utilisables, en théorie des graphes par exemple, étaient à peine sortis à l'époque, et l'on ne disposait pas encore des grands résultats d'inapproximabilité "probablement approximativement optimaux" de Håstad *et al.* autour de ces problèmes. Je ne fus d'ailleurs pas déçu de mon approche (on s'en rendra compte dans la première partie) : les résultats se sont rapidement accumulés, tous plus négatifs les uns que les autres. Aux articles ou se trouvent des résultats sur l'approximabilité de la formule minimale consistante, se succédèrent des résultats sur l'approximabilité de l'erreur minimale de formules de taille contrainte, puis des résultats sur la difficulté de remplacement des critères traditionnelles utilisés en classification, etc, etc. . Il ne se passait de mois sans qu'un résultat négatif ne tombe, sans qu'un théorème ne vienne "améliorer" un dernier résultat négatif obtenu. Durant les périodes ou je programmais, ce genre d'activité était d'ailleurs devenu une sorte de passe-temps : dès

que je m'escrimais trop à trouver des bugs dans mes programmes, je m'arrêtais parfois, le temps d'améliorer un de mes résultats négatifs ! Pourrait-il y avoir en effet de vengeance plus douce, devant un code refusant obstinément de marcher pour des raisons défiant la plus ordinaire des logiques, que de lui montrer que de toute façon, même quand il marchera, il existera toujours des cas pathologiques devant lesquels il ne pourra rendre qu'un résultat médiocre, voire à peine meilleur que des solutions triviales consistant à ne pas même l'exécuter ! C'était effectivement amusant, et sans doute la raison pour laquelle ces résultats ont été nombreux (!), mais souvent très localisés dans le temps. En effet, je ne me suis jamais acharné durant de longue périodes à démontrer des résultats négatifs d'approximabilité. Jamais, sauf pour l'un d'entre eux.
Un seul problème m'occupe en effet depuis le début (et encore aujourd'hui lorsque j'ai des bugs dans mes programmes...), à intervalles réguliers, et m'a aidé à trouver des outils pour résoudre une partie des autres : l'approximabilité de DNF, l'un des grands problèmes autour du modèle d'apprentissage PAC de [Valiant, 1985].
Les méthodes que j'ai développé autour de ce problème m'ont servi dans d'autres problèmes qui ne sont pas forcément du ressort direct du modèle PAC, mais elles ont quasiment toutes un point commun, qui est un parallèle à mon sens amusant avec une des techniques les plus puissantes en classification : le Boosting [Freund and Schapire, 1997]. Le Boosting est une méthode permettant de combiner de manière extrêmement puissante des *solutions* d'algorithmes d'apprentissage (au sens d'un modèle plus faible que PAC) pour en tirer une formule de beaucoup plus grande taille, mais qui est beaucoup plus fiable que n'importe laquelle de ses formules composantes. Si je présente ainsi cette technique de combinaison de résultats positifs dans un paragraphe consacré à des résultats négatifs, c'est tout simplement parce que la technique qui m'a servie à obtenir mes résultats négatifs les plus forts procède de la même manière, mais elle combine des *instances* de problèmes difficiles reliés à l'apprentissage, d'une manière telle que l'instance résultante (du *même* problème), de taille raisonnablement plus grande, soit cette fois d'une difficulté à résoudre sans commune mesure avec n'importe laquelle de ses composantes. Ces résultats font l'objet de la première partie de ce mémoire.

Si l'apprentissage automatique fait la part "belle" à ce genre de résultats pour la partie algorithmique, il dispose d'une toute autre boite à outils pour étudier les capacités des algorithmes en généralisation : les phénomènes de concentration de variables aléatoires, dont les plus connus se trouvent dans les inégalités de Chernoff-Hoeffding. Si j'ai utilisé avec une parcimonie relative ce genre d'outils en apprentissage et Data Mining (j'étais sans doute trop occupé par ma quantité de résultats négatifs d'apprenabilité ou d'approximabilité), ce fut par contre une de mes préoccupations que de les utiliser dans un domaine de la classification où je les avait peu rencontrées : la segmentation d'images. Dès le fin de ma thèse, j'avais commencé à poursuivre quelques directions de recherche avec Christophe Fiorio, du LIRMM, sur de nouveaux algorithmes de segmentation possédant à la fois une complexité en temps et/ou espace presque optimale (voire optimale), et surtout permettant d'obtenir un minimum de propriétés (mêmes probabilistes) sur la segmentation finale. Christophe possédait à l'époque une structure de données perfectionnée, qui rendait tout algorithme de segmentation très rapide (*i.e.* beaucoup plus rapide que beaucoup de résultats que j'ai pu rencontrer dans la littérature consacrée), pourvu qu'on sache rapidement répondre au test de fusion de deux régions. Ayant déjà la structure de données, le code, et l'algorithme, il ne me restait plus qu'à trouver la pièce manquante : ce fameux test. Ce fut un terrain presque idéal pour expérimenter l'utilisation de quelques-uns des derniers résultats sur les phénomènes de concentration de variables aléatoires. En 1998, juste avant mon départ pour les Antilles, Michel Habib organisa une école autour des algorithmes randomisés et autres outils probabilistes de la conception et de l'analyse d'algorithmes. L'article de Colin McDiarmid, "Concentration" ([McDiarmid, 1998]) fut un déclic : il présentait clairement les tout derniers résultats en la matière. Avec Christophe, nous commençâmes à tester ces bornes, avec de bons résultats en général. Il ne me manquait plus, pour ainsi dire, qu'un lien avec l'apprentissage. Ce fut une courte discussion avec M. J. Kearns lors de ICML/COLT/ILP/UAI/AAAI'98 à Madison qui m'en donna l'idée. Dans un article qu'il présentait avec Y. Mansour, Il démontrait un résultat particulièrement élégant, permettant de savoir comment élaguer un arbre de décision avec une précision statistique redoutable. Chose amusante : ces résultats étaient en réalité difficilement utilisables en pratique[1] dans le cas des arbres de décision (!), mais il devait être possible de se servir des idées dans le domaine de la segmentation d'images, avec sans doute des raisons d'espérer que les idées s'adaptent mieux. Ce fut une de mes lignes directrices de recherche, exposée en détail dans la troisième partie.

Et puis, il y eut la plus grosse partie de toutes, celle qui s'est doucement élaborée depuis la fin de ma thèse, puis a littéralement explosé à la faveur d'une collaboration sans faille avec un collègue rencontré en Guadeloupe, Marc Sebban, autour d'un travail qu'on pourrait aisément qualifier "de dingues" lorsqu'on le regarde rétrospectivement. Cette partie concerne tous les algorithmes que j'ai étudié, autour de l'induction de concepts, depuis les algorithmes de réduction de données jusqu'aux algorithme d'induction de nouvelles classes de représentation de concepts. J'avais commencé à étudier les algorithmes d'induction durant ma thèse (mon premier article porte précisément sur ce sujet, [Nock and Gascuel, 1995]). Mon objectif était à l'époque de construire des formules faisant partie d'une nouvelles classe de représentation de concepts, qui avait été appelée les "comités de décision". Ce travail s'est ensuite étendu en fin de thèse aux listes de décision, et plus récemment à diverses classes, mais en utilisant un type d'algorithme "normalisé", car relativement indépendant du formalisme utilisé. L'idée était simple : utiliser et adapter les derniers résultats théoriques de Kearns *et al.* autour de l'induction et de l'élagage de certaines classes de représentation de concepts (surtout les arbres de décision). Ma collaboration avec Marc Sebban a vu mes activités autour de ce sujet "Boostées" tant nous avions d'idées autour de deux domaines de l'apprentissage automatique et du Data Mining qui ont reçu tout récemment un regain d'intérêt : la sélection de variables et de prototypes, c'est à dire des algorithmes permettant de simplifier des jeux de données sans perdre en capacité de généralisation (parfois même, en y gagnant). Notre collaboration n'en a été que plus fructueuses, car nous arrivions en plus avec des idées largement complémentaires et différentes, sur des sujets communs. Marc apportait sa connaissance du traitement et de la simplification des structures de voisinage à l'aide de critères informationnels, j'apportais ma connaissance des méthodes d'induction telles le Boosting, avec la

[1] En réalité, ça ne marchait pas du tout...

volonté d'en faire autre chose qu'un (excellent) algorithme d'induction, mais bel et bien une procédure puissante permettant aussi de réduire un jeu de données en en supprimant soit des variables, soit des exemples. Notre collaboration a montré l'intérêt de ces deux grands types d'approches. Cette partie étant sans aucun doute centrale, elle se trouve entre les deux précédentes. Il s'agit également de la thématique dans laquelle s'insère Patrice Lefaucheur, qui vient tout juste de s'inscrire en thèse sous la direction officielle d'Ahmed Ainouche, et sous mon encadrement. Ses travaux d'investigation portent sur des résultats récents autour des algorithmes de Boosting, afin de construire des procédures particulièrement rapides d'induction et de fouille de données sur de grandes bases de données. Ses travaux sont d'ailleurs co-supportés par des partenaires industriels Martiniquais des domaines de l'assurance et du *credit management*.

Enfin, tout le travail exposé dans cette Habilitation a donné lieu a plusieurs dizaines de publications (au moment de sa rédaction) internationales, référées, éditées, et publiées internationalement (par AAAI Press, Elsevier, IEEE Press, Kluwer, MIT Press, Morgan Kaufmann, Springer, etc.). Sachant qu'un article est, par essence, policé, "optimisé", et *self-contained* dans sa présentation (aux écarts éventuels des *sketches* de preuves), j'ai choisi de garder une taille raisonnable au mémoire (une quarantaine de pages), tout en incluant suffisamment de mes publications en annexes (une quinzaine), pour permettre au lecteur de trouver les idées originales, les preuves, les algorithmes, les expérimentations, sans surcharger inutilement le corps de l'exposé. Ceci implique aussi que, si toutes les publications sont référencées (pages 234-238), certaines ne sont que brièvement citées dans le texte, et leurs résultats ne sont parfois pas même exposés. J'ai fait le choix de privilégier un corps de texte uniforme, plutôt que d'en faire, en quarante pages environ, un catalogue d'abstracts. Je me suis juste laissé aller à ajouter dans la troisième partie de ce mémoire quelques images issues de mes derniers algorithmes de segmentation. Quelques-unes de ces magnifiques images ont été saisies par Pascal Jappy, et réutilisées avec sa permission, cela va de soi.

Première partie

Quelques aspects algorithmiques des problèmes de classification

1 L'apprentissage au sens de Valiant

"L'apprentissage au sens de Valiant" est un des piliers de l'apprentissage automatique. Ce modèle -car il s'agit d'un modèle d'apprenabilité- est d'une simplicité extrême dans sa formulation (il répond lui même déjà en quelque sorte au principe de parcimonie d'Occam, voir plus loin), et pourtant il étudie l'un phénomènes des plus complexes : la capacité d'apprentissage d'un système, qu'il soit biologique ou non [Valiant, 1984]. Loin de moi l'idée de discuter des implications philosophiques, physiques, ou psychologiques de ce genre de modèle : ceci dépasse largement le cadre de cette Habilitation, et d'autres l'ont très bien fait avant [Boucheron, 1992]. Il serait de même certainement trop long et fastidieux d'en faire un historique ne serait-ce que probablement approximativement exhaustif : l'article fondateur du modèle PAC (pour *P*robablement *A*pproximativement *C*orrect), [Valiant, 1984], est déjà cité près de quatre-cent fois dans les articles recensés par le `citeseer` [2]. Il s'agit certainement d'une borne très inférieure au nombre réel de citations de cet article. Cette première partie de mon Habilitation n'a d'autre ambition que de présenter la version du modèle couramment acceptée comme étant "la première", ainsi que mes contributions formelles dans ce modèle, et autour de quelques variantes développés depuis. Il est par contre utile de souligner que les résultats obtenus aujourd'hui autour de PAC vont certainement bien au delà des aspirations même des pionniers. En 1988, par exemple, Kearns lui-même doutait quelque peu de l'applicabilité des algorithmes de Boosting [Kearns, 1988], proposés pour renforcer les résultats positifs autour de PAC, alors qu'il suggérait déjà un des principes fondamentaux de ce qui allait être, à peine 10 ans plus tard le classifieur sur étagère jugé "le plus puissant au monde" [Freund and Schapire, 1997, Friedman et al., 2000].

En apprentissage au sens de PAC, l'objectif est d'inférer un concept à partir d'exemples. En fait de concept, il s'agirait d'avantage de la représentation de ce concept, puisque le concept lui-même est une notion ensembliste, caractérisant par exemple les clients d'un site Web qui achètent contre ceux qui passent sans s'arrêter [Kohavi and Brodley, 2000], ou bien encore les images de nappes de pétrole contre les artéfacts d'imagerie radar [Kubat et al., 1998]. Évidemment, il existe des exemples moins mercantiles.

Appelons l'ensemble de tous ces exemples le *domaine*, noté $\mathcal{X}$. Chaque élément de $\mathcal{X}$ est en réalité un couple, composé d'une observation et d'une classe. Chaque observation est décrite sur un ensemble de n variables (par exemple le temps de connexion, la couleur d'un pixel, etc.), et chaque classe prend ses valeurs dans $1, 2, ..., l$, avec l potentiellement beaucoup plus grand que 2. Dans les études théoriques autour de l'apprentissage automatique, on fixe généralement $l = 2$, surtout pour montrer des résultats négatifs. Dans ce cas en effet, l'objectif est généralement de montrer que la difficulté apparaît "déjà" dans les cas les plus simples possibles, et s'étend donc de fait à ceux pour lesquels $l > 2$. Les deux classes sont représentées par "$+$" et "$-$", et sont appelées respectivement la classe positive et la classe négative. La classe positive rassemble généralement tous les exemples qui font partie du concept (ou, de manière équivalente, qui en satisfont la représentation).

[2] `http ://citeseer.nj.nec.com/context/6901/0`.

L'objectif est simple : il s'agit de reconstituer aussi fidèlement que possible le découpage en classes dans $\mathcal{X}$, à l'aide d'une représentation d'un concept choisie dans une classe de représentation de concepts $\mathcal{C}$. Le modèle original de Valiant fait une hypothèse fondamentale : le concept qu'on cherche à approximer, appelé concept cible, est aussi un élément de $\mathcal{C}$. Par conséquent, la règle de Bayes, d'erreur nulle, est représentable dans $\mathcal{C}$, et toute la difficulté est alors de se rapprocher suffisamment de ce concept. L'une des classes parmi les premières à avoir été étudiées est la classe des arbres de décision [Breiman et al., 1984]. Elle reste encore l'une des plus utilisées [Kohavi and Brodley, 2000]. Avec l'avènement du Data Mining, qui introduit le décideur (Businessman ou utilisateur lambda) dans le processus d'induction, on a constaté un besoin de diversifier le nombre de classes $\mathcal{C}$ disponibles pour des raisons d'interprétabilité, et des études d'apprenabilité ou des algorithmes se sont développées autour de nouvelles classes de représentation de concepts $\mathcal{C}$ [Kohavi and Sommerfield, 1998, Nock and Gascuel, 1995, Nock and Jappy, 1998b, Rivest, 1987] (et bien d'autres).

Ce qui complique bien évidemment la tâche d'apprentissage, c'est qu'on ne dispose pas d'un accès à tout $\mathcal{X}$, bien au contraire. Les exemples de $\mathcal{X}$ sont tirés aléatoirement grâce à un objet appelé oracle [Kearns and Vazirani, 1994], qui rétribue ces exemples en accord avec une distribution de probabilité $\mathcal{D}$. Fait important, on ne connaît pas cette distribution, et on ne fait aucune hypothèse dessus, mis à part celle qu'elle ne fluctue pas (par exemple au cours du temps). L'erreur produite par l'hypothèse h construite par rapport au concept cible c est mesurée par $\mathbf{Pr}_{\mathcal{D}}[h \neq c]$, où "$(h \neq c)(x)$" est l'évènement "$h$ et c donnent une classe différente à l'exemple x". Il apparaît alors difficile d'exiger que cette erreur soit systématiquement nulle, voire même limitée par une quantité $\epsilon < 1$, même si ϵ s'approche de 1. Il sera en effet toujours possible de tirer dans les cas pathologiques des échantillons d'exemples si peu représentatifs (quelque soit le temps fini alloué) que même une hypothèse h ne faisant aucune erreur sur ces échantillons aura une erreur sur $\mathcal{X}$ plus grande que ϵ. Le modèle PAC autorise ces situations, puisqu'il requiert "seulement" que $\forall 0 < \epsilon, \delta < 1$ donnés en entrée de l'algorithme, la sortie h satisfasse $\mathbf{Pr}\left[\mathbf{Pr}_{\mathcal{D}}[h \neq c] > \epsilon\right] < \delta$.

Outre ces considérations purement statistiques, pour qu'un algorithme d'apprentissage d'une classe $\mathcal{C}$ soit "PAC", il faut en plus que sa complexité en temps soit bornée par un polynôme en les divers paramètres du problème : $n, |c|, 1/\epsilon, 1/\delta$, ou $|c|$ est la taille de codage du concept cible (s'il s'agit d'un arbre de décision, $|c|$ peut être le nombre de sommets internes, le nombre de feuilles, etc.). Bien évidemment, un algorithme est PAC pour une classe $\mathcal{C}$ ssi il satisfait aux deux exigences précédentes $\forall 0 < \epsilon, \delta < 1$, $\forall c \in \mathcal{C}$, et pour toute distribution $\mathcal{D}$ (inconnue, mais fixée).

2 La complexité de l'apprentissage...

Le modèle de Valiant a ceci de remarquable qu'il propose deux propriétés très distinctes (statistiques et algorithmiques) pour qu'un algorithme puisse être considéré comme un "apprenti" au sens de PAC. De ce fait, il propose deux moyens conceptuellement simples pour démontrer que la tâche d'apprentissage est difficile dans certains cas particuliers de classes de représentation de concepts $\mathcal{C}$. En utilisant la contrainte statistique, on peut par exemple chercher à montrer que la classe $\mathcal{C}$ est si complexe que l'apprendre nécessiterait tout simplement trop d'exemples. Typiquement, ce serait le cas si $\mathcal{C}$ représentait l'ensemble de toutes les parties d'un ensemble. Cette difficulté est bien capturée par une notion apparue bien avant le modèle de Valiant : la dimension de Vapnik-Chervonenkis (VC-dimension, [Vapnik and Chervonenkis, 1971]). Il s'agit d'un paramètre combinatoire caractérisant en quelque sorte un seuil de taille d'échantillon d'exemples m pour lequel $\mathcal{C}$ reste encore assez puissante pour en générer toutes les dichotomies possibles (sachant que $l = 2$, il s'agit de générer toutes les classifications possibles des m observations, ce qui nécessite 2^m concepts distincts). Plus ce paramètre est grand, plus $\mathcal{C}$ est puissante, mais plus il faut d'exemples pour l'apprendre : statistiquement, la difficulté de la tâche d'apprentissage augmente. Il faut noter que cette difficulté statistique reste indépendante de la ressource algorithmique de l'apprenti [Ehrenfeucht et al., 1988]. Tout un pan de la recherche en théorie de l'apprentissage s'est ainsi focalisée sur le calcul ou l'approximation de cette VC-dimension pour des classes $\mathcal{C}$ très diverses [Karpinski and Macintyre, 1997, Koiran and Sontag, 1998, Wenocur and Dudley, 1981] (et bien d'autres).

Mais alors, *Quid* de l'aspect algorithmique du modèle PAC de Valiant ? Démontrer des résultats négatifs en se servant de la contrainte algorithmique du modèle, c'est essentiellement s'inspirer de la théorie de la complexité structurelle, et de résultats négatifs existant déjà sur un certain nombre de problèmes difficiles à résoudre ou à approximer. Ce système de preuve est, là encore, relativement bien codifié depuis [Kearns et al., 1987]. Il consiste, *en général*, à faire une réduction d'un problème traditionnel issu de domaines de la combinatoire, vers un problème d'apprentissage, réduction telle qu'on sait que la propriété de NP-Complétude ou bien le ratio d'inapproximabilité du premier problème va se transmettre vers le second, moyennant un effort de démonstration plus ou moins important. Évidemment, en fait de problème d'apprentissage, on devrait d'avantage parler de problème de décision ou d'optimisation *relié* à l'apprentissage PAC ; on n'étudie pas directement un résultat négatif d'apprentissage PAC. Pour passer à ce dernier résultat, il faut généralement faire l'effort d'une randomisation de l'hypothèse de complexité sur le problème de décision ou d'optimisation, mais celle-ci n'est pas gênante, puisqu'elle affecte en réalité à peine la force de l'hypothèse (voir par exemple un des premiers articles en la matière, [Kearns et al., 1987]).

Le problème de ce genre de réduction, c'est qu'il n'aboutit en général qu'à des réponses négatives très partielles pour l'apprentissage PAC, surtout lorsqu'on veut considérer les classes les plus générales de formalismes de représentation. Par exemple, la réduction de [Hyafil and Rivest, 1976] peut très bien s'étendre à une propriété négative d'apprentissage PAC sur des arbres de décision, mais elle ne montre certainement pas que les arbres de décision au sens le plus général ne sont pas apprenables dans le modèle PAC. Tout au plus montre-t'elle que les arbres de décision de profondeur (très) limitée ne sont pas apprenables dans le modèle PAC. Une des leçons de mon travail d'habilitation, c'est que les résultats positifs les plus forts peuvent aussi parfois suggérer des résultats négatifs extrêmement forts. Le Boosting est une des techniques de classification les plus puissantes, qui consiste à combiner de façon subtile différentes solution d'un problème de classification, de manière à ce que cet ensemble solution soit beaucoup plus performant que n'importe laquelle de ses parties [Freund and Schapire, 1997]. Dans le cas des résultats négatifs, je me suis aperçu qu'il est parfois aussi possible de combiner diverses instances difficiles d'un problème en apprentissage, pour en faire une instance beaucoup plus difficile à approximer que n'importe laquelle de ses composantes. Ce principe m'a d'ailleurs servi non seulement dans le cadre de l'apprentissage PAC, mais aussi dans d'autres, qu'il s'agisse d'étudier les capacités d'apprentissage robuste, ou carrément la combinatoire des problèmes de réduction de données.

Avant de détailler tous les résultats que j'ai pu obtenir autour de l'application ce principe, je vais en prendre un exemple, autour d'une des conjectures négatives d'apprenabilité les plus célèbres (et les plus anciennes) dans le modèle PAC : celle de la classe DNF.

2.1 Ratios d'inapproximabilité très larges pour DNF

En 1985, Valiant étudie l'apprenabilité de la classe des Formules Normales Disjonctives [Valiant, 1985], c'est à dire des disjonctions de conjonctions. Une DNF contient des règles (aussi appelées monomes), Booléennes (chacune des n variables de description prend la valeur 0 ou 1). On dispose de deux classes ($l = 2$), la classe associée à chaque règle étant la classe positive (+). La classe négative rassemble donc sur $\mathcal{X}$ tous les exemples ne satisfaisant aucune règle du concept cible. Puisqu'on on ne place aucune contrainte sur le nombre de règles ou sur leur taille (dans DNF), alors tout concept sur $\mathcal{X}$ est exprimable par une DNF. Et même si on disposait d'un domaine $\mathcal{X}$ dans lequel l'erreur de la règle de Bayes serait non nulle, alors cette règle de Bayes elle-même s'exprimerait sous la forme d'une DNF, dans laquelle il suffirait de mettre toutes les observations pour lesquelles la classe à prédire dans le règle de Bayes est "+".

La motivation de Valiant pour l'étude de DNF a des origines computationnelles, mais aussi psychologiques :

> "[...] the attraction for this class is that humans appear to like it for representing knowledge as evidenced, for example, by the success of the production system paradigm and of Horn Clauses logics [...]"

Valiant démontra que si on limite la taille de chaque règle par une constante k, alors la classe correspondante $\mathcal{C}$, k-DNF, est apprenable au sens de PAC. L'algorithme utilisé est intéressant car c'est en fait une des premières astuces permettant de montrer l'apprenabilité PAC de classes complexes en se basant sur des classes beaucoup plus simples : l'algorithme revient en effet à apprendre une clause (une disjonction) dans un espace de description ou les n variables initiales sont remplacées par les $\mathcal{O}(n^k)$ règles possibles[3].
La question restant en suspens dans l'article de Valiant est le statut de DNF elle-même au regard de PAC. Valiant conjecture d'ailleurs qu'il serait fort possible qu'une classe

[3]Nous avons utilisé une astuce identique pour isoler une des plus grandes classes PAC apprenables [Nock and Gascuel, 1995].

telle DNF ne soit pas PAC apprenable. Depuis, une quantité impressionnante d'articles se sont focalisés autour de cette grande question de la théorie de Valiant, sans jamais arriver à une réponse pour DNF elle-même, sauf à relâcher la contrainte algorithmique ou autoriser une information plus riche que celle fournie par le tirage aléatoire des exemples, auquel cas des résultats positifs semblent effectivement abordables [Balcazar et al., 2000, Jackson, 1997, Klivans and Servedio, 2001].

Je reste persuadé que l'apprentissage PAC de DNF est une tâche difficile. Avant d'entrer dans les détails, les résultats que nous avons obtenu autour de cette problématique peuvent être formulé de trois manière différentes :

- Il ne peut exister d'algorithme PAC pour DNF de petite complexité polynômiale,
- Tout algorithme d'apprentissage PAC éventuel sortira de temps en temps des hypothèses de taille énorme par rapport au concept cible,
- Il ne peut exister d'algorithme de compression efficace de la DNF cible en une autre DNF de taille beaucoup plus grande.

En 1998, nous avons pu isoler le plus grand ratio d'inapproximabilité connu à l'époque pour DNF, en utilisant un type de réduction original [Nock et al., 1998]. Avant de l'expliquer plus en avant, il est sans doute nécessaire d'expliquer les approches traditionnelles de ce problème, et surtout leurs limites. La première réduction montrant la difficulté d'apprentissage d'une sous-classe de DNF est en réalité une des premières preuves négatives d'apprenabilité, due à [Kearns et al., 1987]. Ils montrent qu'approximer la plus petite DNF est aussi difficile qu'approximer le problème de la coloration de graphes [Garey and Johnson, 1979]. Depuis [Feige and Kilian, 1996], on sait qu'il ne peut être approximé à moins de $|X|^{1-\alpha}$, pour toute constante $\alpha > 0$, où $|X|$ est le nombre de sommets du graphe, sous une hypothèse de complexité introduisant les classes de complexité randomisées, $NP \not\subseteq ZPP$ (voir l'article [Nock et al., 1998] figurant en annexe pour les définitions de ces classes). Ceci implique qu'aucun algorithme ne peut prétendre, à partir d'un graphe dont le nombre minimal de couleurs serait k (pas nécessairement une constante), trouver une coloration en temps polynômial du graphe utilisant au plus $k|X|^{1-\alpha}$ couleurs[4]. C'est énorme, et quasiment optimal : il existe en effet un algorithme permettant d'approximer la coloration en au plus $k|X|^{1-o(1)}$ couleurs [Ausiello et al., 1999]. Le résultat correspondant qu'on obtient pour DNF est en revanche largement sous-optimal. En effet, le résultat de [Feige and Kilian, 1996] implique qu'aucun algorithme polynômial ne peut trouver une DNF consistante avec un échantillon d'exemples et contenant au plus

$$|h| \quad \leq \quad |c| \times n^{1-\alpha} \tag{1}$$

règles, où le paramètre de taille, $|c|$, est le nombre de règles de la DNF consistante minimale. La quantité de droite est aussi une borne inférieure pour la complexité en temps de tout algorithme PAC éventuel pour DNF. Or, si le ratio trivial d'approximabilité pour la coloration de graphes est $|X|$ (on donne une couleur différente à chaque sommet), il

[4]On peut même relâcher la condition d'obtention "systématique" de cette solution.

serait par contre largement exponentiel - 2^n - pour DNF (on place tous les exemples positifs dans la DNF). Ainsi, un résultat équivalent á [Feige and Kilian, 1996] postulerait par exemple que DNF n'est pas approximable à moins de 2^{n^α}, pour une constante $\alpha > 0$ potentiellement petite (il semble en effet qu'à la lecture de [Klivans and Servedio, 2001], il serait difficile de faire pire!), autrement dit, aucun algorithme ne pourrait trouver une formule consistante h satisfaisant

$$|h| \quad \leq \quad |c| \times 2^{n^\alpha} \quad . \tag{2}$$

Depuis 1993, je suis ce problème de la PAC-apprenabilité de DNF. C'est à cette époque que j'ai eu l'idée de ne plus attaquer ce genre de problèmes en apprentissage par des réductions classiques faisant intervenir un problème de complexité ordinaire (telle la colorabilité de graphes). En fait, on pourrait presque dire que le modèle d'apprentissage PAC est tellement "bien fait" que les réductions ordinaires n'égratignent qu'à peine les propriétés d'apprenabilité. Elles sont en tout cas nettement insuffisantes pour attaquer les classes les plus grandes. En quelque sorte, tout comme le modèle PAC semblait avoir besoin de techniques permettant de Booster ses résultats positifs [Kearns, 1988], il semblait que les résultats négatifs avaient aussi bien besoin de Boosting! L'idée de cette réduction est simple : il s'agit de faire une réduction à l'intérieur d'un même problème, plutôt qu'entre deux problèmes différents, en combinant des instances d'un même problème difficile de manière à avoir une augmentation brutale (exponentielle) du ratio d'inapproximabilité avec le nombre d'instances combinées.
A l'époque, j'avais commencé à faire une telle réduction pour DNF, mais je ne voyais pas exactement jusqu'où on pouvait améliorer le résultat initial de [Kearns et al., 1987], même en le combinant avec celui de [Feige and Kilian, 1996]. Il manquait en quelque sorte un catalyseur pour que la réduction ait vraiment lieu. Ce fut un article que m'envoya Stéphane Boucheron, [Hancock et al., 1996]. C'est en effet dans cet article que les auteurs montraient, entr'autre, comment "booster" des résultats d'inapproximabilité pour les arbres de décision, mais sans vraiment utiliser la puissance de ces réductions, qu'ils appelaient "Blow-up".
Dans le cas de DNF, nous avons pu montrer que sous l'hypothèse $NP \not\subset ZPP$ (la même que celle impliquant le résultat en ineq. 1), aucun algorithme polynômial n'est capable de trouver une DNF consistante avec un échantillon d'exemples et contenant au plus

$$|h| \quad \leq \quad |c|^{\frac{146}{145}-\alpha} n^\beta m^{\frac{1}{19}-\gamma} \tag{3}$$

règles, $\forall \alpha, \gamma > 0$, et *surtout* $\forall \beta > 0$. Ainsi, la borne sur l'exposant de n en ineq. 1 disparaît. Nous avons formulé ce théorème sous la forme d'une contrainte sur la taille de h pour tout *rasoir d'Occam* pour DNF. Un rasoir d'Occam est un algorithme de temps polynômial capable, à partir d'un échantillon d'exemples de taille m, de retourner une formule h consistante avec les exemples et dont la taille admet une dépendance en m de l'ordre de $m^{1-\alpha}$ ($\alpha > 0$). En d'autres termes, h "comprime" pour ainsi dire l'échantillon d'exemples, au moins pour les grands échantillons (en effet, la taille limite admet aussi un facteur polynômial en la taille du concept cible). Fait notable, en dehors de l'esthétique purement formelle du modèle PAC, il a réussi à trouver dans cette notion de compression une démonstration plaisante d'un concept philosophique datant du XIV^e siècle[5], selon laquelle les entités ne doivent pas être multipliées sans nécessité. Il s'agit là de ce qui a été formalisé sous l'expression de "rasoir d'Occam", et qui a sans doute contribué à populariser le modèle PAC malgré quelques critiques récentes [Domingos, 1998]. Fait remarquable dans le modèle PAC, [Blumer et al., 1987] ont montré qu'un rasoir d'Occam pour une classe $\mathcal{C}$ est aussi un algorithme d'apprentissage PAC pour $\mathcal{C}$. [Board and Pitt, 1992] ont montré que pour un certain nombre de classes $\mathcal{C}$, il s'agit d'une équivalence : l'existence d'un algorithme d'apprentissage implique l'existence d'un rasoir d'Occam. Pour que cette équivalence existe pour une classe $\mathcal{C}$, il suffit qu'elle vérifie une propriété dite *de clôture par liste d'exception*. Malheureusement, on ne dispose pas pour tous les cas de classes de représentation de concepts d'une telle propriété. Là encore, le statut de DNF est inconnu. L'intérêt du "boosting" des résultats négatifs tel que nous l'avons pratiqué pour DNF n'est pas seulement l'obtention de ratio d'inapproximabilité beaucoup plus grand que sans boosting, et ce sous la même hypothèse de complexité. En fait, plus on s'autorise une hypothèse de complexité forte (*i.e.* moins elle est "réaliste"), et plus ce ratio explose. Par exemple, on peut remplacer la classe ZPP par la classe équivalente dont le temps est borné par une quantité en $\mathcal{O}(2^{|I|^{\Omega(1)}})$ (où I dénote l'instance du problème). L'hypothèse de complexité devient alors $NP \not\subset ZPTIME\left[2^{|I|^{\Omega(1)}}\right]$. Notons qu'il s'agit de l'hypothèse la plus forte qu'on puisse faire avant de tomber dans le trivialement faux (tous les problèmes de NP sont solvables par des algorithmes déterministes de temps exponentiel), et que cette hypothèse a déjà été utilisée, même sous une forme encore plus forte (car introduisant d'autres formes de randomisation), par exemple dans [Pillaipakkamnatt and Raghavan, 1994]. Cette hypothèse nous permet par exemple de remplacer la borne de droite en eq. 3 par celle-ci, beaucoup plus forte :

$$|h| \quad \leq \quad |c| \times 2^{n^{\Omega(1)}} \quad . \tag{4}$$

Sous réserve de ce que cache exactement le "Ω" (chose extrèment difficile à savoir : la réduction globale regroupe en réalité quatre réductions [Nock et al., 1998]), on peut conclure sinon à la non-PAC apprenabilité de DNF, tout du moins à la difficulté algorithmique énorme de la tâche. L'article [Nock et al., 1998] figure en annexe (page 47).

2.2 Résultats divers sur les clauses de Horn

Apprentissage robuste, et en dégradation limitée

La technique de réduction que j'avais utilisé pour DNF, consistant à "Booster" des instances de problèmes difficiles, s'est aussi révélée utile dans d'autres problèmes.
L'ILP, ou programmation logique inductive (*I*nductive *L*ogic *P*rogramming), est une branche très active de la recherche en informatique, au confluent de l'apprentissage automatique et de la logique. Son objectif est l'apprentissage de concepts exprimables sous la forme d'ensembles de clauses de Horn plus ou moins restreintes. Tous les résultats que j'ai obtenu ont porté sur le cas "simple" ou il y a deux classes ($l = 2$).
Après quelques études d'apprenabilité sur les grandes classes de clauses de Horn, en

[5]William of Occam, 1285-1349.

montrant la difficulté de l'apprentissage [Kietz and Dzeroski, 1994], quelques auteurs ont réussi à démontrer l'apprenabilité PAC de sous-classes relativement contraintes [Dzeroski et al., 1992]. Ces résultats ne nous satisfaisaient cependant pas. En effet, les classes pour lesquelles ces résultats positifs étaient prouvés étaient si contraintes qu'on avait du mal à imaginer comment le concept cible pouvait être "forcé" à faire partie de telles classes en pratique, hypothèse pourtant fondamentale du modèle PAC. A la limite, on pouvait même se demander quelle légitimité pouvaient avoir certaines de ces classes, tant il était peu probable qu'un domaine même modestement complexe y soit jamais codable.
En d'autres termes, il nous semblait beaucoup plus judicieux de déplacer l'étude de ces classes trop restreintes, non dans le cadre PAC où c et h sont dans $\mathcal{C}$, mais plutôt dans un cadre d'apprentissage robuste [Höffgen and Simon, 1992]. Dans un tel cadre, seule h fait partie de $\mathcal{C}$, et l'on remplace la contrainte statistique de PAC par celle, beaucoup plus réaliste, que

$$\mathbf{Pr}\left[\mathbf{Pr}_{\mathcal{D}}[h \neq c] > \min_{h^* \in \mathcal{C}} \mathbf{Pr}_{\mathcal{D}}[h^* \neq c] + \epsilon\right] \quad < \quad \delta \ . \tag{5}$$

Cette fois, le modèle permet de prendre en compte les cas ou la règle de Bayes n'est d'erreur nulle, et les cas ou les données sont bruitées. Bien évidemment, force fut de constater que les résultats étaient nettement moins positifs, voire plus positifs du tout. Par exemple, nous avons prouvé dans [Jappy et al., 1996] que même l'ensemble des clauses de Horn étant l'intersection de la plupart des ensembles connus pour être alors PAC-apprenables (donc, une classe extrêmement simple), ne permettait pas cet apprentissage robuste. Ceci était du reste valable même en limitant le nombre de clauses de Horn à une seule !

Ce premier résultat n'utilisait pas la technique de boosting de résultats négatifs. En utilisant cette technique, nous avons pu obtenir des résultats beaucoup plus forts, en exhibant par exemple une corrélation entre le nombre de clauses de Horn et la difficulté d'approximer, même de très loin, l'ineq. 5. Supposons que nous disposions de n prédicats. n joue dans le cas de l'ILP un peu le même rôle que notre nombre de variables de description des exemples dans le cadre PAC au début de ce chapitre, avec la différence notable que n peut être extrêmement grand dans le cas de l'ILP. Dans le résultat que nous avons obtenu [Nock and Jappy, 1998a], nous avons quelque peu déplacé la problématique de l'apprentissage vers celle de l'approximation d'un problème d'optimisation ordinaire, ainsi présenté :

> **Instance** : deux échantillons d'exemples positifs (S^+) et négatifs (S^-), un poids $w(.) \in I\!N$ pour chaque exemple.
> **Solutions faisables** : $h \in \mathcal{C}$, où $\mathcal{C}$ rassemble les ensembles de $g(.)$ clauses de Horn sans symboles de fonction[6].
> **Coût** : $\sum_{(x \in S^+ \wedge h(x)=0) \vee (x \in S^- \wedge h(x)=1)} w(x)$.

Il s'agissait *a priori* de montrer qu'aucun algorithme polynômial ne peut renvoyer une

[6] $g(.)$ peut être une constante, une fonction de divers paramètres, ...

hypothèse h de l'ensemble des solutions faisables, dont le coût satisfasse :

$$\sum_{\substack{(x \in S^+ \wedge h(x)=0) \\ (x \in S^- \wedge h(x)=1)}} w(x) \quad \leq \quad K \times \min_{h^* \in \mathcal{C}} \sum_{\substack{(x \in S^+ \wedge h^*(x)=0) \\ (x \in S^- \wedge h^*(x)=1)}} w(x) \ . \qquad (6)$$

Un coup d'oeil permet de repérer un lien de parenté entre l'ineq. 6 et l'ineq. 5. En fait, il est bien connu depuis [Höffgen and Simon, 1992] au moins que des résultats négatifs d'approximabilité sur ce problème peuvent se transmettre à l'apprentissage moyennant une randomisation bien placée sur les classes de complexité utilisées. Par exemple, démontrer un résultat négatif d'inapproximabilité de ratio K sur l'ineq. 6, c'est aussi démontrer la non-apprenabilité de $\mathcal{C}$ dans un modèle d'apprenabilité où l'on relâche la contrainte de l'ineq. 5 par celle-ci :

$$\mathbf{Pr}\left[\mathbf{Pr}_{\mathcal{D}}[h \neq c] > K \times \min_{h^* \in \mathcal{C}} \mathbf{Pr}_{\mathcal{D}}[h^* \neq c] + \epsilon\right] \ < \ \delta \ . \qquad (7)$$

Si K est une constante, on obtient le modèle d'apprentissage *en dégradation limitée* [Höffgen and Simon, 1992]. Si le résultat est négatif (pour toute constante), alors on obtient un résultat de *dégradation illimitée*. Nous avons obtenu plus qu'un résultat de dégradation illimitée [Nock and Jappy, 1998a], puisque le ratio d'inapproximabilité obtenu explose avec le nombre de clauses de Horn autorisées (la valeur de $g(.)$), et peut parfois prendre des valeurs non constantes :

1. si $g(.)$ est une constante, alors on peut choisir n'importe-quelle constante pour K dans l'ineq. 6 (donc aussi 7),
2. si $g(.)$ devient une quantité polylogarithmique en n (en $\mathcal{O}(\log^\alpha n)$), alors K devient n'importe-quelle fonction quasi-polynômiale de n (en $\mathcal{O}(n^{\log^\alpha n})$) dans l'ineq. 6 (donc, également, 7).

Il faut noter que le premier résultat s'obtient sous l'hypothèse de complexité $NP \not\subset P$, alors que le second nécessite l'hypothèse un peu plus forte $NP \not\subset QP$ (QP est l'équivalent de P, où l'on remplace l'exigence de temps polynômial par une exigence quasi-polynômiale). Par contre, le second résultat s'applique aussi aux algorithmes quasi-polynômiaux. La contrepartie à payer reste donc modeste, pour une hypothèse de complexité qu'on retrouve en outre largement dans la littérature. L'article [Nock and Jappy, 1998a] figure en annexe de ce document (page 52).

Remplacement du taux d'erreur

Les clauses de Horn ont ceci d'intéressant qu'elles sont relativement faciles à manipuler, en particulier pour explorer des problèmes originaux. Un de ces problèmes est soulevé depuis quelques années dans des articles touchant essentiellement au Data Mining. Un certain nombre d'auteurs (parmi lesquels [Kubat et al., 1998, Provost et al., 1998, Provost and Fawcett, 1998, Provost and Fawcett, 1997]) ont critiqué l'utilisation de la fréquence (ou la probabilité) d'erreur pour juger des capacités d'un algorithme. D'un point de vue Data Mining, une telle utilisation systématique est en effet discutable : juger un système seulement sur ce critère, c'est faire par exemple abstraction des problèmes dans lesquels une ou plusieurs classes se retrouvent en proportion ridicules, si bien que prédire la classe majoritaire dans tous les cas limite déjà le taux d'erreur à quelques % seulement, sans qu'on modélise quoi que ce soit du problème. Par exemple, [Kubat et al., 1998] présentent un problème de détection de nappes de pétrole, dans lequel l'objectif est de distinguer sur des images radar les nappes de pétroles potentielles des artéfacts d'imagerie. Sur de telles images, la règle majorité, qui prédit systématiquement un artéfact (un comble !), obtient un taux d'erreur plus que confortable, de 4% environ. Est-il besoin de préciser que n'importe-quel utilisateur préférerait sans aucun doute un système qui ferait 6% d'erreurs, mais sur *chaque* classe, ce qui reviendrait tout de même à trouver 94% des nappes de pétrole ! D'une manière plus générale, nombreux sont les articles en apprentissage automatique qui ne jugent pas les algorithmes uniquement sur leur taux d'erreur, mais aussi sur d'autres critères [Nock and Gascuel, 1995, Nock and Jappy, 1998b].

Un certain nombre de critères ont été proposé pour remplacer le taux d'erreur dans l'évaluation des classifieurs. On peut distinguer grossièrement trois catégories de critères pour remplacer ainsi le taux d'erreur : ceux qui complètent le taux d'erreur par une autre contrainte (taille de la formule [Nock and Gascuel, 1995, Nock and Jappy, 1998a], erreur sur une classe en particuliers, etc.), ceux qui remplacent l'erreur par un autre critère [Kubat et al., 1998], et ceux qui ont (re)popularisé la démarche ROC (*Receiver Operating Characteristic*) qui remplace le taux d'erreur par un point ou une courbe sur le pavé (Taux de Faux Positifs)$\times$(Taux de Vrais Positifs) (voir par exemple [Egan, 1975, Provost and Fawcett, 1998, Provost and Fawcett, 1997]).

Je me suis intéressé à la complexité du remplacement du taux d'erreur, en particulier aux cas ou un tel remplacement induit un coût algorithmique non négligeable par rapport à l'utilisation du taux d'erreur seul. En prenant le cas des clauses de Horn très simples comme classe $\mathcal{C}$, les résultats ont été surprenants [Nock, 1999]. En fait, il s'agirait d'avantage d'un fardeau algorithmique, bien plus que d'un coût ordinaire, puisque les propriétés suivantes sont vérifiées :

- tous ces critères sont équivalents au regard de la réduction,
- le problème devient subitement extrêmement difficile, alors qu'il est trivial à résoudre avec l'erreur seule,
- même si on autorise la multiplication arbitraire des clauses de Horn, le résultat reste difficile. Ici, dans une certaine mesure, arbitraire peut même vouloir dire exponentiel,
- la difficulté apparaît soudainement avec le nouveau critère : elle n'est pas fonction, en particulier, de la "force" de la contrainte.

Le troisième point est, à mon sens, important, car il s'agit à ma connaissance d'une des premières preuves de complexité en théorie de l'apprentissage qui, non seulement ne place aucune borne sur le nombre de clauses de Horn, mais en plus peut considérer que le temps d'écriture d'un ensemble quelconque de clauses n'est pas plus coûteux que pour une seule. En un certain sens en effet, on peut donc supposer qu'on autorise un nombre exponentiel de clauses. A la limite, une procédure qui écrirait *toutes* les clauses de Horn entre aussi dans cette catégorie ! Dit autrement, construire une seule clause de Horn ne serait pas plus coûteux que de construire un ensemble quelconque... L'article [Nock, 1999] figure en annexe (page 55).

Bien entendu, il ne s'agit pas du tout de dire que celui qui cherche sa nappe de pétrole doive troquer son informaticien et son ordinateur contre un médium et une carte IGN. Bien au contraire : il s'agit uniquement de montrer que certains problèmes deviennent beaucoup plus difficile à résoudre lorsqu'on complexifie ainsi le critère de décision (ce qui est somme toutes intuitif !), ou bien que certains formalismes de représentation de concepts se prêtent certainement moins bien que d'autres à ce genre d'exercice. Je ne m'attendais tout de même pas à un tel effet de seuil dans la difficulté de ce problème (on passe en effet du trivial au "presque impossible").

2.3 Résultats en réduction de données

Je me suis aussi intéressé à une problématique fortement combinatoire, celle de la réduction de données, pour laquelle à ma connaissance seuls deux articles vraiment consacré ont été publiés (la même année) [Charikar et al., 2000, Nock and Sebban, 2000b]. D'autres, comme [Blum and Langley, 1997] sont en réalité beaucoup plus généraux et synthétiques. D'un point de vue combinatoire, la réduction de données s'intéresse essentiellement en la réduction du nombre de variables de description d'un domaine (*i.e.* le nombre de colonnes), la réduction du nombre d'exemples (*i.e.* le nombre de lignes, on parle alors de sélection de prototypes), ou bien des deux à la fois.

Avec Marc Sebban, nous avions constaté que le problème de la réduction de données était sans doute beaucoup plus difficile que certains résultats négatifs ne le laissaient penser : par exemple, la réduction de [Blum and Langley, 1997], qui consistait d'avantage en une réécriture du problème du "Set Cover" [Garey and Johnson, 1979] qu'en une réduction *sensus stricto*, était, nous semblait-il, bien loin de capturer toute le difficulté du problème de la sélection de variables. Ce fut là encore l'occasion d'utiliser cette technique de "Boosting" de résultats négatifs.

Le domaine de la sélection d'exemples ou d'attributs n'est certainement pas nouveau (en particulier en Statistiques), mais il a connu récemment un regain d'intérêt, avec l'augmentation dramatique de la taille des jeux et bases de données (à titre d'exemple, l'article [John et al., 1994] fait partie des 10 articles les plus couramment cités en Intelligence Artificielle[7]). Un prétraitement rapide qui enlèverait l'information redondante ou bruitée réduirait grandement la complexité des étapes suivantes du Data Mining, souvent très coûteuses (par exemple, les algorithmes d'induction) ; il faciliterait aussi la visualisation des données, et pourrait même améliorer la performance des classifieurs. Cette dernière remarque est contre-intuitive, car un classifieur qui réaliserait l'optimum de Bayes serait à coup sûr monotone, *i.e.* sa performance ne pourrait se dégrader en ajoutant de l'in-

[7] `http ://citeseer.nj.nec.com`, `http ://citeseer.nj.nec.com/ArtificialIntelligence/`.

formation (des variables ou des exemples). En réalité, la règle de Bayes est un optimum "idéal", et difficile à approcher, et les algorithmes utilisés en pratique ne sont pas monotones [Kohavi, 1994, Sebban and Nock, 2002]. Dans cette situation, l'ajout de quantités substantielles de bruit peut réduire l'efficacité du système, voire carrément en supprimer tous les avantages. Le Boosting [Freund and Schapire, 1997] est un exemple de méthode de classification très puissante en temps normal, mais pour laquelle la présence de bruit dans les données peut affecter dramatiquement les performances [Bauer and Kohavi, 1999].

Avec Marc, nous avons donc repris les définitions formelles de l'utilité (*relevance* en anglais) qu'on a pu trouver dans la littérature, pour une variable ou un exemple, et nous les avons clusterisées en deux catégories. La tâche de formalisation n'est pas forcément aisée, comme le souligne [Blum and Langley, 1997], car cette utilité doit être définie par rapport à quelque chose, mais cet objet ne peut pas toujours être facilement isolé.

Les deux catégories que nous avons remarqué définissent l'utilité soit par rapport à une distribution, soit par rapport à un concept. Dans le premier cas, il s'agit de supprimer des variables ou des exemples en s'assurant la conservation d'une mesure d'information. Dans le deuxième cas, il s'agit de s'assurer par exemple que les différents concepts d'erreur nulle exprimables sur le sous-ensemble de données n'étaient pas trop différents de ceux qu'on pourrait construire sur toutes les données. Les problèmes d'approximation que nous avons étudié se calquent sur le modèle de problème suivant :

- **Instance** : échantillon d'exemples positifs (S^+) et négatifs (S^-), décrits sur un ensemble de n variables.
- **Solutions faisables** : $S'^+ \subseteq S^+$ et $S'^- \subseteq S^-$, où l'on a enlevé des variables (pour la sélection de variables) ou des exemples (pour la sélection de prototypes), et vérifiant une propriété d'utilité **P**.
- **Coût** : $|S'^+| + |S'^-|$.

Afin de clarifier cette notion de propriété d'utilité, voici à titre d'exemple deux définitions que nous avons étudié, tirées de [John et al., 1994]. Appelons $V = \{v_1, v_2, ..., v_n\}$ l'ensemble des n variables, et $V_i = V \backslash \{v_i\}$ $(i = 1, 2, ..., n)$. On note $v_{\backslash i}$ une assignation de toutes les variables de V_i.

Déefinition 1 *Une variable v_i est dite **fortement** utile ssi il existe v, y et $v_{\backslash i}$ pour lesquels $\Pr(v_i = v, V_i = v_{\backslash i}) > 0$, et tels que :*

$$\Pr(Y = y | v_i = v, V_i = v_{\backslash i}) \neq \Pr(Y = y | V_i = v_{\backslash i}) \ . \tag{8}$$

Déefinition 2 *Une variable v_i est dite **faiblement** utile ssi elle n'est pas fortement utile, et il existe un sous-ensemble de variables V'_i de V_i pour lequel il existe v, y et $v_{\backslash i}$ tels que $\Pr(v_i = v, V_i = v_{\backslash i}) > 0$, avec :*

$$\Pr(Y = y | v_i = v, V'_i = v'_{\backslash i}) \neq \Pr(Y = y | V_i = v'_{\backslash i}) \ . \tag{9}$$

En d'autres termes, une variable est faiblement utile ssi elle devient fortement utile après avoir supprimé un sous-ensemble de variables. Notre article [Nock and Sebban, 2000b]

	Distribution	Concept
Variables	$n^{\Omega(1)}$(*, voir def. 2)	$n^{\Omega(1)}$ (*)
Prototypes	$n^{\Omega(1)}$	$2^{n^{\Omega(1)}}$ (*)

FIG. 1 – Résumé des résultats les plus forts obtenus dans [Nock and Sebban, 2000b]. Les ratios d'inapproximabilité suivis de "(*)" sont presque optimaux.

rassemble une douzaine de théorèmes sur la difficulté de résoudre ou d'approximer le problème de la sélection de variables ou de prototypes. Seule la définition 1 permet de trouver un algorithme polynômial (simple) de minimisation. On peut d'ailleurs faire une comparaison avec le résultat obtenu sur le remplacement du taux d'erreur dans mon étude sur les clauses de Horn [Nock, 1999] : toutes les autres définitions (dont la définition 2 plus haut) ont abouti à des résultats négatifs extrêmement forts. La figure 1 résume les différents ratios d'inapproximabilité qui ont été obtenu selon la nature des données qu'on supprime (exemples ou variables), et selon la propriété d'utilité **P** (par rapport à une distribution ou un concept). Ce qui est remarquable c'est que mis à part la case en bas à gauche (sélection de prototypes avec mesure d'utilité par rapport à une distribution), tous les résultats ne sont pas loin d'être optimaux sous réserve de ce que cache la notation "Ω". En effet, pour la réduction du nombre de variables, le ratio trivial d'approximabilité est en fait n (on garde tout !), pour les exemples décrits avec des variables binaires, il vaut par contre 2^n (on garde encore tout !). L'autre fait remarquable est la différence d'approximabilité des deux notions d'utilité de [John et al., 1994] en définition 1 et 2. Un gouffre de complexité sépare ces deux notions pourtant voisines ! L'article [Nock and Sebban, 2000b] figure en annexe (page 84).

3 ... et quelques résultats positifs

Je me suis intéressé aussi à l'obtention de résultats positifs d'apprentissage dans le modèle PAC, mais pour des formalismes plus puissants que ceux utilisés couramment en apprentissage. Avec Pascal Jappy et Babak Esfandiari, je me suis respectivement intéressé à l'apprenabilité de sous-classes des Graphes Conceptuels [Jappy and Nock, 1998], et de formalismes dérivés des langages de description des chroniques pour l'aide à la supervision de réseaux [Quinqueton et al., 1997, Nock and Esfandiari, 1998]. Tous ces résultats utilisent fondamentalement l'astuce de Valiant pour apprendre k-DNF : on montre que les formules qu'on apprend sont en réalité la composition de nouveaux descripteurs polynômialement énumérables avec une classe qui est déjà PAC-apprenable. L'article [Nock and Esfandiari, 1998] intègre aussi quelques résultats simples dans les cas ou on autorise la distribution sur les exemples à varier dans certaines limites au cours du temps. A ma connaissance, l'article [Jappy and Nock, 1998] fût le premier à prouver des résultats (positifs) d'apprenabilité sur certaines sous-classes de formalisme des graphes conceptuels.

4 Conclusion

Une partie de ces résultats a fait l'objet d'un exposé dans le cadre de la remise du prix Simon Régnier de la Société Francophone de Classification [Nock, 2001a].

Deuxième partie

De la réduction de données à l'induction et l'interprétation de concepts

5 Brève introduction

Même si elle n'est pas la plus longue dans ce manuscrit, il s'agit certainement de la partie la plus fournie de cette Habilitation. Depuis le début de ma thèse, il s'agit en effet de mon sujet principal de préoccupation, consistant à construire des algorithmes d'induction aussi fiables que possible, et sur lesquels on place des contraintes diverses sur la sortie, en particulier sur sa taille [Nock, 1998].
L'induction est un concept simple dans son principe : il s'agit, à partir d'un échantillon d'exemples observés, de construire la représentation d'un concept qui nous permette à la fois de classer des observations inconnues de la manière la plus fiable possible, mais aussi d'interpréter cette représentation pour en tirer des conséquences, des enseignements sur le domaine en question. C'est en quelque sorte un modèle "PAC-pratique", avec des contraintes similaires (efficacité algorithmique, généralisation), et de nouvelles (taille de la sortie, interprétabilité, etc.). Évidemment, malgré toute sa beauté, on trouve dans ces applications pratiques toute la limitation du modèle PAC théorique, qui donne essentiellement tous ses résultats positifs sous contrainte de tirage d'échantillons d'exemples de taille suffisante. En pratique, peu nombreux sont les domaines où l'on disposerait d'un oracle permettant d'échantillonner autant d'exemples que nécessaire pour satisfaire aux exigences de PAC. On dispose plutôt de jeux de données de taille fixée, sur lesquels on ne peut souvent qu'espérer de bonnes qualités en généralisation, sans pouvoir les certifier, même au seuil d'un risque statistique (le paramètre δ du modèle PAC).

Une partie de mon travail a consisté non seulement à élaborer de nouveaux algorithmes pour des classes de représentation de concepts connues, mais aussi pour de nouvelles, afin de permettre une interprétation et une fouille des données aussi précise que possible, les modèles (formules) préférables dépendant du domaine mais aussi de l'utilisateur. Cette dernière problématique est typiquement celle du Data Mining. Au cours du temps, ma recherche s'est quelque peu déplacée en amont du problème, avec les travaux que j'ai effectué avec Marc Sebban, travaux qui consistaient essentiellement à remarquer qu'on pouvait sans doute "aider" les algorithmes d'induction sur un certain nombre de points (complexité, fiabilité, etc.) simplement en faisant un pré-traitement des données consistant à en enlever l'information qui pouvait être en quelque sorte "néfaste" pour ces algorithmes.

Globalement, cette recherche a donc couvert une partie largement significative de l'induction, depuis le pré-traitement des données jusqu'aux choix des représentations de concepts utilisées. C'est aussi celle qui a aboutit au plus grand nombre de publications, avec par exemple quatre articles consacrés publiés à l'International Conference on Machine Learning [Nock and Gascuel, 1995, Nock and Jappy, 1998b, Sebban and Nock, 2000c, Sebban et al., 2001b] (un cinquième est aussi sorti dans l'ICML, mais il concernait des résultats théoriques [Jappy et al., 1996]).

Afin d'en dégager mes diverses contributions à ce vaste domaine, j'ai choisi de présenter ce chapitre en suivant les grandes étapes de l'induction, depuis le pré-traitement des données jusque vers les choix de sortie d'algorithmes d'induction, et leur interprétation.

6 Sélection de variables et de prototypes

Comme je l'ai déjà souligné auparavant (page 11), dans un "monde algorithmiquement parfait", la sélection de variables ou de prototypes (des exemples typiques, représentatifs) n'aurait pas droit de cité, sauf à prendre en compte des contraintes algorithmiques, ou de stockage. C'est parce que les algorithmes d'induction utilisés en pratique ne sont pas monotones qu'il peut être extrêmement utile d'effectuer une étape de filtrage des données. Un des grands écueils de l'apprentissage automatique est d'ailleurs la construction de modèles si complexes sur les données observées qu'il finit par capturer une partie de plus en plus ténue du modèle général, en étant donc de moins en moins fiable. C'est le problème de l'*overfitting*. A l'origine, ce problème était essentiellement abordé par la recherche directe de modèles suffisamment simples, au nom de grands principes, tels celui du rasoir d'Occam [Blumer et al., 1987], ou bien du principe de description minimale [Rissanen, 1986]. Le problème est que cet objectif de simplicité est largement infaisable d'un point de vue algorithmique, pour un grand nombre de classes de représentation de concepts [Blum and Rivest, 1988, Hyafil and Rivest, 1976, Nock and Gascuel, 1995, Nock and Jappy, 1998b] (et beaucoup d'autres). [John et al., 1994] montrent d'ailleurs à quel point la structure d'un arbre construit par des algorithmes tels ID3 [Quinlan, 1986], CART [Breiman et al., 1984] ou bien C4.5 [Quinlan, 1994] peut être altérée profondément par l'ajout de variables ainsi *inutiles*[8]. [John et al., 1994] montrent aussi que faire une sélection de variables peut avoir un impact très positif sur la taille de la structure construite (parfois, elle peut être plus petite de plusieurs ordres de grandeur), sans parfois augmenter le taux d'erreur (voire même en le réduisant significativement dans les cas pathologiques). Nous avons fait des observations similaires dans [Sebban and Nock, 2000b, Sebban et al., 2001a]. Dans le cas des algorithmes de classification basés sur les plus proches voisins, la situation est aussi critique : [Aha, 1992] remarque que le stockage de l'algorithme IB3 augmente exponentiellement avec le nombre de variables *inutiles*.

On pourrait résumer l'intérêt de la réduction de données en trois points :

1. réduction de la complexité des étapes ultérieures,
2. amélioration de la fiabilité des classifieurs,
3. amélioration de la visualisation, et de la compréhensibilité des classifieurs.

Évidemment, le premier argument n'a de justification que si l'algorithme de réduction de données a déjà une complexité limitée, en tout cas beaucoup plus limitée que celle des algorithmes qu'on peut utiliser par la suite. Ceci étant, il est remarqué par [John et al., 1994] que les algorithmes ayant une phase d'élagage, comme C4.5 [Quinlan, 1994], sont parfois tellement coûteux sur la phase d'élagage, que même un algorithme de réduction de données utilisant de manière répétitive la phase de construction de formules (de C4.5 !) peut

[8] Il s'agit de la traduction littérale de l'anglais, *irrelevant*. J'ai choisi de garder ce terme, mais un terme sans doute de circonstance serait *dangereuses*, tant le fait de conserver ces variables peut endommager la classification.

avoir un coût très raisonnable par rapport à une seule exécution de l'algorithme complet, sachant qu'en plus il réduit la phase d'élagage de l'algorithme d'induction (car il réduit la taille du jeux de données). Nous avons fait des observations similaires dans [Sebban et al., 2001a].

Avec Marc Sebban, nous avons voulu aborder le problème de la réduction de données à la fois sous l'angle de la sélection de variables, mais aussi de la sélection de prototypes. Ces deux approches sont en effet largement complémentaires : enlever une variable, c'est enlever toute une colonne de données (m si on dispose de m exemples) ; enlever un exemple, c'est enlever une ligne de données ($n + 1$ si on dispose de n variables) : comme on a en général $m \gg n$, la sélection de prototypes est chirurgicale là où la sélection de variables est beaucoup plus brutale, mais la complexité requise pour enlever une quantité de données fixée est beaucoup plus faible avec cette dernière.

Ce fut aussi un de nos objectifs (et article, [Sebban and Nock, 2000a]) que de combiner les deux mécanismes de la réduction de données, dans la première approche (à ma connaissance) intégrant un mécanisme *identique* de sélection de variables et de prototypes dans un seul algorithme. Les trois sous-parties qui suivent présentent d'ailleurs respectivement la sélection de variables, la sélection de prototype, et la sélection combinée variables/prototypes.

6.1 Sélection de variables

En accord avec la terminologie de [John et al., 1994], il existerait deux grands types d'algorithmes de sélection de variables : les approches *filter*, et les approches *wrapper*. Bien que la traduction littérale Française de ces termes soit raisonnable (respectivement *filtre* et *enveloppeurs*), je garderait ici volontairement les termes anglais. Les premières approches, *filter*, sélectionnent des variables indépendamment des algorithmes d'induction qui leur feront suite. En un certain sens, elles "filtrent" les variables, selon des mesures qui peuvent varier : choix d'un algorithme d'induction (indépendamment de ceux qu'on peut utiliser) [Cardie, 1993], mesure d'utilité [John et al., 1994], etc. Cette approche a le désavantage de ne pas prendre en considération l'algorithme qu'on pourra utiliser par la suite, mais, sous réserve que la sélection est faite avec un bon critère, elle est certainement la bonne pour attaquer les notions d'utilité traditionnelles, qui ne reposent sur aucune dépendance algorithmique [John et al., 1994, Nock and Sebban, 2000b]. L'autre approche de [John et al., 1994], appelée *wrapper*, tient compte cette fois de l'algorithme d'induction subséquent, puisque l'algorithme de sélection de variables *enveloppe* l'algorithme d'induction, dont il se sert pour sélectionner les variables. La sélection est donc beaucoup plus adaptée à *cet* algorithme d'induction, mais elle repose sur ses biais, et est sans doute moins performant pour traiter les définitions formelles de l'utilité.

[John et al., 1994] choisissent cette dernière approche pour faire de la sélection de variables, en évaluant la qualité des sous-ensembles de variables à l'aide de C4.5, et d'une procédure de validation croisée. L'enveloppeur est un algorithme glouton qui procède en pas-à-pas, en ajoutant à chaque étape la variable la meilleure à un ensemble initialisé à $\emptyset$. Il s'agit d'un approche *forward stepping*, mais son équivalent *backward stepping* est aussi possible : elle consiste simplement à enlever la moins bonne des variables à chaque pas, en commençant avec l'ensemble de toutes les variables. Il y a aussi des approches intermédiaires [John et al., 1994].

Les approches du problème de la sélection de variables que nous avons eu avec Marc Sebban appartiennent aux deux catégories *filter* et *wrapper* (il y en a même une hybride). Sans utiliser la classification de [John et al., 1994] sur les types d'approches, on pourrait séparer nos approches en deux types : celles exploitant des résultats de théorie de l'information (plutôt *filter*), et celles issues d'algorithmes de Boosting (plutôt *wrapper*).

Gains d'information et sélection de variables : des approches *filter* et hybrides

Dans [Sebban and Nock, 2002], nous présentons un algorithme de sélection de variables qui repose à la base sur une mesure d'information tabulée, calculée sur un arbre recouvrant de poids minimal (MST). Il s'agit donc typiquement d'une approche *filter*. Sans exposer totalement l'algorithme (l'article [Sebban and Nock, 2002] figure en annexe, page 99), en voici les grands principes :

- On construit sur tout l'échantillon d'exemples une structure géométrique rassemblant tous les exemples (au départ, un MST)
- Sur cette structure, on calcule une quantité appelée "incertitude totale" U. Cette quantité est une somme pondérée d'entropies quadratiques, chacune étant mesurée localement au niveau de chaque exemple.
- L'entropie quadratique calculée au niveau d'un exemple traduit la "pureté" du découpage en classes au niveau sous-MST induit dont les arêtes portent l'exemple. Plus l'entropie est faible, et moins il existe de classes dans ce voisinage (également, plus une ou plusieurs classes deviennent largement majoritaires).
- A chaque fois qu'on enlève une variable, on recalcule le MST, et donc l'incertitude totale U (chaque variable enlevée cause potentiellement une modification du MST). L'objectif est donc de "mesurer" ce qu'apporte chaque MST au sens de cette incertitude.
- On calcule un critère,

$$RCG = \frac{U_0 - U}{U_0} \tag{10}$$

  pour chaque MST possible. U_0 est l'incertitude qu'on aurait si on enlevait toutes les variables de description (de manière équivalente, il s'agit de l'incertitude du découpage en classe dans tous l'échantillon d'exemples). Ce critère, pour chaque MST, ressemble non pas à une incertitude, mais plus à un *gain de certitude relatif* (RCG, R*elative* C*ertainty* G*ain*), qui compare le gain obtenu par l'information géométrique du MST, par rapport à l'absence de structure de voisinage.
- RCG est tabulable dès que la taille de l'échantillon d'exemple excède des seuils raisonnables ($m > 30$), sous une hypothèse H_0 que la structure géométrique du MST n'apporte pas d' "information" supplémentaire par rapport à l'absence de structure

géométrique. On peut donc calculer un risque "seuil" α_v pour chaque variable v, au-delà duquel on rejette H_0.
- L'algorithme est alors simple dans son principe : c'est une procédure *forward stepping* qui sélectionne à l'instant courant la variable pour laquelle le risque α_v est le plus petit, et telle qu'on rejette encore l'hypothèse d'identité entre le découpage local des distributions et celui obtenu sans structure géométrique (H_0).

Nous avons également testé le remplacement de la structure du MST par celle, plus courante en apprentissage, du voisinage d'un algorithme de plus proche voisin (PPV). Cette seconde modification prend donc en compte une structure de voisinage qui est celle d'un classifieur (en plus, nous avons utilisé ce classifieur par la suite), mais avec un critère de qualité qui n'est pas celui des PPV traditionnels (le taux d'erreur). Il s'agit là d'une approche que nous avons qualifié d'*hybride* dans [Sebban and Nock, 2002], tant elle se situe entre les approches *wrapper* et *filter* traditionnelles.

Le Boosting au service de la sélection de variables : une approche *wrapper*

Le Boosting est une méthode de classification qui, informellement, construit (dans l'acception de [Freund and Schapire, 1997]) un séparateur linéaire (on prend le cas simple ou le nombre de classes $l = 2$) avec de multiples classifieurs dits "de base", ou encore "faibles". Il s'agit d'une méthode extrêmement puissante, cataloguée par L. Breiman comme la méthode de classification sur étagère la plus puissante du monde [Friedman et al., 2000]. L'adjectif "faible" pour chaque classifieur n'est d'ailleurs pas utilisé au hasard : en effet, il est possible de montrer que même si chaque classifieur est à peine plus efficace qu'un jet de pièce non-biaisée (*i.e.* le hasard !), alors on est capable de construire un séparateur linéaire dont l'erreur sera *arbitrairement* petite sur l'échantillon d'apprentissage (on a même des résultats sur tout $\mathcal{X}$, [Schapire et al., 1998]). Évidemment, il y a un truc, qui tient dans l'hypothèse que chaque classifieur de base est ainsi "faible", mais sur *toute* distribution sur les exemples. L'astuce est alors de construire des classifieurs sur le même échantillon d'exemples, *à la différence prêt* qu'on modifie la distribution des exemples pour que ceux qui ont eu tendance à être mal classés par les classifieurs précédents aient un poids plus important. En étant légèrement scabreux dans notre raisonnement, on pourrait dire que même un classifieur ainsi faible sera bien obligé de bien classer un jour ces exemples !

Dans [Sebban and Nock, 1999], nous avons testé l'apport potentiel du Boosting aux approches de type *wrapper*, dans l'esprit de [John et al., 1994]. L'approche s'est révélée concluante, mais l'algorithme n'était pas suffisamment rapide. Non content, en effet, de répéter un certain nombre de fois le même algorithme d'induction, comme dans les approches *wrapper*, nous effectuions à chaque étape un algorithme de Boosting, qui *lui aussi* répète un certain nombre de fois un appel à une procédure construisant des classifieurs ! Notons tout de même que notre approche n'était tout de même pas si gourmande, puisque les classifieurs de base étaient basés sur les plus proches voisins, donc ayant une partie induction réduite à sa partie ténue ! Mais tout de même, ça n'était pas totalement satisfaisant.

Nous avons donc recherché un moyen éventuel permettant, non d'améliorer les résultats du Boosting, mais la convergence de l'algorithme vers les bonnes valeurs, sachant que nous nous étions placés dans un cas très particulier de domaine $\mathcal{X}$: les exemples n'ayant que très peu de variables de description au départ (l'algorithme était une approche *forward stepping*), beaucoup d'exemples se partageaient la même description. En d'autres termes, le problème disposait dès le départ d'une règle de Bayes dont la probabilité d'erreur était (largement) positive, cadre dans lequel le modèle PAC de Valiant [Valiant, 1984] n'est plus tout-à-fait à l'aise, tout du moins dans sa version initiale. Sachant que le Boosting avait été précisément formulé pour répondre au besoin d'obtention d'algorithmes forts dans le modèle PAC[9], il ne se trouvait de fait pas tellement "à l'aise" dans un tel cadre (on trouvera des critiques à ce sujet dans les dernières pages de [Breiman, 1996]). Il n'y a qu'à regarder, par exemple, le théorème 1 de [Schapire and Singer, 1998] (voir plus bas), pour s'en rendre compte.

Nous avons donc, avec Marc Sebban, essayé d'accélérer l'algorithme de Boosting le plus célèbre, **AdaBoost**, dans ces cas-là où l'optimum de Bayes s'approche des "sommets". Nous avons obtenu, entr'autre, une règle de mise à jour des poids des exemples qui généralise celle de [Schapire and Singer, 1998], et qui a une particularité : on peut en attendre une convergence plus rapide de l'algorithme précisément dans nos cas de figure ou les exemples partagent des descriptions identiques mais des classes différentes [Nock and Sebban, 2001b]. Je vais en présenter le résultat principal, une généralisation du théorème 1 de [Schapire and Singer, 1998].

La figure 1 présente **AdaBoost**. Les notations utilisées dans l'algorithme sont les suivantes :

- la distribution générée à l'instant $t = 1, 2, ..., T$ est notée D_t. Cette distribution n'est définie que sur l'échantillon d'apprentissage, nommé LS,
- on suppose que chaque hypothèse faible renvoie une valeur dans $\mathbb{R}$ avec un signe "+" pour ses exemples positifs, et "$-$" pour les exemples négatifs (si elle renvoie des valeurs dans $\{0, 1\}$, on les remplace par $\{-1, 1\}$).

Toute la beauté de l'analyse de [Schapire and Singer, 1998] est de montrer de manière extrêmement simple que la partie cruciale de l'algorithme est en fait le coefficient de normalisation, Z_t. En effet, [Schapire and Singer, 1998] obtiennent :

Théorème 3 *L'erreur sur LS satisfait l'inégalité suivante :*

$$\frac{|\{x : H(x) \neq y\}|}{m} \leq \prod_t Z_t \ . \qquad (13)$$

Ce théorème est important : c'est celui qui permet de trouver l'expression des α_t dans l'eq. 11. Pourtant, si le problème n'admet pas un optimum nul (au moins deux exemples de

[9] A l'époque en effet, [Kearns and Valiant, 1989, Kearns, 1988], le modèle PAC pêchait quelque peu par l'absence de résultats positifs très significatifs. Le Boosting devait en quelque sorte servir à combiner des algorithmes modestes pour trouver un algorithme puissant.

Algorithm 1: `AdaBoost`(LS)

Input: Échantillon d'exemples $LS = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$

for $i = 1$ ***to*** m **do** $D_1(i) = 1/m$;

for $t = 1, 2, ..., T$ **do**

Donner LS avec D_t à l'apprenti faible;

Recevoir l'hypothèse faible $h_t : \mathcal{X} \rightarrow I\!R$;

Choisir α_t :

$$\alpha_t = \frac{1}{2} \log \frac{1 + r_t}{1 - r_t} \quad (11)$$

$$r_t = \sum_{i=1}^{m} D_t(i) y_i h_t(x_i) \quad (12)$$

Mettre à jour :

for $i = 1$ ***to*** m **do**

$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$;

Output: $H_T(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

deux classes différentes partagent la même observation), alors le théorème 3 ne rend pas bien compte de l'évolution de l'erreur : le produit des Z_t ne peut en effet pas converger vers 0. Avec Marc, nous avons repris le théorème, et avons montré le théorème suivant (légèrement abrégé par rapport à [Nock and Sebban, 2001b]) :

Théorème 4 *L'erreur sur LS satisfait l'inégalité suivante :*

$$\frac{|\{x : H(x) \neq y\}|}{m} \leq k_{T+1} \times \prod_t Z_t + \epsilon^* \ , \quad (14)$$

où ϵ^* est l'erreur minimale sur LS (l'erreur "locale" de Bayes), et k_{T+1} un paramètre dépendant de D_{T+1}.

L'avantage de ce théorème est de proposer une mise à jour des distributions légèrement différente de celle de [Schapire and Singer, 1998], mise à jour qui rejoint celle de [Schapire and Singer, 1998] dans les cas où $\epsilon^* = 0$. Dans le cas de notre problème de sélection de variables, où $\epsilon^* > 0$ très fréquemment, nous avons testé cette nouvelle mise à jour. L'objectif était de calculer le gain, en nombre de classifieurs, qu'elle permettait d'atteindre, et ce pour des performances équivalentes en terme de classificiation par rapport à [Schapire and Singer, 1998]. Dans nos expérimentations, nous avons par exemple observé que, dans certains jeux de données, on pouvait économiser jusqu'à 70% du nombre total de classifieurs construits [Nock and Sebban, 2001b]. Dans les (nombreux) cas où l'induction d'un classifieur est coûteuse, un tel gain est largement appréciable même pour de modestes échantillon d'apprentissage ! L'article [Nock and Sebban, 2001b] figure en annexe, page 112.

6.2 Sélection de prototypes

Historiquement, la sélection de prototypes fut d'abord destinée à l'amélioration de l'efficacité de la règle des plus proches voisins [Hart, 1968]. Conceptuellement, la règle des plus proches voisins (PPV) [Cover and Hart, 1967] est probablement la plus simple des règles de classification. D'un point de vue pratique cependant, l'algorithme n'est pas adapté aux grands jeux de données, à cause de ses contraintes de stockage : la classification impose en effet de regarder, au moins dans le pire des cas, tous les exemples stockés. Les premiers travaux dans le domaine ont donc cherché à réduire cette contrainte de stockage. L'approche de [Hart, 1968], la règle des PPV condensés, cherche ainsi à réduire la taille de l'échantillon d'exemples de manière à en conserver un sous-échantillon qui permette une classification correcte des autres exemples[10]. D'autres approches ont suivi, [Brodley, 1993, Gates, 1972, Skalak, 1994, Wilson and Martinez, 2000] (et bien d'autres). Malgré les moyens divers pour clusteriser les algorithmes de sélection de prototypes (représentation des instances, type d'instances sélectionnées, direction de la recherche, etc. [Wilson and Martinez, 2000]), force est de constater que la fonction optimisée dans ces algorithmes est très souvent fonction du taux d'erreur. Avec Marc Sebban, nous avons abordé ce problème avec des outils similaires à nos approches de la sélection de variables : Le gain d'information d'une structure géométrique, et le Boosting.
Dans un autre registre, nous avons aussi abordé l'influence de la sélection d'exemples dans les algorithmes d'induction d'arbres de décision, sur le temps mis pour construire les arbres, sur la taille des arbres obtenus (avec ou sans élagage), et sur la facilité de fouille de ces arbres [Sebban and Nock, 2000b, Sebban et al., 2001a]. Nous avons également abordé tout récemment l'utilisation des Support Vector Machines (SVM) de Vapnik [Vapnik, 1998] pour un même objectif d'amélioration de la règle des PPV par sélection de prototypes [Sebban and Nock, 2001].

Approches basées sur un gain d'information

Ces approches ressemblent assez, dans l'esprit, à ce que nous avons fait en sélection de variables [Sebban and Nock, 2002]. Il s'agissait en quelque sorte, en retirant des exemples, de modifier une structure de voisinage (dans ce cas, celle des k-PPV, où k était une constante), de manière à en conserver la plus grande quantité d'information, tout en s'arrangeant pour que cette structure soit "plus informative" que l'absence même de structure géométrique (ce qui reviendrait au même que de supposer que tous les exemples partagent la même description).

L'article [Sebban and Nock, 2000d] fut notre première contribution autour de cette idée. Il fut suivi par un article accepté à ICML, [Sebban and Nock, 2000c], lui aussi autour de ce principe (il figure en annexe, page 120).

[10]Il n'est pas difficile de montrer, à l'aide du problème du "Set Cover" [Garey and Johnson, 1979], qu'il s'agit là déjà d'un problème déjà difficile à approximer.

Approches dérivées du Boosting

L'utilisation du Boosting pour faire de la sélection de prototypes suit une démarche radicalement différente de notre approche *wrapper* pour faire de la sélection de variables [Sebban and Nock, 1999, Nock and Sebban, 2001b]. Dans ce cadre en effet, le Boosting donne un algorithme très clair, et très simple pour faire de la sélection de prototypes. Notre approche a par ailleurs, je pense, contribué à éclaircir un des points de [Bauer and Kohavi, 1999] (le point 8) sur l'utilisation des algorithmes de PPV avec des méthodes de Boosting.

Dans un algorithme de plus proches voisins, ce sont les exemples issus du voisinage d'un point qui votent pour ce point ci. Par conséquent, si l'on applique un algorithme de Boosting tel **AdaBoost**, le poids α_t (voir algorithme 1) sera affecté non à des hypothèses, mais à des exemples. Sachant que ce poids représente à la fois le classement (son signe) mais aussi la *confiance* en sa classification (sa valeur absolue), il est facile de voir qu'on dispose d'un critère permettant de sélectionner efficacement les exemples. Dans [Nock and Sebban, 2001a, Nock and Sebban, 2000a, Nock and Sebban, 2001d], nous montrons ainsi qu'en effectuant l'algorithme **AdaBoost** sur le voisinage des PPV, si on sélectionne (avant de l'oter) l'exemple e qui a le plus grand coefficient α_e, alors on obtient effectivement (en répétant cette procédure jusqu'à sélection du nombre voulu de prototypes) un algorithme efficace et largement compétitif avec les approches en vogue de la sélection de prototypes.
Le problème majeur de cet algorithme, c'est qu'on ne sait pas quand arrêter la sélection. C'est d'ailleurs aussi un peu le problème du Boosting dans le cas général : comment choisir la valeur du T dans l'algorithme 1 ?
Avec Marc Sebban et Stéphane Lallich, nous nous sommes penchés sur ce problème, et avons proposé un critère d'arrêt et une amélioration de l'algorithme de sélection de prototypes, qui a été publiée à ICML [Sebban et al., 2001b] (l'article figure en annexe, page 130).

6.3 Sélection combinée

L'approche que nous avions eu avec le gain d'information en sélection de variables et d'exemples a ceci d'intéressant que le même algorithme nous servait à faire deux choses différentes. En effet, qu'on supprime des variables ou des exemples, on calcule toujours une structure géométrique sur laquelle on élabore un gain d'information, le RCG (eq. 10). Nous avons donc testé un algorithme combiné, dont les résultats ont été publiés à UAI [Sebban and Nock, 2000a]. A notre connaissance, il s'agissait là de la première approche combinée des problèmes de la sélection de variables et de prototypes, à l'aide du même algorithme. L'article [Sebban and Nock, 2000a] figure en annexe, page 139.

7 Induction de Concepts

Il s'agit là du domaine qui m'a motivé depuis le début de ma thèse. Comme il n'y a pas vraiment de difficulté à présenter les algorithmes en question, j'ai choisi de m'attarder sur les raisons qui ont motivées leurs études. Le lecteur trouvera un article récemment accepté, [Nock, 2002c], en annexe (page 148).

L'induction de concepts de classes diverses se heurte à des difficultés algorithmiques telles, avec la satisfaction des contraintes multiples qu'on s'impose (taux d'erreur, taille des formules, etc.), que la seule ressource reste parfois celle des algorithmes heuristiques. C'est certainement une bonne chose dans la mesure où elle propose une solution à un problème dont on sait pertinemment que la meilleure est dans une large mesure algorithmiquement inaccessible. En revanche, c'est loin d'être satisfaisant. J'ai toujours eu une volonté de cerner les propriétés des algorithmes que je présente brièvement ci-dessous (et des autres, qui figurent en bibliographie), dussent-elles se révéler des plus négatives. Par exemple, il y a toujours eu une recherche de la complexité dans le pire des cas pour la résolution des problèmes d'induction de diverses classes de représentation de concepts avec contraintes sur la taille des formules, ne serait-ce que pour borner ce qu'on pourrait faire de meilleur que les algorithmes que je proposais. A cette époque (j'étais en milieu de thèse), il ne se passait pas une semaine sans que j'obtienne des résultats des plus négatifs, même pour des classes déjà bien connues (à titre d'exemple, les listes de décision de [Rivest, 1987], [Nock and Jappy, 1998b]). Le diagnostic ne se faisait pas attendre pour les quelques algorithmes d'induction dont l'objectif était, dans les meilleurs des cas, de franchir tout de même cette barrière : il s'agissait d'heuristiques. Ce qui m'a souvent dérangé, c'est que c'était presque un vocabulaire palliatif pour un objet algorithmique non identifié. Je me souviens encore du premier rapport que j'ai obtenu sur un article, celui de ICML'95 [Nock and Gascuel, 1995]. Un des rapporteurs me disait que l'algorithme (IDC) était effectivement très joli, conçu de manière originale, et me demandais si je ne pouvais pas ajouter un théorème sur l'approximabilité de l'erreur optimale (en gros, l'algorithme réalise une approximation d'un "Set Cover" [Garey and Johnson, 1979], mais avec un algorithme en quelque sorte "parallèle"). Je n'en ai presque pas dormi pendant plusieurs jours, sans arriver à rassembler les bribes de résultats que j'obtenais pour trouver *la* solution que je cherchais. L'article a été publié, mais sans ce petit résultat. Cette remarque m'a tout de même renforcé dans l'idée qu'une heuristique qui marche sans un minimum de propriétés théoriques qui l'expliquent, c'est quelque chose qui est certes utilisable (bien que parfois...), mais dont les conclusions, les enseignements restent bien trop limités.
Le *speech* de Robert Schapire lors de COLT'98 fut d'ailleurs, au regard de ces considérations, un véritable coup de foudre. Le Boosting, tel qu'il le présentait, se trouvait être un modèle d'achèvement algorithmique, statistique, tout autant que théorique et pratique : une véritable démarche pragmatique de référence en apprentissage automatique. Il s'agit en effet d'une des rares approches tellement belles et puissantes, même si elle souffre encore de quelques problèmes, qui pouvait presque se passer du passage obligatoire des algorithmes habituels d'apprentissage automatique (au sens large), l'épreuve du feu : le test sur les *benchmarks* de l'UCI *repository* contre des algorithmes de référence. Bien sûr, le Boosting n'a pas fait l'économie cette médecine, par exemple dans [Freund and Schapire, 1996]. Cependant, là encore, alors que la plupart des algorithmes traditionnels, lorsque évalués de la sorte, ne peuvent que rarement faire l'économie de tests statistiques pour confirmer leurs qualités, l'article de [Freund and Schapire, 1996] montre des résultats tellement forts que tout test, voire toute discussion est superflue. C'est simplement beau.

Personnellement, le nom de l'UCI, "*repository*", m'a toujours fait penser à une sorte de médecine miracle (sans doute parce que *repository* est l'un des plus proches voisins syntaxiques de *suppository* dans le dictionnaire). Qu'on ne se méprenne pas : je ne dénonce nullement l'utilisation de l'UCI (modérée tout de même : une consommation abusive d'UCI est parfois dommageable à la "santé" algorithmique, et sujette à certaines dérives). Ce ne sont pas, non plus, mes partenaires industriels Martiniquais qui contrediront l'intérêt de l'utilisation raisonnée des approches de tests sur certains jeux de données. Cependant, j'ai souvent l'impression, lorsque je les utilise, de me soumettre à l'effet d'un sirop contre la toux : c'est facile, on se sent mieux, mais ça laisse un certain goût amer.
En réalité, il y a certainement des propriétés à tirer sur certains algorithmes d'apprentissage, peut-être plus intéressantes que les algorithmes eux-mêmes. Par exemple, l'article de [Kearns and Mansour, 1996] fût à mon sens essentiel : non seulement il démontrait l'efficacité des mécanismes récursifs de division des noeuds internes des arbres de décision (dans les algorithmes de type CART, C4.5 [Breiman et al., 1984, Quinlan, 1994]), mais en plus il proposait une amélioration des algorithmes de division qui avait un caractère essentiel : la quasi *optimalité* statistique. La où l'on retrouve la grande beauté de certaines approches théoriques, c'est que le critère qu'ils proposent est *aussi* celui de [Schapire and Singer, 1998] pour le Boosting, lorsqu'on l'applique à l'induction d'un arbre de décision.

Récemment, je me suis remis à l'étude d'un certain nombre d'algorithmes, sans doute de manière beaucoup plus théorique. Avec mon thésard Patrice Lefaucheur (qui est d'ailleurs supporté en partie par des partenaires Martiniquais des domaines de l'assurance et du *credit management*), nous regardons ces problèmes. Un premier article, [Nock, 2002c], vient d'être accepté pour publication. Cet article présente un nouvel algorithme d'induction de comités de décision (une nouvelle classe de représentation de concept dont je reparle plus bas), entièrement basé sur des propriétés de Boosting ou des phénomènes de concentration de variables aléatoires. Parmi les résultats théoriques qu'il démontre, l'un se trouve être amusant, car il étudie une conjecture de [Schapire and Singer, 1998]. L'un des problèmes de Boosting les moins faciles algorithmiquement est relié à l'ordonnancement de classes à affecter à un exemple : plutôt que de décider une ou plusieurs classes, il s'agit d'ordonner (même partiellement) les classes, des plus "probables" aux moins probables. Il s'agit d'un problème répertorié comme étant le *ranking loss* Boosting dans [Schapire and Singer, 1998]. A partir du moment où le nombre de classes est une variable, [Schapire and Singer, 1998] ont conjecturé la NP-Complétude d'un problème général de *ranking loss* Boosting. Je me suis intéressé à ce problème, sachant qu'après tout nombre de problèmes que j'avais déjà étudié se trouvaient être non seulement NP-Complets, mais encore très difficiles à approximer. Fort de mon expérience, je m'apprêtais presque à utiliser la technique de Boosting de résultats négatifs... Et là, surprise : ce problème, qui est un problème de minimisation, est polynômial, mais nécessite des algorithmes complexes de résolution. J'ai aussi pu trouver un algorithme d'approximation asymptotiquement optimal (quand l croît), très simple et quadratique en temps. Sans avoir été frustré par ces résultats positifs, j'ai tout de même étudié le problème de maximisation correspondant, chose qui n'a absolument aucun intérêt en classification. Là ou le Boosting fait encore très fort, c'est que c'est *ce* dernier problème qui, par chance, est NP-Difficile. J'avais trouvé

	adhere	?	¬adhere
$(\neg Ongoing\_project) \wedge (\neg Education)$	-1	-1	1
$(\neg Ongoing\_project) \wedge (Lengthy\_proc) \wedge (\neg Wholesaler)$	1	-1	1
default $\vec{D}$	0.32	0.68	0

Fig. 2 – La DC obtenue sur le jeu de données de la DDAF Martinique.

mon candidat, et l'honneur était sauf, d'une certaine manière ! L'article [Nock, 2002c] figure en annexe, page 148.
Un autre article de ce type est en cours de soumission avec Patrice, qui concerne un nouvel algorithme de Boosting, qui conserve sa propriété de Boosting dans le cadre de l'apprentissage robuste (voir l'eq. 5, page 8, pour la correspondance apprentissage PAC-apprentissage robuste), chose qui est impossible pour **AdaBoost** d'après un résultat de [Höffgen and Simon, 1992]. L'actualité des résultats de ce dernier article fait qu'ils n'ont peut-être pas (encore) leur place dans cette Habilitation, mais je reste Confiant : ils auront leur place dans une thèse de Doctorat.

Pour en revenir à ce vaste sujet de l'induction, depuis le début de ma thèse, je m'intéresse à une classe de représentation de concepts dont je n'avais vu d'équivalents nulle part (ni Olivier Gascuel d'ailleurs, c'est lui qui m'a suggéré de les regarder d'un peu plus prêt alors que je commençais mon stage de DEA), les comités de décision (DC). Pour reprendre des commentaires récents à propos de la faiblesses d'interprétabilité de résultats du Boosting [Bauer and Kohavi, 1999], on pourrait presque dire que DC "représente" une partie facilement interprétable des fonctions discriminantes polynômiales, puisqu'elle apporte des réponses agréables à certaines problématiques, mêmes récentes, comme celle de la construction de séparateurs linéaires dans les cas multi-classes.

La figure 2 présente un exemple de DC obtenue sur un jeu de données de la Direction Départementale de l'Agriculture et de la Forêt de Martinique[11]. Sans entrer dans les détails, il s'agit de prédire si un agriculteur est pour la souscription d'un Contrat Territorial d'Exploitation (CTE), contre, ou bien dans l'expectative. Ces contrats sont d'une nature originale, car ils sont parmi les premiers à proposer des financements aux agriculteurs (et des formations à de nouvelles techniques), *en échange* de l'adaptation ou du changement de leurs cultures. Parmi la cinquantaine de variables de description (de nature extrêmement diverses), un algorithme de [Nock, 2002c] en a retenu 4 (Booléennes) :

- $Ongoing\_project$, vraie ssi l'agriculteur dispose d'un projet pour son exploitation,
- $Education$, vraie ssi l'agriculteur a eu une formation (initiale, ou stages),
- $Lengthy\_proc$, vraie ssi il trouve que la procédure d'adhésion est longue,
- $Wholesaler$, vraie ssi il vend toute sa production à un grossiste.

[11] Merci à L'ENESAD, la DDAF Martinique et Lise Jean-Louis pour l'autorisation de publication de ce résultat.

La classification d'un exemple à partir d'un tel DC se fait simplement en sommant les vecteurs dont les règles sont satisfaites par l'exemple, puis en attribuant la classe dont la composante est maximale. Si plusieurs classes ont la composante maximale, alors on prend, parmi ces classes, celle qui a la composante maximale du vecteur par défaut. Par exemple, les agriculteurs n'ayant pas de projet en cours, et n'ayant pas bénéficié de formation auront tendance à ne pas contracter un CTE... Chose intéressante, et que ne mettaient pas du tout en avant les arbres de décision (nous avons aussi testé C4.5 [Quinlan, 1994]), le DC met aussi en évidence le rôle du facteur éducation dans l'attitude des agriculteurs. Le principal algorithme d'induction de [Nock, 2002c] repose en fait sur une large amélioration, dans un cadre formel, d'autres algorithmes d'induction de DC, qui figurent dans [Nock and Gascuel, 1995, Nock and Jappy, 1999b], et dans une version qui date encore d'avantage, dans [Gascuel et al., 1998].

Je me suis aussi intéressé à un nouvel algorithme d'apprentissage pour les listes de décision (DL) de [Rivest, 1987], en utilisant un algorithme radicalement différent de CN2 [Clark and Boswell, 1991]. A l'époque en effet, DL n'avait pas d'algorithmes d'induction calqués sur le modèle de C4.5 [Quinlan, 1994] (ou CART [Breiman et al., 1984]). Les listes de décision sorties par CN2 étaient souvent beaucoup plus grandes que celles sorties par C4.5 sur les mêmes problèmes, sans être nécessairement plus fiables, ce qui contredisait les observations élémentaires sur les capacités d'expression de DL par rapport aux arbres de décision [Rivest, 1987]. Comme je l'ai déjà présenté, on sait, depuis [Kearns and Mansour, 1996], pourquoi C4.5 a tendance à construire des arbres de décision *relativement* petits à taux d'erreur fixé. On sait même comment l'améliorer. Ceci repose, brièvement, sur la minimisation d'un critère très particulier sur la partition de l'échantillon d'exemples induite par la formule, et bien sûr, sur l'addition de nouveaux tests (donc sur la modification de cette partition) qui vont contribuer à la minimisation de ce critère. Partant du principe qu'une liste de décision induit aussi très simplement une partition de l'échantillon d'exemples (différente de celle d'un arbre de décision), nous avons adapté ce schéma d'induction, et obtenu des résultats qui sont beaucoup plus en accord avec les relations qu'entretiennent DL et les arbres de décision, en particulier sur la taille des formules [Nock and Jappy, 1999a, Nock and Jappy, 1998b]. Ce schéma d'induction a d'ailleurs été étendu depuis, à d'autres classes [Nock et al., 1999].

Enfin, je me suis intéressé plus récemment, dans le cadre d'un contrat avec le Secrétariat d'état aux DOM-TOM dont je suis responsable, à la construction d'algorithmes de classification performants dans le cadre de la prédiction de la pollution marine en zone inter-tropicale. Ce problème nous a par ailleurs motivé pour la reprise et l'amélioration d'une borne d'erreur sur l'algorithme du plus proche voisin dans le cas d'échantillons d'exemples de taille finie [Drakopoulos, 1995], chose que nous avons par ailleurs obtenu dans [Nock and Sebban, 2001c] (l'article figure en annexe, page 181). Les résultats sur notre problème de pollution marine figurent en partie dans [Nock et al., 2000].

Troisième partie

Classification, apprentissage automatique et segmentation d'images

8 Algorithmique, classification, concentration et segmentation

Depuis le début du XX^e sc., et le mouvement Gestalt en psychologie [Wertheimer, 1938], il est reconnu que la segmentation joue un rôle important dans la vision humaine. Pourtant, aujourd'hui encore, cette tâche de classification conceptuellement simple - grouper les pixels d'une image en régions *perceptuellement* distinctes - reste un des grands challenges de la communauté vision [Felzenszwalb and Huttenlocher, 1998]. La tâche est d'autant plus difficile que la segmentation sur ordinateur ne se fait, dans le cas général, sans aucun modèle de l'image, ni connaissance à priori. Seule l'information fournie par chaque pixel (couleurs, luminosité, etc.) et l'organisation spatiale des pixels sont susceptibles d'être utilisées. En quelque sorte, on aimerait tirer de cette information de (très) bas niveau une organisation perceptuelle de l'image, sachant bien qu'en plus de nombreuses images peuvent aboutir à des interprétations différentes selon les personnes [Shi and Malik, 2000]...
Le compromis semble être, comme le suggèrent [Shi and Malik, 2000], que la segmentation ne doit pas chercher à isoler les *bonnes* régions d'une image. Elle devrait plutôt aboutir à une partition *possible*, parmi celles qui seraient les plus probables [Shi and Malik, 2000]. Par ailleurs, sachant que la segmentation n'est souvent qu'une étape préliminaire au traitement d'images, il vient de fait qu'elle doit se faire en une complexité aussi basse que possible. Le simplement polynômial n'est d'ailleurs même pas intéressant : il faudrait qu'elle s'approche de la complexité linéaire, *i.e.* optimale.

La problématique de la segmentation d'image ressemble beaucoup à celle de l'apprentissage. Elle partage la notion d'incertitude, la contrainte de complexité. Si, par ailleurs, on suppose que l'image observée n'est après tout pas plus déterminée par avance que le tirage d'un exemple, alors l'analyse d'un algorithme de segmentation peut aussi se faire sur la base d'un équivalent de l'"oracle" du modèle PAC. Cet oracle rétribuerait des scènes plutôt que des exemples, scènes sur lesquelles chaque pixel serait échantillonné selon une distribution *inconnue* mais fixée pour l'objet en question (dont provient le pixel). Ce sont quelques remarquent qui m'ont convaincu, dès 1998 environ [Fiorio and Nock, 1998], de tester l'apport des outils algorithmiques, probabilistes, utilisés (et perfectionnés) en apprentissage automatique. Cette recherche, que j'ai mené en parallèle avec mes autres préoccupations en apprentissage et Data Mining, a d'ailleurs porté ses fruits, puisqu'un des derniers articles consacré a été publié dans l'IEEE CVPR ([Nock, 2001b][12]).

Si elle a sans aucun doute sa place à coté de mes travaux en apprentissage automatique ou Data Mining, c'est parce qu'en matière d'imagerie, ma recherche a convergé vers des problématiques de classification, mais aussi parce qu'elle a cherché à utiliser de plus en plus des outils qui avaient été développés en apprentissage automatique, et qui étaient susceptibles, sous réserve de transformations ou d'adaptations, de s'appliquer (potentiellement mieux) au domaine de la segmentation d'images. Ces travaux, ce sont ceux de [Kearns and Mansour, 1998], qui montrent comment élaguer un arbre de décision de ma-

[12]Cette année, la conférence a été la plus grande de ses 25 ans d'histoire. L'article [Nock, 2001b] a été pris en présentation orale, dont le taux de sélection était <8%.

nière à trouver, avec un risque statistique contrôlable (sous réserve qu'on dispose de suffisamment d'exemples, évidemment...), le sous-arbre optimal sur tout le domaine $\mathcal{X}$. Ce qui est intéressant dans l'approche de [Kearns and Mansour, 1998], dans le contexte des arbres de décision, c'est qu'ils avaient précisément montré auparavant l'intérêt des mécanismes de division récursifs des arbres de décision, pour obtenir un grand arbre, en optimisant rapidement l'erreur sur l'échantillon d'apprentissage, [Kearns and Mansour, 1996]. L'article [Kearns and Mansour, 1998] était en quelque sorte le "complément nécessaire" pour que la réduction de l'erreur sur l'échantillon d'apprentissage s'accompagne aussi d'une réduction d'erreur sur tout le domaine $\mathcal{X}$. Dans le cadre de la segmentation d'images, on peut aussi considérer que l'image contient autant d' "observations" que de pixels, et que l'on recherche à regrouper ces pixels entre eux, tout en maîtrisant les bruits éventuels de la scène (par avance inconnue). Regrouper les pixels, c'est donc presque comme élaguer un (très grand) arbre de décision. Tout en passant par une extension des modèles usuels d'apprentissage automatique (les bornes de Chernoff-Hoeffding sont inutilisables, sauf à faire des hypothèses trop restrictives, car les équivalents des classes en apprentissage automatique sont en réalité des tuples de couleurs en analyse d'images), j'ai petit à petit développé une catégorie d'algorithmes de segmentation d'images, qui contrastent avec un certain nombre d'approches actuelles de la discipline. La machinerie mathématique sous-jacente est en effet largement compatible avec les exigences algorithmiques de la segmentation d'images, ce qui est très loin d'être le cas dans toutes les approches utilisant des outils mathématiques pour segmenter l'image. Même les plus utilisées aujourd'hui, comme [Shi and Malik, 2000], ne font pas l'économie de lourdes procédures de manipulations matricielles. Par ailleurs, même s'ils disposent d'algorithmes d'approximation prétendus plus rapides (en particulier pour leur calcul de valeurs propres, dans [Shi and Malik, 2000]), ils sont utilisables seulement dans des cas particuliers. Les sections suivantes détaillent chronologiquement l'évolution de mes travaux de recherche dans ce domaine.

9 Réduire la complexité...

Le premier article que j'ai publié dans ce domaine est [Fiorio and Nock, 1998]. La motivation essentielle de ce travail était de concilier deux objectifs qui, à mon sens, s'excluaient trop souvent des publications en segmentation d'images. Le premier était le soucis de maintenir une complexité minimale de l'algorithme de segmentation. Le second, la nécessité d'obtenir une segmentation sur laquelle on pourrait énoncer un certain nombre de propriétés statistiques d'homogénéité sur la segmentation.
Ce qui m'a frappé dans la littérature sur la segmentation d'images, c'est de voir qu'il y a probablement approximativement autant d'échelles de valeurs sur lesquelles on place les propriétés des algorithmes, que d'articles. L'échelle la plus élastique est sans doute celle de la complexité, même si on ne la désire que raisonnable (*i.e.* sans aller jusqu'à notre volonté de s'approcher de l'optimum). Dans l'article [Xu et al., 1998] par exemple, l'efficacité de l'algorithme est jugée sur la base d'un temps d'exécution qui prend "moins d'une *minute* CPU sur une *SPARC-20*". Dans un autre article, [Shi and Malik, 2000] donnent des temps d'exécution de l'ordre de "*2 minutes* sur des images 100×120". Plus subtil, [Comaniciu and Meer, 1997] donnent des temps d'exécution de "moins de 10 secondes sur

des images 512×512", mais sur le dernier modèle d'UltraSPARC...
D'un point de vue statistique, il y a aussi des choses à dire : les algorithmes se basent parfois sur des hypothèses de distribution ou des modèles qui n'ont aucune raison d'exister d'avantage que d'autres sur les images traitées en pratiques, ou bien alors font reposer leurs analyses sur des hypothèses avouées trop restrictives [Zhu and Yuille, 1996]. Ce point est certainement le plus discutable, mais il est aussi sans doute le plus difficile à résoudre : l'obtention de résultats forts passe parfois nécessairement par ce genre d'hypothèses [Zhu and Yuille, 1996].

Mon premier article, produit avec Christophe Fiorio ([Fiorio and Nock, 1998], voir en annexe, page 189), fut une première tentative quasi extrémiste d'approche du problème, puisque l'algorithme était optimal en temps et en espace (il utilisait la structure de données `Union-Find` de [Fiorio and Gustedt, 1996]), et ne reposait sur aucun modèle *a priori* sur la structure de l'image. Il fonctionnait selon un schéma très simple :

- on débute avec l'image contenant autant de régions que de pixels,
- l'image est parcourue récursivement, de l'échelle du pixel à celle de l'image tout entière,
- on teste la fusion entre régions distinctes correspondant à des pixels adjacents, soit horizontalement, soit verticalement.

Chaque test de fusion se faisait selon un prédicat de fusion, que nous voulions non paramétrique. Nous le voulions aussi suffisamment général pour qu'il intègre éventuellement des contraintes dépassant l'information basique de l'image (par exemple, l'appartenance à certains objets, une information sémantique, etc.). Ceci s'est fait, mais pas sans le sacrifice d'une partie de l'attrait statistique qu'il aurait sans doute fallu aussi donner à un tel critère. Cette partie statistique, ce fut l'étape suivante.

10 ...sans réduire la qualité statistique

L'étape suivante fut la construction d'un test statistique qui permette autant que possible un jugement raisonnable sur la qualité d'une fusion envisagée entre deux régions. L'article [Fiorio and Nock, 2000a] est notre première tentative, avec Christophe Fiorio. L'idée, à l'époque, était de commencer à utiliser les résultats et les boites à outils utilisées en apprentissage automatique, autour des phénomènes de concentration de variables aléatoires [McDiarmid, 1998]. Ils sont, depuis l'invention du modèle PAC, un des outils principaux de l'étude des algorithmes d'apprentissage [Kearns and Vazirani, 1994]. La raison qui nous motivait était alors un modèle de génération d'image supprimant toutes les hypothèses restrictives (normalité, homoscédasticité, etc.) autour de la génération de l'image. Évidemment, moins on fait d'hypothèses sur la structure d'une image, et, intuitivement, moins les résultats qu'on peut en tirer sont "forts" (de manière générale)... mais plus ils s'appliquent !

Dans ce modèle (voir [Fiorio and Nock, 2000a]), l'image observée I est générée à partir

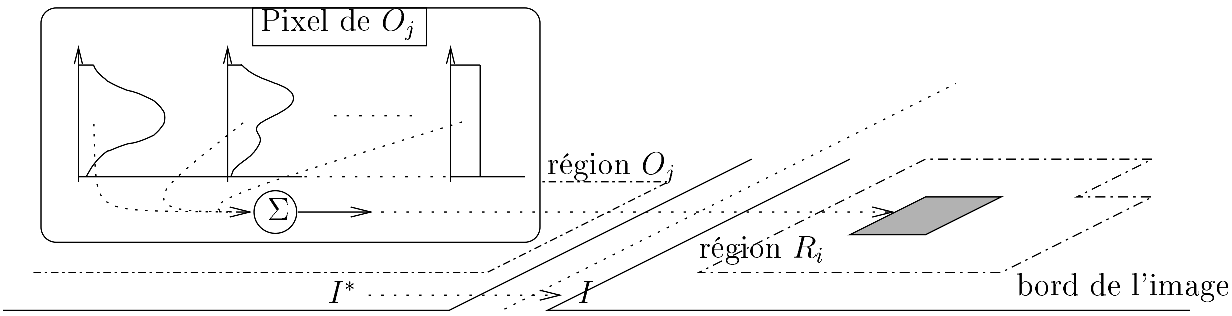

FIG. 3 – Génération d'un niveau de couleur (**R**, **G** ou **B**) de I^* vers I.

d'une image théorique, dans laquelle on n'observerait en quelque sorte que des distributions (et pas, par exemple, des niveaux de couleurs), voir figure 3. Dans cette image théorique, on associe à chaque pixel un nombre Q de distributions par couleurs (ou par variable, si on ne décrit pas I dans le système classique **RGB**). Chose cruciale, on ne fait aucune hypothèse de distribution sur les $3 \times Q \times |I|$ distributions de l'image, mise à part une chose fondamentale : l'indépendance dans l'échantillonnage des distributions (pour des raisons techniques). Dans cette image théorique, I^*, il existe bien évidemment un moyen pour retrouver les objets qu'on cherche à approximer dans l'image observée I : à l'intérieur d'un même objet de I^*, pour chaque variable (**R**, **G**, **B** si on représente l'image en coordonnées **RGB**), la somme des espérances mathématiques des Q distributions est la même pour tous les pixels de cet objet. Cette espérance est en retour différente pour au moins un niveau de couleur entre deux objets adjacents de I^*.

La partie fondamentale de ce modèle est sa grande liberté dans les distributions de chaque pixel : en dehors de la condition sur les espérances, on peut très bien imaginer des cas ou *chacune* des $3 \times Q \times |I|$ distributions est différente des autres...
C'est bien évidemment ici qu'intervient le facteur Q : c'est en quelque sorte un “prix à payer” pour autant de liberté dans la génération des objets. Plus sa valeur est petite, plus le modèle est général, mais moins les résultats qu'on pourra trouver seront fiables, et plus on risquera en particulier la sous-segmentation de l'image (en utilisant les inégalités de concentration telles qu'on le verra dans toute la suite). Cette dernière situation correspond aux cas où chaque objet de I^* se trouve dans une seule région de la segmentation, mais il existe des régions dans la segmentation de I comportant plus d'un objet véritable de I^*.

Ce modèle, nous l'avons intégré à un prédicat de test de fusion de régions, que nous avons intégré à l'algorithme `scanline` de [Fiorio and Gustedt, 1996]. Cet algorithme procède simplement en faisant un parcours de gauche à droite et de haut en bas, et pour chaque pixel rencontré, teste (si c'est possible) la fusion de sa région avec celles correspondant aux pixels du haut et de gauche. Toute la difficulté de l'approche de type concentration dans un tel algorithme est que l'algorithme n'est pas bien adapté *a priori* à ce genre d'approche. Pour comprendre ceci, il est nécessaire de voir qu'une approche basant la fusion sur des inégalités de concentration est nécessairement très optimiste lorsque les régions sont *petites*. On aura donc tendance à accepter la fusion de petites régions avec d'autres même si elles ne sont pas apparemment issue du même objet dans I^*. Le prédicat de fusion devient ensuite de plus en plus exigeant lorsqu'on teste la fusion de régions de grandes tailles. Si on parcourt l'image ainsi que nous l'avions fait dans `scanline`, le risque est grand pour qu'en commençant à fusionner la partie haute d'une région, on fasse l'erreur de la fusionner également avec une partie de tout ce qui se trouve à sa gauche ou à sa droite dans I^*. Pour réaliser notre algorithme, il nous fallait donc un prédicat "taillé sur mesure", non seulement sur la quantité dont nous allions mesurer les phénomènes de concentration, mais aussi pour l'inégalité de concentration que nous allions utiliser.

Cette quantité que nous avons mesuré est une "variation de variabilité intra-region" dans l'image au cours des fusions. Prenons par exemple le cas ou l'image est exprimée en niveaux de gris (dans le cas **RGB**, on multiplie simplement le nombre de tests de fusion, en en faisant trois au lieu d'un).

Soit $\{R_1, R_2, ..., R_n\} = P(I)$ une partition de l'image observée I. on appelle $p_{i,j}$ la valeur observée du j^e pixel de la région R_i (*i.e.* la somme des Q tirages aléatoires selon les Q distributions du pixel), et $p_{i,j,k}$ la sortie de la k^e distribution pour le j^e pixel de la région R_i. Ainsi, on aurait $\forall i \in \{1, 2, ..., n\}$, $\forall j \in \{1, 2, ..., |R_i|\}$, $p_{i,j} = \sum_{k=1}^{Q} p_{i,j,k}$. Appelons $\overline{R}_i$ la moyenne des niveaux de gris pour R_i, et $\overline{I}$ celle de l'image observée tout entière I. Alors on définit quatre niveaux de variabilité :

- La Variabilité Intra-Pixel (IPV)

$$V_{\mathbf{p}}(I, P(I)) = \sum_{i,j,k} \left(p_{i,j,k} - \frac{p_{i,j}}{Q}\right)^2 . \qquad (15)$$

- La Variabilité Intra-Région (IRV)

$$V_{\mathbf{i}}(I, P(I)) = \frac{1}{Q} \sum_{i,j} \left(p_{i,j} - \overline{R}_i\right)^2 . \qquad (16)$$

- La Variabilité Extra-Région (ERV)

$$V_{\mathbf{e}}(I, P(I)) = \frac{1}{Q} \sum_{i} |R_i| \left(\overline{R}_i - \overline{I}\right)^2 . \qquad (17)$$

- La Variabilité Totale de I

$$V(I) = \sum_{i,j,k} \left(p_{i,j,k} - \frac{\overline{I}}{Q}\right)^2 . \qquad (18)$$

On a bien sûr

$$V(I) = V_{\mathbf{p}}(I, P(I)) + V_{\mathbf{i}}(I, P(I)) + V_{\mathbf{e}}(I, P(I)) \ , \qquad (19)$$

et il est facile de voir qu'au fur et à mesure de la fusion de régions, c'est l'ERV qui va petit-à-petit passer dans l'IRV. En fusionnant deux régions R et R', cet accroissement de variabilité intra-région, $\Delta_{\mathbf{i}}(R, R')$ vaut :

$$\Delta_{\mathbf{i}}(R, R') \quad = \quad \frac{|R||R'|}{Q(|R| + |R'|)} \left(\overline{R} - \overline{R'}\right)^2 \quad . \tag{20}$$

Dans [Fiorio and Nock, 2000a], nous avons tiré une inégalité de concentration de $\Delta_{\mathbf{i}}(R, R')$ pour des régions R et R' qui proviendraient d'un même objet dans I^*. A partir de là, il fût facile d'en tirer un prédicat de fusion, permettant de savoir si on doit fusionner ou non R et R'. L'article [Fiorio and Nock, 2000a] figure en annexe, en page 199.

Dans [Fiorio and Nock, 2000b], nous avons réalisé la première extension de l'article précédent, permettant de modifier l'algorithme pour qu'il prenne mieux en compte le prédicat : c'est ainsi que l'algorithme procède initialement par un ordonnancement de tous les pixels de l'image selon un ordre croissant de variabilité entre le pixel et son voisinage, pour qu'ensuite les tests de fusions commencent par les pixels dans les régions relativement homogènes. L'étape suivante fut une intégration complète d'un ordre très particulier, qui d'une certaine manière était suggéré par le prédicat de fusion de régions. L'article [Fiorio and Nock, 2000b] figure en annexe, page 210.

11 Un compromis

Le problème de [Fiorio and Nock, 2000b] est que, si l'ordre s'est révélé utile en pratique, en revanche il n'a pas pu être justifié formellement d'une quelconque manière. Je me suis intéressé à un nouvel algorithme dans lequel mon objectif était d'obtenir à la fois une borne de concentration, et un algorithme permettant d'approcher l'algorithme optimal de test des fusions de régions. C'est ce qui est exposé dans [Nock, 2001b]. Tout d'abord, le modèle de génération d'image est le même : on dispose d'une image théorique, inconnue, à partir de laquelle l'image observée est générée. Ce qui change, c'est le critère permettant d'évaluer "l'écart" entre deux régions dont on teste la fusion. Ce critère repose simplement sur la moyenne observée de la région, et sur une inégalité de concentration simple, faîte pour l'occasion. $\forall \mathbf{a} \in \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$, appelons $\overline{R}_{\mathbf{a}}$ la moyenne observée de la couleur $\mathbf{a}$ pour la région R de I, et $\mathbf{E}_{\mathbf{a}}(R)$ l'espérance mathématique pour la région R et la couleur $\mathbf{a}$ dans l'image théorique I^*. Si $\mathbf{a} \in \{1, 2, ..., g\}$, et si on dénombre $N_{|R|}$ régions possibles de taille $|R|$, alors on a

Théorème 5 $\forall \delta > 0$*, la probabilité pour qu'il existe une région de taille* $|R|$ *telle que*

$$|\overline{R}_{\boldsymbol{a}} - \boldsymbol{E}_{\boldsymbol{a}}(R)| \quad \geq \quad g\sqrt{\frac{1}{2Q|R|} \log \frac{2N_{|R|}}{\delta}} \tag{21}$$

est $< \delta$.

Ce premier théorème donne le prédicat de fusion $\mathtt{P}(R, R')$ permettant de savoir si on fusionne deux régions R et R'. On a en effet :

$$\mathtt{P}(R, R') = \begin{cases} \mathtt{vrai} & \text{si } \forall \mathbf{a} \in \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}, |\overline{R'}_{\mathbf{a}} - \overline{R}_{\mathbf{a}}| \leq b(R) + b(R') \\ \mathtt{faux} & \textit{sinon} \end{cases} ,$$

avec $b(R) = g\sqrt{\frac{1}{2Q|R|}\left(\ln\frac{2}{\delta'} + \log N_{|R|}\right)}$. L'avantage de ce prédicat, c'est qu'il suggère aussi l'algorithme qui devrait être utilisé en théorie. En effet, si on était capable de faire les tests de fusion à l'intérieur d'une vraie région de I^* avant de les faire entre une partie quelconque de cette région et une voisine, alors on peut démontrer qu'avec une forte probabilité, la segmentation obtenue serait une sous-segmentation de l'image I^* : toute vraie région de I^* serait dans une seule région de I, mais certaines régions de I contiendraient plus d'une vraie région. A partir de là, le δ du théorème 5 peut être adapté pour diminuer la "force" de cette sous-segmentation [Nock, 2001b].

Ceci à l'avantage de proposer un algorithme très simple d'ordonnancement avant de faire les tests de fusion : il suffit d'ordonner les couples de pixels adjacents selon une valeur croissante d'une valeur de similarité (par exemple, le maximum de l'écart dans les trois coordonnées **R**, **G**, **B**). Ensuite, il n'y a plus qu'à traverser cette liste ordonnée de couples, en testant la fusion entre les régions des pixels correspondant, si bien sûr les pixels n'appartiennent pas déjà à la même région. Une des qualités de cet algorithme, c'est de ne nécessiter aucun paramétrage *ad hoc*, contrairement aux approches habituelles en segmentation d'images [Felzenszwalb and Huttenlocher, 1998, Shi and Malik, 2000, Zhu and Yuille, 1996]. Pour plus de détails, on pourra consulter [Nock, 2001b], qui figure en annexe, page 214.

La page Web

`http ://www.martinique.univ-ag.fr/~rnock/Articles/CVPR01/`

présente aussi un certain nombre de tests qui ne figurent pas dans la trentaine initiale de CVPR, avec le code binaire de l'algorithme, et la présentation (au format `.ppt`) faîte à CVPR en décembre 2001. Les figures 4, 5 et 6 résument un certain nombre de résultats obtenus sur certaines images (attention, les tailles ne sont pas celles des images initiales). Chose à signaler, toutes les segmentations ont été obtenues

- sans manipulation des images d'aucune sorte (débruitage, etc.),
- avec les même paramètres pour l'algorithme (il n'y a pas eu d'optimisation en fonction du domaine, de l'image, etc.).

Lors de CVPR, une discussion avec des personnes venant de laboratoires d'Intel et Sharp a soulevé l'intérêt d'étendre l'algorithme au cas ou l'on fait de la segmentation d'images avec occlusions, c'est-à-dire, pour reprendre le modèle de génération d'images, dans les cas où il existe des régions de I^* qui sont "coupées" par d'autres. L'article [Nock, 2002a] présente cette extension. Il vient d'être accepté, et figure en annexe, page

223.

Je poursuis actuellement deux grandes directions de recherche dans le cadre de cet algorithme de segmentation : tout d'abord, j'aimerais rendre l'algorithme tolérant à certaines formes de bruits très retors dans l'image (une version a été acceptée à ICASSP'02, [Nock, 2002b]), mais surtout, j'aimerais calculer formellement l'erreur faîte par l'algorithme. Dans le cas de l'élagage des formules Booléennes telles les arbres de décision [Kearns and Mansour, 1998], on sait que ce genre de procédure (qui ne marche pas en pratique, fort heureusement pour des considérations de taille des données, restriction que nous n'avons pas en segmentation d'images) est en théorie très puissante sur de grandes formules et avec beaucoup d'exemples. Je dispose de résultats préliminaires étendant ceux de [Kearns and Mansour, 1998], en cours de soumission. Sans entrer dans le détail, j'ai obtenu une borne généralisant celle de [Kearns and Mansour, 1998], qui reste très bonne dans le cas de la segmentation d'images, pour peu qu'on dispose d'un algorithme qui fonctionne à information incomplète (il ne connaît pas la position des vraies régions dans I^* !), mais qui connaît un ordre particulier de test de fusion. Cet algorithme n'est pas implémentable, mais il se trouve que les approches [Nock, 2001b, Nock, 2002a] sont précisément des approximations, très rapides, de cet algorithme.

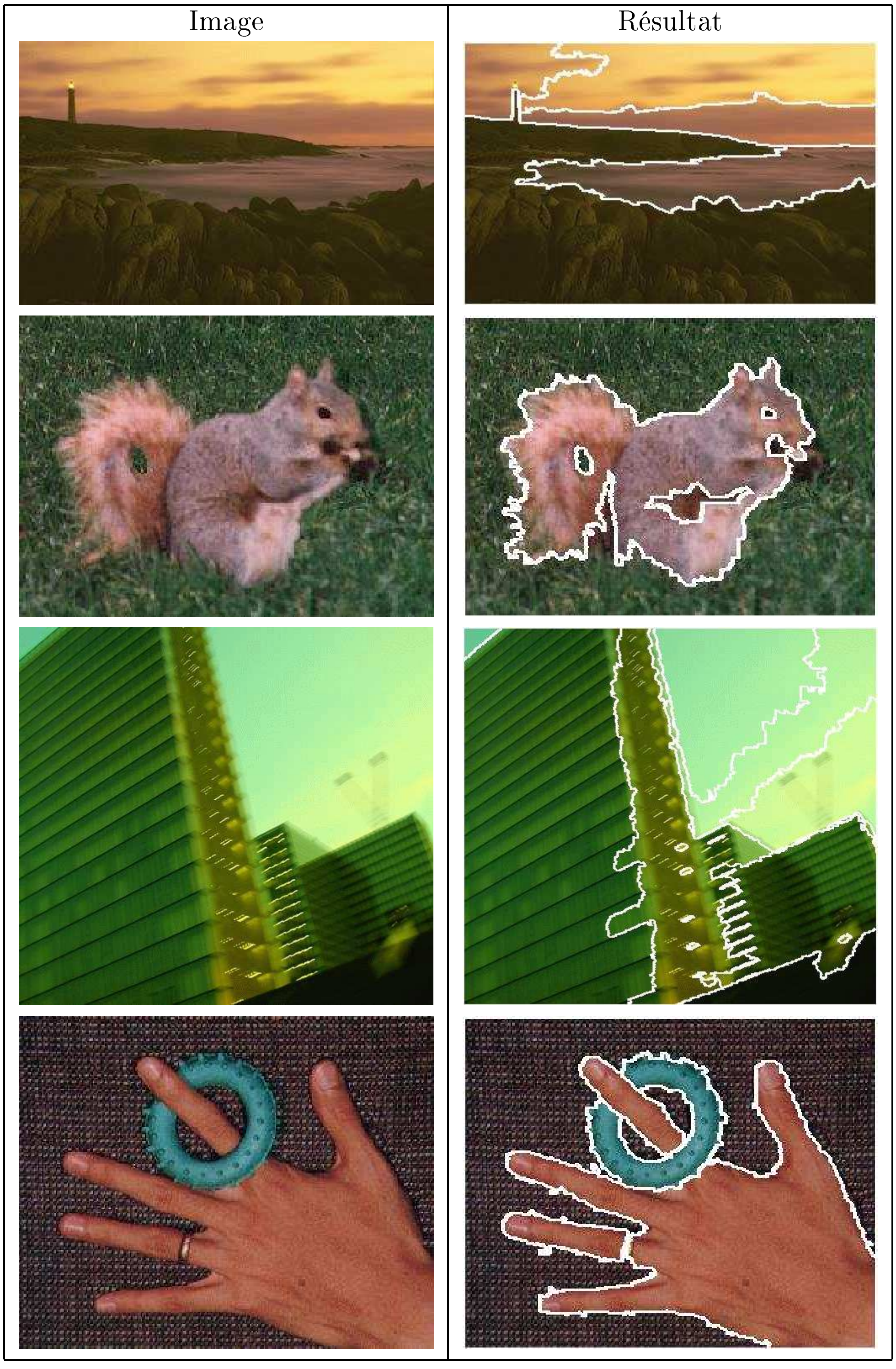

FIG. 4 – Quelques expérimentations sur diverses images. Les résultats de segmentation montrent l'image initiale sur laquelle on a superposé en blanc ou en noir la segmentation obtenue.

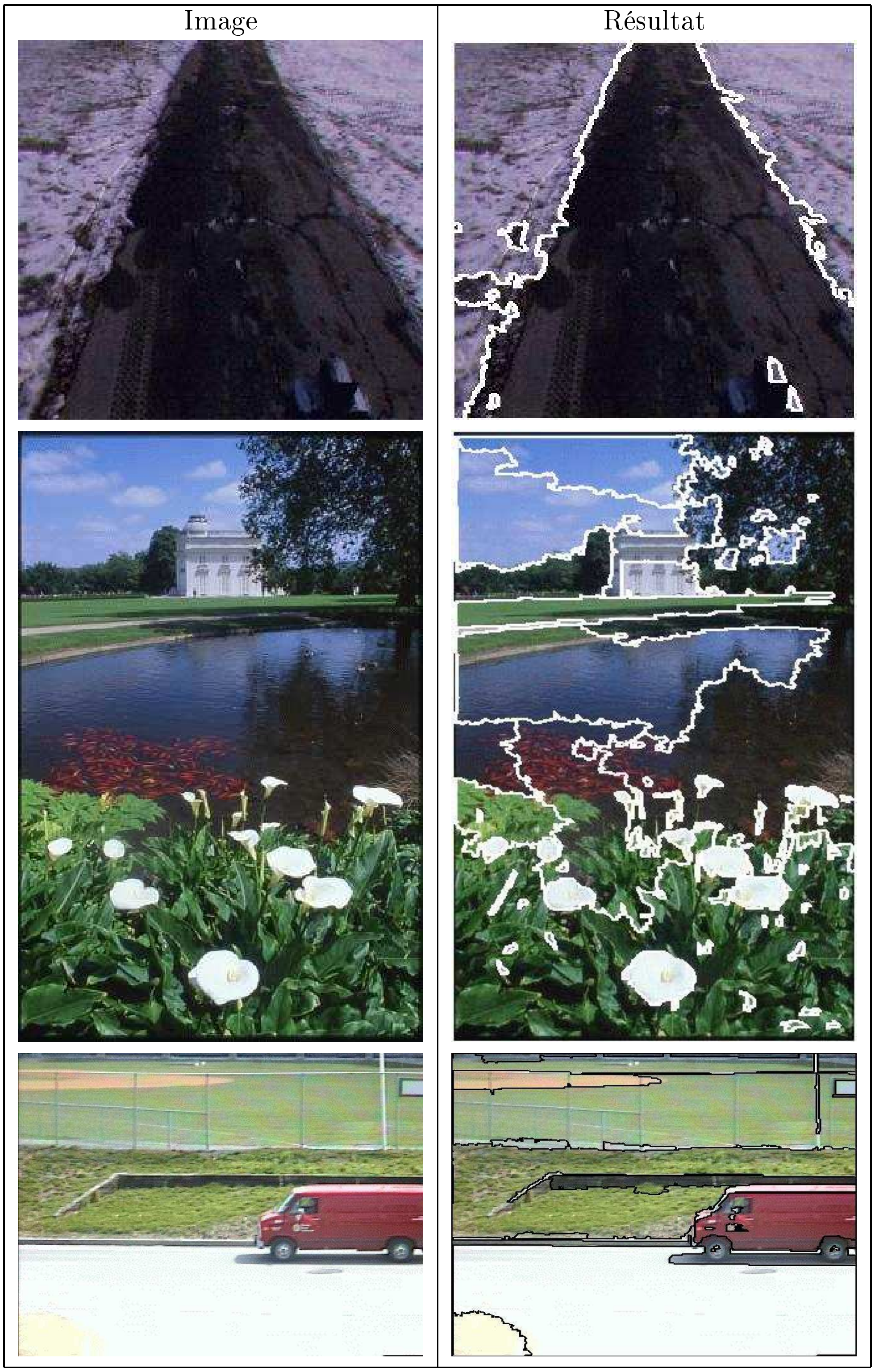

FIG. 5 – D'autres résultats.

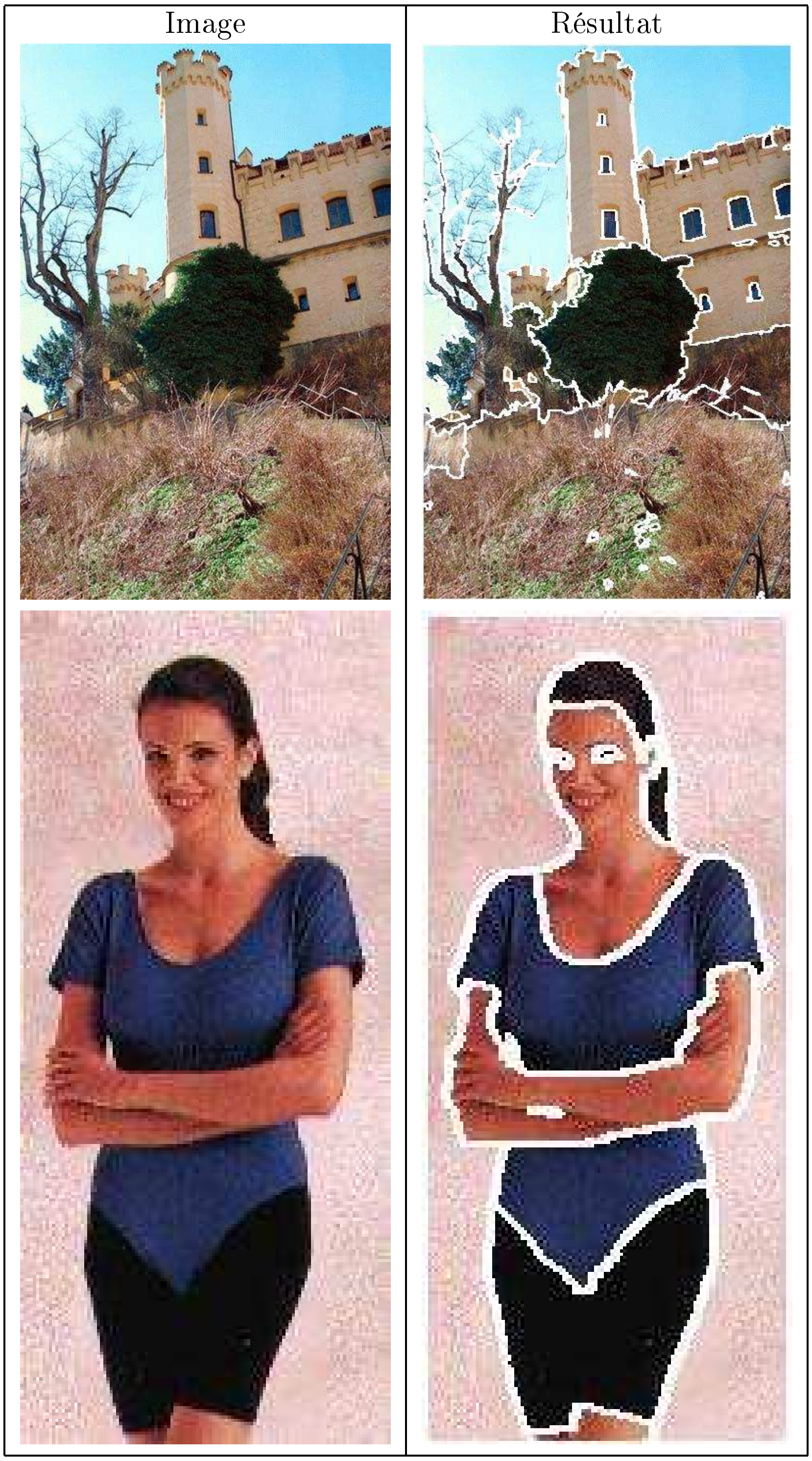

FIG. 6 – Encore d'autres résultats.

Conclusion

Cette rédaction fut la première synthèse de mes travaux depuis bien longtemps (bien sûr, tout est relatif). J'espère simplement ne pas y avoir perdu le lecteur, tant j'ai eu moi-même l'impression d'accumuler des résultats sans toujours savoir dans quel boite les mettre. De toute façon, la diversité de mes recherches est aussi à l'image de celle de ma formation, de mon parcours professionnel, tout autant que de mon parcours géographique. Dans l'attente qu'elle s'affirme dans d'autres domaines, je reste convaincu qu'elle peut être, en recherche, la source de nombreuses idées, de nombreuses collaborations, contrats, et donc, de fructueuses publications !

Depuis que je suis arrivé en Martinique, nous avons travaillé à la création d'un groupe de recherche pluridisciplinaire et multi-sites en mathématiques appliquées et informatique. Il rassemble une quinzaine d'enseignants-chercheurs en Guadeloupe et Martinique, dans des domaines si divers (mais complémentaires) qu'il serait long de tous les énumérer. Je suis là encore persuadé que cette diversité sera la clef de notre succès. En tout cas, je le souhaite : celui qui nous a rassemblé en partie ici, en Martinique (Serge Lapiquonne), le mérite certainement plus que chacun d'entre nous, ne serait-ce que pour les nombreuses batailles qu'il a dû mener pour l'établissement de la recherche en sciences en Martinique.

Dans le contexte insulaire Martiniquais, la recherche revêt un tout autre rôle qu'en métropole. Notre université est en effet le seul établissement d'enseignement supérieur à proposer les services de chercheurs à ce point qualifiés, et elle n'a aucun concurrent dans le contexte des Antilles-Guyane. Ajouté à cela, nous sommes les tout premiers chercheurs en sciences dures en Martinique, et n' “existons” que depuis trois ans maintenant.
Ici sans doute plus qu'ailleurs, je suis persuadé que nous avons un rôle à jouer, dans notre *recherche*, pour faire avancer non seulement nos étudiants, mais aussi nos partenaires, qu'ils soient institutionnels ou privés. Des premiers contacts que j'ai eu localement avec des industriels, j'en ai acquis la conviction que la recherche appliquée peut (doit ?) se connecter avec le tissus économique local. Il y a des opportunités à prendre, que nous sommes probablement les seuls à pouvoir honorer, si nous savons conserver à notre recherche une envergure suffisante. J'ajouterai que certains industriels — locaux — ont peut-être de l'argent à gagner à nos cotés. Cela ne doit choquer personne : la monnaie qu'utilise le chercheur, c'est la publication, et elle est autrement plus décriée aujourd'hui par certains.
A ce sujet, j'ai pu mesurer avec Patrice Lefaucheur, mon thésard, à quel point la *recherche* en Data Mining peut être profitable *aussi* à des industriels, même de taille “modeste” (le

tissus économique de la Martinique est essentiellement composé de petites structures). Bien sûr, une telle activité représente une casquette de plus pour l'enseignant-chercheur, et le système ne nous encourage pas forcément encore à sortir de notre laboratoire douillet pour aller découvrir le monde. Personnellement, je me demande si je suis jamais entré complètement dans "mon" laboratoire. En réalité, je pense tout simplement ne pas avoir le profil. L'avenir me dira sans doute si j'ai raison, mais je compte bien développer mes préoccupations en Data Mining, et aussi en segmentation de séquences vidéo, non seulement car ce sont des domaines porteurs en recherche, mais aussi parce que certaines applications sont envisageable à l'échelle même de la Martinique.

Mes préoccupations théoriques ne me quitteront, elles non plus, pas de si tôt. Après tout, j'ai aussi choisi ce métier pour faire perdurer cette possibilité de pouvoir s'intéresser à des problèmes aussi difficiles et passionnants (et parfois, peut-être, bien inutiles) que l'apprenabilité de DNF dans le modèle PAC de Valiant. Répondre, surtout par la négative, à un tel problème, ne fera très probablement que confirmer ce que beaucoup (qui ont renoncé ?) pensent et écrivent dans de nombreux articles, mais c'est une si jolie question dans un si beau modèle... Ne pas l'aborder, c'est peut-être laisser passer un résultat tout aussi joli, et peut-être une preuve qui l'est d'avantage. Bien sûr certaines questions théoriques auxquelles je m'intéresse auraient certainement des implications en pratique : la création d'un algorithme de Boosting dans le modèle d'apprenabilité robuste est typiquement de celles-ci, et j'y travaille avec Patrice. Mais le désir d'établir un cheminement de la théorie vers la pratique, pour peu qu'on le veuille raisonnablement fondé, ne se fait pas facilement, et le temps nécessaire pour le parcourir s'inscrit sur une échelle bien longue, j'ai pu le tester en analyse d'images...

Après, il y a encore d'autres questions, d'autres problèmes, et des idées qui ne demanderaient sans doute qu'un peu de temps pour qu'on les couche sur un article. Même si l'énergie est facilement renouvelable, surtout sous ces latitudes clémentes, le temps, qui s'écoule de la même manière qu'ailleurs, vient parfois à manquer. Alors, bien sûr, j'ai toujours eu — et j'aurai toujours — la volonté de faire avancer mes travaux et ceux de mes collaborateurs, qu'ils s'approchent de la théorie de la complexité, des statistiques, probabilités, ou bien de tout autre domaine ; mais qui pourrait bien prétendre savoir où ils m'emmèneront, au fil de ce *long random walk* [Devroye et al., 1996], surtout quand je regarde où ils m'ont déjà emmené...
En définitive, ce mémoire d'habilitation n'est jamais qu'une borne "publimétrique" bien modeste le long de ce chemin dont, après tout, je n'espère en réalité rien d'autre en recherche que de nouvelles problématiques, de nouveaux challenges, de nouvelles collaborations, et de nouveaux résultats.

En somme, des ambitions là encore bien diverses.

—

Schoelcher (Martinique), Juillet 2002