Université Lumière Lyon II

Année 2002

THÈSE

pour obtenir le grade de

Docteur

en

Informatique

présentée et soutenue publiquement par

Fabrice Muhlenbach

le 16 Décembre 2002

Évaluation de la qualité de la représentation en fouille de données

préparée au sein du laboratoire ERIC
sous la direction de
Djamel A. Zighed
et Stéphane Lallich

COMPOSITION DU JURY

M. Jean-Pierre Barthélemy	Rapporteur	(Professeur, ENST-Bretagne)
M. Amedeo Napoli	Rapporteur	(Chargé de Recherche CNRS, LORIA Nancy)
M. Jean-François Marcotorchino	Examinateur	(Directeur de Recherche, Univ. MLV et Paris VI)
M. Gilbert Saporta	Examinateur	(Professeur, CNAM de Paris)
M. Djamel Abdelkader Zighed	Directeur de thèse	(Professeur, Université Lyon II)
M. Stéphane Lallich	Co-directeur de thèse	(Maître de Conférences, Université Lyon II)

Résumé

L'extraction des connaissances à partir de données (ECD) cherche à produire de nouvelles connaissances utilisables en tirant parti des grandes bases de données. Avant de procéder à la phase de fouille de données – étape phare de l'ECD – pour pouvoir opérer un apprentissage automatique, un ensemble de questions et de problèmes se posent : comment avoir a priori une idée de la manière dont les étiquettes de la variable à apprendre peuvent être séparées en fonction des variables prédictives ? comment traiter les bases pour lesquelles nous savons que des étiquettes sont fausses ? comment transformer des variables prédictives continues en variables discrètes en tenant compte globalement des informations de la variable à prédire ?

Nous proposons diverses réponses à ces problèmes. Ces solutions exploitent les propriétés d'outils géométriques : les graphes de voisinage. Le voisinage entre des individus projetés dans un espace à p dimensions nous fournit un moyen de caractériser la ressemblance entre les exemples à apprendre. À partir de ceci, nous élaborons un test statistique basé sur le poids des arêtes qu'il faut retirer dans un graphe de voisinage pour n'avoir que des sous-graphes d'une seule étiquette, ce qui nous informe a priori de la séparabilité des classes. Nous prolongeons ces réflexions dans le cadre de la détection d'individus dont l'étiquette est douteuse : nous proposons une stratégie de suppression et de réétiquetage des exemples douteux dans l'échantillon d'apprentissage afin d'augmenter la qualité des modèles prédictifs exploitant cet échantillon de données. Ces travaux sont étendus au cas particulier où la variable à prédire est numérique : nous présentons un test de structure pour la prédiction d'une telle variable. Enfin, nous proposons une méthode de discrétisation supervisée polythétique qui repose sur les graphes de voisinage et montrons ses performances en l'employant avec une méthode d'apprentissage supervisé que nous avons développée.

Remerciements

En tout premier lieu, je tiens à remercier le Professeur Djamel A. Zighed de l'Université Lumière – Lyon II pour m'avoir accueilli dans son laboratoire et avoir dirigé ce travail. J'ai rencontré Djamel en 1995 alors que je venais fraîchement de débarquer à Lyon après avoir quitté mon Alsace natale. Autant dire qu'il s'agit déjà d'une longue histoire. Djamel m'avait proposé de faire un stage au cours d'un été au sein de l'Équipe de Recherche en Ingénierie des Connaissances, entre ma licence et ma maîtrise de sciences cognitives, et ceci eut pour conséquence décisive de me donner le goût de la recherche. Qu'il soit ici mille fois remercié pour sa stimulation intellectuelle si communicative, pour son soutien indéfectible dans les moments de doute, et pour sa disponibilité malgré la lourdeur de ses charges administratives.

Merci aussi à Stéphane Lallich de l'Université Lumière – Lyon II qui a co-dirigé cette thèse. Au cours de ces années, j'ai appris à connaître et à apprécier Stéphane. Notre collaboration a été vraiment fructueuse, chacun apportant à l'autre les compétences issues de sa formation d'origine, ce qui nous a permis de confronter des points de vue complémentaires sur un grand nombre de situations.

Je remercie Amedeo Napoli – chargé de recherche CNRS au LORIA de Nancy, et responsable de l'équipe « Orpailleur » – et le Professeur Jean-Pierre Barthélemy – directeur du département « Intelligence Artificielle et Sciences Cognitives » de l'ENST-Bretagne – d'avoir rapporté cette thèse. La finesse de leurs critiques et la pertinence de leurs remarques montrent combien ils ont dû déployer de talent et d'énergie pour fouiller les pages de ce document afin d'en extraire la substantifique moelle.

Je remercie Jean-François Marcotorchino, directeur de recherche aux Universités de Marne-la-Vallée et de Paris VI, directeur scientifique de la société de *data* et *text mining* KALIMA et anciennement directeur scientifique d'IBM-France, ainsi que le Professeur Gilbert Saporta du Conservatoire Na-

tional des Arts et Métiers de Paris d'avoir accepté d'examiner ce travail. Merci encore à Gilbert Saporta de m'avoir fait l'honneur de présider mon jury de thèse.

Outre Djamel et Stéphane avec qui j'ai réalisé l'essentiel de mes travaux, je tiens à adresser mes remerciements aux personnes avec lesquelles j'ai collaborées à l'occasion de cette thèse, en particulier Sophie d'Hondt, Jean-Michel Jolion et Ricco Rakotomalala. Je souhaite également remercier mes collègues du laboratoire ERIC, notamment Radwan qui m'a aidé à me mettre à LaTeX sous l'environnement Windows, ma collègue de bureau Fadila pour sa bonne humeur, et notre indispensable secrétaire Valérie. Merci aussi à tous les autres que je ne citerais pas nommément de peur d'en oublier mais que j'ai côtoyé avec plaisir.

Merci à mes parents. Je leur serai éternellement reconnaissant de m'avoir assuré de leur confiance, de leurs encouragements et de leur soutien sans borne au cours des longues années de mon cursus universitaire, me laissant étudier aussi bien l'informatique que la psychologie ou les sciences cognitives, à Strasbourg, Lyon ou Paris. Merci aussi à mes frères Cédric et Cyril. L'amour bienveillant de ma famille, que j'ai trop peu vue ces derniers temps, représente une force précieuse qui m'a toujours permis d'affronter les aléas de la vie.

La thèse a occupé l'essentiel de mon temps au cours de ces trois dernières années. Cette période a pourtant été aussi celle de ma rencontre avec des personnes partageant un autre de mes centres d'intérêt que l'enseignement et la recherche : la lecture et l'écriture de science-fiction et autres littératures de l'imaginaire. Sans nos déjeuners au kébab du Tonkin, nos « pow-wow » S.-F., nos discussions électroniques, nos sorties aux festivals et conventions, mes études doctorales n'auraient pas eu cette saveur si particulière. Merci à mes amis de la *Gang* de m'avoir fait connaître le petit monde de la science-fiction francophone, de m'avoir fait lire des auteurs de nouvelles et romans passionnants, de m'avoir nourri en idées stimulantes, d'avoir supporté mes jeux de mots et mes occasionnelles sautes d'humeur, d'avoir servi de cobayes à mes gâteaux cuisinés au micro-onde, et surtout de m'avoir tellement apporté autant sur le plan de l'écriture que sur le plan humain. J'espère que ce document saura se montrer digne au niveau stylistique de leurs judicieux conseils.

Lyon, Fabrice Muhlenbach
le 16 décembre 2002

Table des matières

Table des figures

Liste des tableaux

Table des algorithmes

Introduction générale

L'informatique, qui se définit comme « la science et la technique du traitement automatique de l'information au moyen des ordinateurs », couvre aujourd'hui à peu près toutes les branches de l'activité humaine. Dès son origine, elle est marquée par le projet de concevoir une machine intelligente [Gan90]. Puisant ses sources dans la Grèce Antique où – selon la légende – des robots servaient le dieu ingénieur Héphaïstos et se nourrissant des réflexions sur le corps humain considéré comme une horloge au *siècle des Lumières* avec La Mettrie [dLM00], l'intelligence artificielle est un domaine qui se voit proposer un protocole expérimental dès le milieu du XX^{e} siècle grâce à Turing [Tur50], l'un des pères de l'informatique [Tur37].

L'intelligence artificielle (IA) a pour projet de simuler sur ordinateur des processus de la pensée ou des comportements qui, si on les rencontre, sont qualifiés d'intelligents : les comportements cognitifs, de prise de décision, mais aussi de perception.

Ayant trait au raisonnement symbolique et à la résolution de problèmes, l'IA se distingue des techniques numériques par son souci d'explicabilité. En effet, de manière générale, les systèmes typiques de l'IA sont transparents à leurs utilisateurs, à l'opposé d'une approche de type « boîte noire ».

Ce souci d'explicabilité replace aussi l'IA au sein de l'approche pluridisciplinaire des sciences cognitives. Selon l'approche cognitive, l'IA est la réalisation de programmes imitant dans leur fonctionnement l'esprit humain [AS93]. Dans ce champ de recherche, les développements récents de l'informatique et de l'intelligence artificielle apportent leurs lumières sur des problèmes classiques de psychologie et de philosophie de l'esprit. Les anciennes questions concernant le langage, le raisonnement ou notre manière de voir le monde trouvent une nouvelle forme d'investigation à travers les sciences de la modélisation. Pour Gardner [Gar93], l'entreprise cogniticienne se caractérise par la référence faite aux activités mentales, et en particu-

lier à l'utilisation des représentations, c'est-à-dire des connaissances ou des croyances existant dans la mémoire des individus. D'autre part, le cogniticien admet que l'ordinateur est le modèle le plus fiable du fonctionnement de l'esprit humain. Même si Gardner restreignait l'approche cogniticienne au seul humain en mettant l'accent sur les capacités langagières, il est plus généralement établi aujourd'hui que les sciences cognitives sont une tentative contemporaine de répondre scientifiquement à des questions épistémologiquement anciennes concernant la nature du savoir humain ou animal, ses composantes, ses sources, son développement et son utilisation, mais aussi des questions plus actuelles comme celles concernant le lien entre la cognition et l'émotion [KK95].

De par leur nature, les sciences cognitives ont à voir avec les connaissances. Nous entendons par « connaissance » l'idée exacte d'une réalité, de sa situation, de son sens, de ses caractères, de son fonctionnement. Les connaissances concernant un problème donné constituent un modèle de ce problème et de la situation du monde qui lui correspond car les problèmes ne peuvent être résolus que par l'intermédiaire de connaissances. Ainsi, les sciences cognitives peuvent être vues comme une approche pluridisciplinaire qui a pour projet de rendre compte des connaissances phénoménologiques (celles qui s'expriment en langue naturelle) en termes de connaissances scientifiques (celles qui sont prédictives, universelles et nécessaires). Ainsi, pour en revenir aux aspects plus concrets de l'intelligence artificielle, les connaissances utilisables peuvent être divisées en trois ensembles : des connaissances universelles et éternelles, des connaissances universelles et évolutives, des connaissances « typiques » (celles pour lesquelles il existe des contre-exemples), chacune de ces connaissances ayant éventuellement un mode de représentation plus adapté [Kay92].

En parallèle de cette approche, une autre communauté de chercheurs a tiré parti de l'ordinateur à travers sa grande rapidité de calcul et d'accès aux informations. L'outil informatique a permis de traiter de grands tableaux de données issues d'enquêtes ou d'expériences, et ceci sans effectuer d'a priori statistique sur ces derniers. Les résultats obtenus par cette manière de procéder étaient très encourageants mais celle-ci, n'étant pas très rigoureuse, s'est vivement fait critiquer sous les appellations "*data mining*" et "*data fishing*" par les statisticiens. Cependant, malgré les oppositions, cette démarche empirique a été popularisée, par exemple en France à travers les ouvrages de Benzécri sur l'analyse de données [Ben73] et a fini par remporter un véritable succès.

L'ordinateur, en plus de permettre de grandes capacités de traitements, a aussi permis de stocker de plus en plus de données. Vers la fin des années 1980, les spécialistes du domaine des bases de données ont cherché à exploiter le contenu de leurs bases. Les données, au lieu d'être considérées comme des archives inexploitables, sont alors devenues une masse d'informations où il était possible d'extraire des connaissances. Par exemple, en s'intéressant aux tickets de caisse des grandes surfaces, il a été possible de repérer des régularités dans les comportements des consommateurs [AIS93]. Les connaissances issues de ces « règles d'association » observées entre les différents produits présents dans les paniers des ménagères ont ainsi mieux défini les types de profils de la clientèle, ce qui a permis d'organiser de manière plus pertinente les rayonnages des magasins. Par conséquent, des simples données stockées dans l'ordinateur, le centre d'intérêt s'est reporté sur les connaissances que l'on pouvait découvrir dans celles-ci, ce qui fit apparaître les notions de « découverte » et « extraction » de connaissances (pour "*knowledge discovery*").

Définie par Fayyad, Piatetsky-Shapiro et Smyth en 1996 [FPSS96], « l'extraction des connaissances » est le procédé non trivial d'identification de connaissances valides, nouvelles, potentiellement utiles et compréhensibles à partir de données. Même si, dans la littérature, les termes « fouille de données » (pour "*data mining*") et « extraction des connaissances à partir de données » – ou ECD – (pour "*knowledge discovery in the databases*" – ou *KDD* –) sont parfois utilisés comme synonymes, nous réserverons l'expression « fouille de données » au seul procédé d'apprentissage automatique. Suivant la définition de Fayyad *et al.*, « l'ECD » désignera dans notre exposé l'ensemble du processus d'extraction des connaissances, à savoir le *pré-traitement des données* (mise en forme et nettoyage des données, traitement des données manquantes, sélection des variables, sélection des individus), la *fouille de données* en tant que telle (méthodes de description, de structuration et d'explication), et le *post-traitement* (validation des modèles, étude de leur intelligibilité).

L'approche de l'ECD est guidée par le souci de répondre à des problèmes concrets. Les modèles issus de cette approche permettent aux entreprises d'avoir une connaissance approfondie de leurs clients, de définir des profils de clientèle ou d'effectuer de la « gestion de relation client » ("*customer relationship management*"). À travers leurs fichiers de patients, les médecins peuvent ainsi bénéficier d'aide au diagnostic. Un autre exemple de domaine d'application où les données sont considérables est celui de la génomique : face à l'immensité des données du patrimoine génétique, il est nécessaire de disposer de méthodes automatisées capable d'extraire des informations per-

tinentes pour la compréhension du génome et espérer, à travers ces modèles, avoir une meilleure compréhension des mécanismes du vivant afin de voir apparaître des applications thérapeutiques aux maladies génétiques.

La thèse que je défends ici s'inscrit dans la double filiation de l'approche des *sciences cognitives* et de celle de l'*extraction des connaissances à partir de données*.

Le premier courant, qui fait de l'ordinateur un modèle de la cognition, doit son engagement théorique d'une longue tradition de recherche faisant collaborer autant les sciences exactes et naturelles (mathématiques, informatique, logique, neurosciences, physique) que les sciences humaines et sociales (psychologie, sciences du langage, philosophie). Or, même si nous n'allons pas nous interroger sur la possibilité de penser d'une machine ou si nous n'allons pas chercher à établir des méthodes et modèles en lien direct avec la cognition humaine, nous allons retirer de cette approche l'intérêt pour les notions de *représentation*, d'*apprentissage* et de *connaissance*.

Du second courant, issu de la rencontre du domaine des bases de données avec celui de la statistique, nous garderons l'esprit général de la démarche de l'ECD avec le souci de tirer parti des volumes de données afin de les valoriser à travers la constitution de modèles prédictifs.

Ainsi, nous rejoignons l'opinion de Kodratoff [Kod99] qui, identifiant six pôles dans l'ECD (les domaines des bases de données, de l'analyse de données, de l'apprentissage symbolique automatique, des réseaux de neurones, des statistiques et de la visualisation), considère que ces six domaines sont à la fois soumis à la pression des exigences techniques et ont certains liens avec les sciences cognitives. Selon lui, les liens qu'entretient l'ECD avec les sciences cognitives sont encore faibles, de manière assez paradoxale, alors qu'il serait souhaitable d'intensifier ces liens et de les unifier pour créer une coordination efficace entre ECD, applications et cognition.

Contributions de la thèse Nous tenons à préciser que nos contributions personnelles seront présentées dans les chapitres 3, 4 et 5, les deux premiers chapitres de cette thèse étant consacrés à un état de l'art sur notre domaine de recherche.

Nous débuterons ce document en situant la problématique générale de l'ECD. Nous nous attarderons sur le champ de l'apprentissage supervisé et présenterons les méthodes d'apprentissage à base d'exemples. De ce mode

d'apprentissage, nous poursuivrons par une exposition des graphes de voisinage, un outil développé dans le cadre de la géométrie computationnelle dont nous ferons amplement usage dans la suite de nos travaux.

Le classement dans les apprentissages à base d'exemples est réalisé par l'attribution à des éléments inconnus de l'étiquette connue d'individus qui leurs sont similaires. Nous préciserons dans le chapitre deux ces notions de similarité et de distance sur lesquelles se fondent aussi la construction des voisinages géométriques.

Le chapitre trois sera dédié à nos travaux sur la séparabilité des étiquettes. Nous indiquerons comment, à partir des graphes de voisinage, nous avons établi un test statistique capable de donner une mesure de la séparabilité des étiquettes dans un espace à p dimensions. Nous proposerons également une version locale de ce test afin de détecter les points « hors place », ou *outliers*, que nous traiterons pour les filtrer de notre base d'apprentissage ou pour leur attribuer une nouvelle étiquette.

Alors que les travaux présentés dans le chapitre trois concerneront plus spécifiquement l'apprentissage supervisé d'une variable à prédire lorsque celle-ci est catégorielle, le chapitre suivant s'intéressera à une généralisation de l'évaluation de la qualité de la représentation dans le cas de l'apprentissage d'une variable numérique. Nous proposerons une adaptation de notre test de séparabilité des étiquettes en utilisant les apports de l'analyse spatiale afin de détecter la présence de structure pour une variable à prédire numérique quand les exemples de la base d'apprentissage se distribuent dans l'espace de représentation.

Le dernier chapitre de ce document de thèse présentera une méthode de discrétisation supervisée polythétique réalisée à partir de la recherche d'amas d'une étiquette donnée obtenue avec un graphe de voisinage. Après nous être essentiellement intéressés aux aspects de pré-traitement dans l'approche de l'ECD, nous ferons un passage dans le domaine de la fouille de données afin d'exposer dans le chapitre cinq une méthode d'apprentissage supervisée procédant par généralisation. Nous indiquerons en outre en quoi cette méthode d'apprentissage symbolique est un complément judicieux de notre méthode de discrétisation polythétique.

Nous notons que les divers tests et méthodes que nous exposerons ont donné lieu à des développements logiciels qui sont venus enrichir une plateforme d'extraction des connaissances appelée « Sipina_W » et diffusée par le laboratoire ERIC de Lyon.

Apprentissage à base d'exemples et graphes de voisinage

Sommaire

Chapitre 1

Apprentissage à base d'exemples et graphes de voisinage

Résumé

Retraçant les grandes lignes de la problématique de l'extraction des connaissances et de l'analyse de données qui est à son origine, nous précisons ici quel est le vocabulaire employé dans ce document.

Les méthodes d'apprentissage supervisé sont très souvent fondées sur la recherche de modèles constitués d'informations générales extraites à partir des exemples vus lors de la phase d'apprentissage. Nous présentons dans ce chapitre une autre approche, l'apprentissage à base d'exemples, où la connaissance réside dans la façon de conserver certains individus de la base d'apprentissage et de procéder à la prédiction de nouvelles données suivant une similarité que celles-ci ont avec les exemples appris.

La similarité des données (selon les variables prédictives) étant associée au voisinage entre des individus représentés dans un espace multidimensionnel, nous exposons enfin les graphes de voisinage, des outils géométriques qui proposent différentes manières de considérer que deux exemples sont voisins ou non.

1.1 Introduction

1.1.1 Extraction des connaissances, apprentissage à base d'exemples et graphes de voisinage

Les sciences cognitives, nous l'avons vu dans la partie introductive, s'intéressent à la connaissance, voire sont même parfois définies comme étant les « sciences de la connaissance ». Or « connaître », c'est produire un modèle du phénomène et effectuer sur lui des manipulations réglées [Dup99]. Ce point de vue est aussi partagé par l'approche de l'extraction des connaissances à partir de données (ECD). La connaissance est l'objectif recherché par l'ECD, ainsi que son nom l'indique. Ces connaissances, nous les définissons comme étant les éléments constitutifs d'un modèle vis-à-vis d'un problème particulier.

Classiquement, les connaissances sont présentées sous la forme de règles du type :

« Si *Condition* Alors *Conséquence* »

c'est-à-dire que lorsque les prémisses d'une règle donnée sont satisfaites, sa conclusion en est déduite.

Toutefois, même si les connaissances sous forme de règles sont les plus aisément compréhensibles, il est possible de concevoir une connaissance en tant qu'*idée exacte que l'on peut avoir d'une réalité, de sa situation, de son sens, de ses caractères ou de son fonctionnement.* À ce titre, les méthodes de prédiction au sens large, les méthodes de structuration, voire les méthodes de visualisation, ont également à voir avec la connaissance. L'analyse de données [BS02] est par conséquent productrice de connaissances puisque ses méthodes apportent, respectivement, une connaissance sur une propriété des données qui est inconnue, elles indiquent la manière dont les données s'organisent entre elles, ou enfin elles font apparaître certains concepts non directement apparents à travers une nouvelle forme imagée.

La connaissance d'un certain type de problème peut ainsi, au lieu de s'exprimer sous forme de règles, se réduire à un modèle décrivant la façon dont des exemples déjà identifiés peuvent servir – à travers des notions de similarité – dans la prédiction de la valeur prise par des exemples inconnus.

Après avoir précisé certains éléments de vocabulaire ainsi que les notations que nous allons employer, nous traiterons dans ce chapitre des méthodes d'apprentissage supervisé. Parmi celles-ci, nous nous intéresserons plus particulièrement aux méthodes d'apprentissage à base d'exemples qui regroupent,

entre autre, l'algorithme du plus proche voisin. Cet algorithme procède au classement de données à prédire à travers une similitude observée entre ces dernières et les exemples appris. La notion de voisinage sur laquelle repose cette similitude sera ensuite étendue aux graphes de voisinage que nous présenterons dans la troisième section de ce chapitre.

1.1.2 Précisions méthodologiques et éléments de vocabulaire

Le document que nous présentons ici est issu de différentes disciplines, chacune porteuse de son propre vocabulaire. À cette complexité s'ajoute des expressions plus ou moins littéralement traduites des ouvrages et articles en langue anglaise. Nous débuterons donc ce chapitre essentiellement destiné aux graphes de voisinage et méthodes d'apprentissage à base d'exemples par une définition claire des termes que nous utiliserons tout en situant ceux-ci dans le cadre méthodologique du domaine de l'ECD.

L'extraction des connaissances à partir de données, suivant la manière dont elle situe son niveau d'analyse, regroupe trois grandes familles de méthodes :

1. les méthodes de *visualisation* et de *description* ;
2. les méthodes de *classification* et de *structuration* ;
3. les méthodes d'*explication* et de *prédiction*.

1.1.2.1 Méthodes de visualisation et de description

Ces méthodes, qui peuvent être uni-, bi- ou multidimensionnelles, ont pour objectif de fournir une compréhension synthétique de l'ensemble des données [ADZ00a]. Des objets peuvent être décrits par des espaces dépassant nos capacités d'entendement, aussi les méthodes de visualisation s'efforcent-elles de trouver une représentation permettant de rendre compte des concepts qui sous-tendent l'organisation des données.

Nous définissons dans cette sous-section les termes « visualisation », « description » et « représentation ».

Définition 1.1.1 Visualisation. *Perception visuelle de ce qui n'est pas normalement visible.*

Définition 1.1.2 Description. *Représentation effectuée au moyen de mots, de dessins ou de tracés géométriques.*

Définition 1.1.3 Représentation. *Fait de mettre quelque chose dans l'esprit et, plus concrètement, de rendre cette chose présente à la vue par une image, un signe ou un symbole.*

1.1.2.2 Méthodes de classification et de structuration

Ces méthodes regroupent les techniques d'apprentissage non supervisé et de classification automatique provenant des domaines de la reconnaissance de formes, de la statistique, de l'apprentissage automatique et du connexionnisme (comme les cartes auto-organisatrices de Kohonen [Koh97]). Au-delà de la première étape de description des données évoquée précédemment, il apparaît souvent souhaitable de chercher à identifier des groupes d'objets semblables, au sens d'une métrique donnée. Ces groupes peuvent correspondre à une certaine réalité ou à des concepts particuliers [ADZ00b].

Cette opération est dénommée :
- « classification » dans le vocabulaire de l'analyse de données ;
- « regroupement » (“*clustering*”) dans celui de l'intelligence artificielle ;
- « catégorisation » dans celui des sciences cognitives.

Les concepts que nous définissons pour la famille de ces méthodes sont la « classe », la « classification » et la « structuration ».

Définition 1.1.4 Classe. *Ensemble d'individus ou de choses qui possèdent des caractères communs.*

Définition 1.1.5 Classification. *Fait de construire des classes.*

Définition 1.1.6 Structuration. *Action de structurer, d'agencer, de disposer et d'organiser différents éléments d'un tout concret ou abstrait.*

1.1.2.3 Méthodes de prédiction et d'explication

Ces méthodes ont pour but de relier un phénomène à expliquer à un ou plusieurs phénomènes explicatifs. Issues des domaines de la statistique, de la reconnaissance de formes, de l'apprentissage automatique, du connexionnisme ou des bases de données (pour la recherche de règles d'associations), ces méthodes sont essentiellement mises en œuvre en vue d'extraire des modèles de classement ou de prédiction [ADZ00c]. Contrairement aux méthodes de structuration, les méthodes prédictives procèdent suivant le principe de l'apprentissage supervisé, appelé aussi « apprentissage avec professeur ». Elles

cherchent à indiquer quelle est l'étiquette d'un objet (la valeur de sa variable à prédire) en fonction des valeurs des variables prédictives.

Nous précisons ci-dessous ce que nous entendons par les notions de « classement », d'« explication » et de « prédiction ».

Définition 1.1.7 Classement. *Action de retrouver l'étiquette (à savoir, la classe déjà connue) d'un objet donné.*

Remarque 1.1.1 *Nous préférons utiliser l'expression « étiquette » ("label") à celle de « classe » pour diverses raisons. La première est que nous comptons réserver le mot « classe » aux méthodes de classification (non supervisée). Lorsqu'une classe est déjà construite, nous parlerons ainsi d'étiquette. La seconde raison découle de l'ambiguïté du terme « classe » ("class") qui, dans la littérature anglaise sur l'apprentissage automatique, désigne à la fois la variable à prédire (« la » classe) et les modalités prises par cette variable (« les » classes).*

Définition 1.1.8 Explication. *Développement destiné à faire comprendre la raison d'une chose.*

Définition 1.1.9 Prédiction. *Déclaration de ce qui doit arriver, fondée sur le raisonnement, l'induction scientifique. Ici, il s'agira plus particulièrement d'indiquer quelle est l'étiquette d'un individu inconnu à partir d'un modèle, rendant cette notion synonyme de « classement ».*

1.1.3 Apprentissage automatique : principe et notations

Nous décrivons ici le formalisme que nous allons employer pour parler de la problématique de l'apprentissage automatique, et plus particulièrement de l'apprentissage supervisé. Le formalisme présenté est adapté de Zighed et Rakotomalala [ZR00].

Soit Ω une population de n individus ou objets concernés par le problème d'apprentissage, que nous diviserons par la suite en deux ensembles Ω_a et Ω_t (avec $\Omega_a \cup \Omega_t = \Omega$).

À cette population Ω est associée une variable statistique particulière à expliquer notée Y, appelée aussi parfois « variable endogène » (ou encore, abusivement, la « variable classe », voire la « classe »).

À chaque individu ω de Ω peut être associée la valeur e de la variable à prédire $Y(\omega)$, c'est-à-dire son étiquette.

La détermination du modèle de prédiction φ est liée à l'hypothèse selon laquelle les valeurs de la variable statistique Y ne relèvent pas du hasard mais de situations qu'il est possible de caractériser. À cet effet, un expert du domaine concerné établit une liste a priori de variables statistiques appelées « variables exogènes » ou « variables prédictives » et notées $\mathbf{X} = (X_1, X_2, ..., X_p)$.

L'objectif est de rechercher un modèle de prédiction φ permettant, pour un individu ω issu de Ω pour lequel la valeur de la variable à expliquer $Y(\omega)$ est inconnue mais dont l'état des variables prédictives $\mathbf{X}(\omega)$ est connu, de prédire cette valeur notée $\widehat{Y}(\omega)$.

L'apprentissage supervisé se propose donc de fournir des outils permettant d'extraire, à partir de l'information disponible sur un échantillon d'apprentissage noté Ω_a, le modèle de prédiction φ.

L'apprentissage aura atteint son objectif s'il parvient à produire un modèle valide. La validation du modèle sera estimée après vérification sur un échantillon test Ω_t (pour lequel l'étiquette est connue pour chacun des individus de cet échantillon) que le modèle de prédiction φ donne bien la valeur de la variable à prédire attendue.

1.2 Apprentissage à base d'exemples

1.2.1 Introduction

L'*apprentissage à base d'exemples*, appelé également apprentissage à base de cas ou d'« instances » – traduction de *instance-based learning* (*IBL*) – produit un classement à travers l'usage d'exemples spécifiques. Ce mode d'apprentissage automatique supervisé n'est pas dénué de fondement d'un point de vue cognitif. En effet, au niveau psychologique, une forme d'apprentissage vraiment très simple consiste à mémoriser l'ensemble (supposé quand même assez restreint) des éléments disponibles pour lesquels les valeurs de tout un groupe de paramètres sont connues. À partir de ces informations connues gardés en mémoire, il devient possible d'attribuer à de nouveaux objets pour lesquels une propriété donnée n'est pas connue (ici, il s'agit de son étiquette) la même valeur que celle dont ils sont « proches » parmi les exemples déjà vus. Nous allons à présent traduire sous forme plus détaillée la démarche de ce genre d'apprentissage dont nous venons de tracer les traits grossiers.

Les méthodes d'apprentissage à base d'exemples procèdent par la conser-

vation des exemples caractéristiques des variables prédictives pour chacune des étiquettes à apprendre. Pour Aha, Kibler et Albert [AKA91], il s'agit des méthodes dérivées du plus proche voisin. Pour Mitchell [Mit97], les méthodes d'apprentissage à base d'exemples regroupent aussi bien les k-plus proches voisins que l'algorithme des plus proches voisins pondérés par la distance, la régression pondérée de manière locale, les fonctions à base radiale ou le raisonnement à partir de cas. Dans un premier temps, nous suivrons le point de vue de Aha *et al.* qui ont défini le cadre général et la méthodologie des algorithmes d'apprentissage à base d'exemples en 1991.

À la différence de la plupart des algorithmes d'apprentissage, les approches à base d'exemples ne construisent pas une hypothèse abstraite mais effectuent un classement des exemples de test à partir de la similarité que ces derniers ont avec des exemples d'apprentissage. La phase d'apprentissage est donc particulièrement simple puisqu'elle se réduit au seul stockage des exemples d'apprentissage. Les principaux traitements ne sont ainsi réalisés qu'en phase de généralisation, les exemples du modèle sont sollicités au moment où un nouvel exemple a besoin d'être prédit. Les méthodes à base d'exemples sont ainsi parfois appelées les systèmes d'apprentissage « paresseux » (*"lazy" learners*) à la différence des systèmes d'apprentissage « avides » (*"eager" learners*) tels que les arbres de décision [Aha97].

Au cours de l'apprentissage à base d'exemples, les objets constituant le modèle sont des points projetés dans un espace multidimensionnel. Étant donné un nouvel exemple, sa relation avec les exemples stockés est examinée selon la valeur d'une fonction cible de ce nouvel exemple. Dans le pire des cas, la vérification nécessite de comparer l'exemple test à chacun des exemples d'apprentissage.

1.2.2 Apprentissage à base d'exemples et reconnaissance des formes

Nous faisons remarquer que les travaux que nous exposons dans cette section se situent dans l'approche de l'apprentissage automatique supervisé et de l'extraction des connaissances. De ce fait, nous n'évoquerons pas réellement les contributions apportées dans le cadre la reconnaissance des formes (RdF).

En effet, les méthodes à base de voisinage ont été largement exploitées et développées dans le domaine de la RdF. L'approche statistique, un des modèles de ce domaine, a ainsi été le terrain d'un grand nombre de travaux précurseurs des apprentissages à base d'exemples comme ceux de Devijver

et Kittler [DK82] ou Duda et Hart [DHS00] qui ont par exemple développé la technique des k-plus proches voisins. Les travaux comme ceux de Parzen [Par62], au lieu de s'intéresser aux k voisins les plus proches pour l'attribution d'une étiquette, considèrent une région de voisinage (« fenêtre de Parzen »).

Jain, Duin et Mao [JDM00] situent le domaine de la fouille de données parmi les applications de la RdF (tout comme peuvent l'être par exemple la bioinformatique ou la reconnaissance de la parole). Pour eux, la fouille de données est une application de RdF qui se caractérise par son objet (la recherche de motifs signifiants), ses motifs d'entrée (des points dispersés dans un espace multidimensionnel) et ses motifs recherchés (des groupes de données compacts et bien séparables).

Notre point de vue est toutefois un peu différent. Pour nous, la fouille de données n'est pas qu'une application de la RdF parmi d'autres. Nous considérons que la fouille de données, même si elle emploie les algorithmes de la reconnaissance des formes [DK82] ou de l'apprentissage automatique [Mit97], est une étape du processus global de l'ECD [FPSS96]. L'ECD met l'accent sur la découverte de connaissances et de nouvelles structures dans les données, or si nous présentons les apprentissages à base d'exemples et divers moyens de considérer que des individus sont voisins ou non, c'est justement parce que ces derniers nous serviront à exprimer des structures et connaissances sur les données.

1.2.3 Propriétés

Aha, Kibler et Albert [AKA91] définissent trois propriétés des algorithmes d'apprentissage à base d'exemples :

1. **Une fonction de similarité**. Cette fonction décrit à l'algorithme la manière dont deux exemples sont proches. Bien que ceci semble trivial, le choix de cette fonction de similarité est particulièrement complexe, en particulier dans les situations où certaines variables sont catégorielles. Nous aborderons ce problème de manière plus approfondie dans le chapitre 2.
2. **Une fonction de sélection des exemples typiques**. Cette fonction décrit à l'algorithme quels sont les exemples qui doivent être gardés en tant que prototypes.
3. **Une fonction de classement**. Cette fonction est celle qui détermine, lorsqu'un nouvel individu est présenté, de quelle manière il est lié aux individus appris.

Aha *et al.* notent également deux autres points caractéristiques de cette approche. Tout d'abord, les algorithmes à base d'exemples supposent que les exemples similaires du point de vue des valeurs des variables prédictives ont aussi des étiquettes similaires (cet élément, qui a trait à la qualité de la représentation, fera l'objet des chapitres 3 et 4). Ensuite, les algorithmes à base d'exemples font l'hypothèse que toutes les variables ont une pertinence équivalente, et donc un poids équivalent dans la décision de classement. De ce fait, pour neutraliser l'effet des différences de grandeurs entre les variables prédictives, Aha *et al.* suggèrent que ces variables soient centrées et réduites.

1.2.4 Limites et améliorations

Un ensemble de limitations découlent des propriétés énoncées ci-dessus. Breiman, Friedman, Olshen et Stone, en 1984 [BFOS84], ont décrit six problèmes dérivés de l'algorithme du plus proche voisin :

1. les algorithmes de type plus proche voisin ont un temps de calcul coûteux en phase de généralisation puisqu'ils sauvegardent tous les exemples de la phase d'entraînement ;
2. ils sont sensibles au bruit sur les variables prédictives ;
3. ils sont sensibles aux variables prédictives non pertinentes ;
4. ils sont sensibles au choix de la fonction de similarité de l'algorithme ;
5. ils ne peuvent pas traiter de manière naturelle les variables prédictives catégorielles et les données manquantes ;
6. ils ne fournissent que peu d'information utilisable concernant la structure des données.

Aha *et al.* ont proposé les algorithmes *IB1*, *IB2* et *IB3* [AKA91] (que nous appellerons globalement "*IBL*") pour montrer comment certaines améliorations permettent de pallier les divers problèmes communs aux apprentissages à base d'exemples évoqués ci-dessus. Nous rappellerons brièvement le principe de ces algorithmes car ils expriment de manière assez claire les points forts et les points faibles de l'apprentissage à base d'exemples.

IB1 est sans doute l'un des algorithmes d'apprentissage à base d'exemples les plus simples qui soit (*cf.* algorithme 1). Il repose sur une fonction de similarité entre les exemples (décrite en 1.2.1) constituée par l'opposé de la racine carrée d'une somme de différences existant entre chacune des valeurs des exemples sur l'ensemble des p variables prédictives.

$$\text{similarité}(\alpha, \beta) = -\sqrt{\sum_{i=1}^{p} f(\alpha_i, \beta_i)} \tag{1.2.1}$$

Cet algorithme peut traiter aussi bien des variables prédictives numériques que catégorielles. Dans le cas de variables numériques, celles-ci sont centrées et réduites, et la fonction f est définie en 1.2.2.

$$f(\alpha_i, \beta_i) = (\alpha_i - \beta_i)^2 \tag{1.2.2}$$

Dans le cas de variables catégorielles ou booléennes, la fonction f est égale à l'expression décrite en 1.2.3 avec la fonction $T(u)$ retournant la valeur numérique 1 dans le cas où l'expression booléenne u est vraie et 0 dans le cas où u est fausse.

$$f(\alpha_i, \beta_i) = T(\alpha_i \neq \beta_i) \tag{1.2.3}$$

Dans le cas où des valeurs de variables prédictives sont manquantes, la différence est supposée maximale, comme décrite en 1.2.4.

$$f(\alpha_i, \beta_i) = 1 \tag{1.2.4}$$

Par conséquent, *IB1* est très proche de l'algorithme du plus proche voisin et n'en diffère que par la normalisation des variables numériques, son traitement incrémental des exemples et sa manière de traiter les valeurs manquantes.

La description du concept C de l'algorithme *IB1* conserve l'ensemble des exemples, et l'indicateur booléen « bon_classement » indique à chaque fois qu'un exemple $\omega_i \in \Omega_a$ est ajouté à C si celui-ci est bien prédit par le modèle en construction. Le tableau « Sim » conserve de manière temporaire la similarité entre un exemple β donné de C et le nouvel exemple de Ω_a.

La procédure de classement est décrite par l'algorithme 2. Aha *et al.* indiquent que de bonnes performances de prédiction sont observées pour ce type de modèle, performances qui sont d'autant meilleures que le nombre d'exemples présents dans C est élevé. Cependant ce type d'algorithme est très lourd car il conserve l'intégralité des exemples présents dans la base d'apprentissage Ω_a.

L'algorithme *IB2*, contrairement à *IB1*, ne conserve que les exemples qui, dans la phase d'apprentissage, permettent d'améliorer le classement (*cf.* algorithme 3). Ainsi, dans *IB2*, au lieu d'emmagasiner tous les exemples, seuls

Algorithme 1 *IB1*

$C \leftarrow \varnothing$ {au départ, la description du concept est un ensemble vide}
pour tout $\alpha \in \Omega_a$ **faire**
  pour tout $\beta \in C$ **faire**
    Sim[β] $\leftarrow$ similarité(α, β)
  fin pour
  $\beta_{max} \leftarrow \beta \in C$ où Sim[β] est maximal
  si $Y(\alpha) = Y(\beta_{max})$ **alors**
    bon_classement $\leftarrow$ **vrai**
  sinon
    bon_classement $\leftarrow$ **faux**
  fin si
  $C \leftarrow C \cup \{\alpha\}$
fin pour

Algorithme 2 Classement pour les algorithmes IBL

pour tout $\alpha \in \Omega_t$ **faire**
  pour tout $\beta \in C$ **faire**
    Sim[β] $\leftarrow$ similarité(α, β)
  fin pour
  $\beta_{max} \leftarrow \beta \in C$ où Sim[β] est maximal
  $\hat{Y}(\alpha) \leftarrow Y(\beta_{max})$
fin pour

sont conservés ceux qui apportent une nouvelle information au modèle, c'est-à-dire ceux pour lesquels l'indicateur « bon_classement » est faux. De la sorte, le nombre d'exemples stockés pour le modèle est fortement réduit.

Algorithme 3 *IB2*

$C \leftarrow \varnothing$
pour tout $\alpha \in \Omega_a$ **faire**
  pour tout $\beta \in C$ **faire**
    $\mathrm{Sim}[\beta] \leftarrow$ similarité(α, β)
  fin pour
  $\beta_{max} \leftarrow \beta \in C$ où $\mathrm{Sim}[\beta]$ est maximal
  si $Y(\alpha) = Y(\beta_{max})$ **alors**
    bon_classement $\leftarrow$ **vrai**
  sinon
    bon_classement $\leftarrow$ **faux**
    $C \leftarrow C \cup \{\alpha\}$
  fin si
fin pour

Dans le cas où les différentes valeurs de la variables à prédire sont bien séparées dans l'espace de représentation $\mathbb{R}^p$, les performances de l'algorithme *IB2* sont comparables à *IB1* avec un modèle bien plus simple que celui-ci.

Toutefois, dans le cas où la base d'apprentissage comporte du bruit ou lorsque cette base contient de nombreuses exceptions, l'algorithme *IB2* conserve à tort les mauvais exemples. Ceci a pour effet de perturber la description du concept C, les étiquettes de ces mauvais exemples sont utilisées pour prédire celles des exemples de test $\omega_i \in \Omega_t$, ce qui a pour effet de diminuer les performances en généralisation du modèle.

Avec *IB3* (*cf.* algorithme 4), ce problème est en partie résolu à travers l'introduction de deux nouveaux éléments :

- pour chacun des exemples sauvegardés dans le modèle C, un résultat de classement est conservé ;
- un test de significativité est utilisé pour déterminer quels sont les exemples qui présentent de bonnes propriétés de prédiction et quels sont ceux supposés être des données buitées.

Le résultat de classement est un score qui dépend du nombre de prédictions correctes et incorrectes effectuées pour chacun des exemples sauvegardés dans le modèle C. En résumant la performance de prédiction d'un exemple donné de C sur les exemples d'apprentissage présentés par la suite,

Algorithme 4 *IB3*

$C \leftarrow \varnothing$
pour tout $\alpha \in \Omega_a$ **faire**
  pour tout $\beta \in C$ **faire**
    Sim$[\beta] \leftarrow$ similarité(α, β)
  fin pour
  si $\exists\{\beta \in C \mid \text{acceptable}(\beta)\}$ **alors**
    $\beta_{max} \leftarrow$ acceptable$(\beta) \in C$ où Sim$[\beta]$ est maximal
  sinon
    $i \leftarrow$ un indice au hasard compris entre 1 et $Card(C)$
    $\beta_{max} \leftarrow \beta \in C$ où similarité(i^{e} exemple, α) est maximale
  fin si
  si $Y(\alpha) = Y(\beta_{max})$ **alors**
    bon_classement $\leftarrow$ **vrai**
  sinon
    bon_classement $\leftarrow$ **faux**
    $C \leftarrow C \cup \{\alpha\}$
  fin si
  pour tout $\beta \in C$ **faire**
    si Sim$[\beta] \geqslant$ Sim$[\beta_{max}]$ **alors**
      mettre à jour le résultat de classement de β
      si le résultat de classement de β est significativement faible **alors**
        $C \leftarrow C - \{\beta\}$
      fin si
    fin si
  fin pour
fin pour

cette information suggère ainsi comment cet exemple peut se comporter à l'avenir en phase de généralisation. Quant au test de significativité, il permet de déterminer quels sont les exemples qui ont de bonnes propriétés de classement – et qui, de ce fait, seront conservés dans C – ainsi que ceux qui sont supposés apporter du bruit – et qui seront retirés de l'ensemble C. Le test de significativité permet ainsi de réaliser un filtrage incrémental des exemples conservés dans le modèle.

Les tests réalisés par Aha, Kibler et Albert indiquent que l'algorithme *IB3* produit un modèle constitué d'un nombre réduit d'exemples, tout comme *IB2* mais qui, tout comme *IB1*, dispose de bonnes capacités de généralisation et de résistance au bruit.

Ces auteurs notent cependant que les performances de l'algorithme d'apprentissage *IB3* (ainsi que celles des algorithmes *IB1* et *IB2*) sont quand même très sensibles au nombre de variables prédictives non pertinentes introduites pour décrire les exemples. En effet, ce genre de modèle suppose un poids équivalent des différentes variables prédictives, par conséquent des variables prédictives non pertinentes vont, autant que celles qui sont pertinentes, participer à la modification de l'espace de représentation dans lequel vont se projeter les données, changeant les distances relatives entre les différents points. Une phase de traitement préalable est ainsi souhaitée au cours de laquelle les variables non pertinentes sont repérées et retirées du modèle afin de diminuer l'espace de représentation et de le limiter aux seules variables qui apportent une réelle information utilisable pour la prédiction des diverses étiquettes.

1.2.5 Autres méthodes d'apprentissage à base d'exemples

1.2.5.1 Apprentissage par les k-plus proches voisins

L'algorithme des *k-plus proches voisins* de Fix et Hodges [FH51, FH52] ou Cover et Hart [CH67] est une généralisation de l'apprentissage par le plus proche voisin ou de l'algorithme *IB1* vu précédemment (*cf.* algorithme 5).

Algorithme 5 Apprentissage par les k-plus proches voisins

$C \leftarrow \varnothing$
pour tout $\alpha \in \Omega_a$ **faire**
  $C \leftarrow C \cup \{\alpha\}$
fin pour

Au lieu de retourner comme valeur d'étiquette celle du point le plus

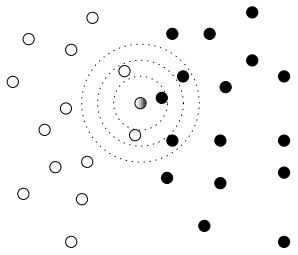

FIG. 1.1 – k-plus proches voisins avec deux étiquettes : noir et blanc

similaire au point non connu, l'algorithme des k-plus proches voisins retourne comme étiquette celle qui est majoritaire parmi les k points les plus proches de celui dont on cherche à prédire l'étiquette (*cf.* algorithme 6).

Algorithme 6 Classement par les k-plus proches voisins

pour tout $\alpha \in \Omega_t$ **faire**
soient $\beta_1, \beta_2, ..., \beta_k \in C$ les k exemples pour lesquels la similarité(α, β_i) est maximale
$\widehat{Y}(\alpha) \leftarrow \text{argmax}_{i=1}^{k}(Y(\beta_i))$ {$\text{argmax}_{i=1}^{n}(f(x_i))$ renvoie la valeur de $f(x)$ la plus fréquente parmi les n exemples}
fin pour

L'emploi de k voisins au lieu d'un seul assure une plus grande robustesse à la prédiction. Classiquement, ce paramètre k est une valeur impaire ($k = 3$ ou $k = 5$) afin d'avoir une majorité plus facilement décidable (en évitant les *ex æquo*) dans le cas relativement fréquent des problèmes d'apprentissage où la variable à prédire comporte deux étiquettes. Toutefois, la valeur de k risque de changer les performances du modèle, comme cela est présenté en figure 1.1 : pour $k = 1$, l'exemple à prédire (au centre) prend l'étiquette « noir » ; pour $k = 3$, l'exemple à prédire prend l'étiquette « blanc » ; pour $k = 5$, l'exemple à prédire prend l'étiquette « noir ». En outre, il existe un autre problème avec ce type d'apprentissage : rien ne justifie a priori qu'une valeur de k puisse être privilégiée à une autre.

Nous faisons remarquer que dans le cas où l'étiquette est numérique et non catégorielle, l'apprentissage par les k-plus proches voisins peut aisément être appliqué. Dans ce cas, la valeur numérique retournée est la moyenne des k points les plus proches de l'exemple à prédire : $\widehat{Y}(\alpha) \leftarrow \frac{\sum_{i=1}^{k} Y(\beta_i)}{k}$

1.2.5.2 Apprentissage par les k-plus proches voisins pondérés par la distance

Une extension simple de cette approche est l'algorithme des *k-plus proches voisins pondérés par la distance* [Dud76]. Dans ce cas, au lieu d'une moyenne simple, l'algorithme utilise une moyenne pondérée par une fonction de la distance entre l'individu à prédire et ses k-plus proches voisins. En général, ce poids est une fonction inverse de la distance.

L'introduction de cette pondération apporte une résistance au bruit encore meilleure que la méthode des k-plus proches voisins.

Ce mode d'apprentissage, qui s'applique plus particulièrement à la prédiction d'une étiquette numérique, peut toutefois être adapté à la prédiction d'une étiquette catégorielle. Pour cela, le classement par les k-plus proches voisins pondérés par la distance est réalisé de la manière suivante : la distance est utilisée pour établir un vote pondéré sur l'ensemble des k voisins.

1.2.5.3 Apprentissages associés aux modèles de régression

Mitchell [Mit97] étend la notion d'apprentissage à base d'exemples à toute méthode d'apprentissage qui ne procède pas par la recherche de caractéristiques présentes parmi les exemples vus en phase d'apprentissage mais conserve l'ensemble de ces derniers pour réaliser la prédiction. De ce fait, il ajoute aux méthodes présentées précédemment d'autres systèmes prédictifs dont certains se rapprochent des modèles de régression.

Dans cette problématique, la variable à prédire est numérique, ce qui est un peu moins commun en apprentissage supervisé. Nous décrirons cependant brièvement l'idée de base de la *régression pondérée localement* et des *fonctions à base radiale*.

Étant donné que l'apprentissage par les k-plus proches voisins consiste en une approximation locale de la variable à prédire Y pour chacun des points ω, pourquoi ne pas réaliser une approximation de $Y(\omega)$ pour les régions environnant chaque point ω ?

Dans le cas de la *régression pondérée localement*, il s'agit ainsi d'adapter une fonction linéaire ou quadratique aux k-plus proches voisins.

Dans le cas des *fonctions à base radiale*, l'approximation globale de Y se fait en terme de combinaisons linéaires d'approximations locales. L'apprentissage des fonctions à base radiale, bien que très lié à la régression pondérée par la distance, s'en diffère par l'ajout d'unités cachées qui modifient leur environnement suivant une loi normale. Par ailleurs, les fonctions à base

radiale, contrairement aux méthodes précédentes, font partie des systèmes d'apprentissage « avide » et non « paresseux ».

1.2.5.4 Raisonnement à partir de cas

Mitchell [Mit97] ajoute aussi aux apprentissages à base d'exemples le *raisonnement à partir de cas*, un domaine de recherche particulièrement étudié en intelligence artificielle en raison de sa grande proximité avec le fonctionnement cognitif humain.

Il s'agit d'un système de résolution de problèmes se fondant sur la réutilisation par analogie d'expériences passées au cours d'un cycle de raisonnement. Les données, pour cette méthode d'apprentissage à base d'exemples, au lieu d'être des points projetés dans un espace multidimensionnel, sont représentées sous forme symbolique. Le raisonnement à partir de cas procède, comme les autres apprentissages à base d'exemples, par le stockage des exemples déjà rencontrés. Dans le vocabulaire du raisonnement à partir de cas, on parle d'expériences conservées en mémoire. Ces expériences sont utilisées pour résoudre de nouvelles situations à travers :

- la recherche de l'expérience ayant une situation similaire au nouveau cas ;
- la réutilisation de cette expérience, totale ou adaptée, pour résoudre le nouveau cas ;
- l'apprentissage par l'éventuel ajout de cette nouvelle expérience à la mémoire.

Un cas est ainsi une expérience passée permettant au système de résoudre des problèmes plus efficacement ou d'éviter des échecs. Les difficultés rencontrées dans cette approche concernent, d'une part, la représentation d'un cas et son organisation en mémoire et, d'autre part, la sélection du cas et la mesure de similarité entre les différents cas qui sous-tendent cette sélection.

La mesure des ressemblances entre les cas, qui est fondamentale car elle permet d'indexer la base de cas, reste – avec la phase d'adaptation de cas – un des plus importants problèmes du raisonnement à partir de cas. Par conséquent, de nouvelles métriques de distance et similarité sont nécessaires afin de quantifier le degré d'appariement entre deux cas.

1.2.6 Bilan de l'apprentissage à base d'exemples

À travers la présentation des méthodes d'apprentissage à base d'exemples les plus classiques que nous venons d'effectuer, nous pouvons retirer un trait

caractéristique de ces méthodes : leur simplicité. Pour les algorithmes *IBL* d'Aha, Kibler et Albert, nous avons vu que la phase d'apprentissage pouvait être plus ou moins raffinée, demandant par exemple l'introduction d'un test de significativité pour ne conserver dans le modèle que les exemples les plus pertinents pour le classement. Nous avons également remarqué que la phase de classement pouvait aussi être plus ou moins simple, avec l'introduction d'un paramètre k à travers l'apprentissage par les k-plus proches voisins. Toutefois, le choix de ce paramètre k n'est pas anodin : dans l'exemple présenté en figure 1.1, nous avons vu que les prédictions effectuées pouvaient beaucoup changer suivant le choix de la valeur k, le nombre de voisins.

Nous pouvons aussi ajouter que, malgré leurs bonnes capacités prédictives et la simplicité de leurs phases d'apprentissage, les méthodes d'apprentissage à base d'exemples sont toutefois limitées en tant que modèle prédictif. En effet, le seul stockage des données d'apprentissage, stockage plus ou moins élaboré suivant les cas, ne fournit pas une information synthétique sur les données. Par conséquent, les modèles d'apprentissage à base d'exemples apparaissent moins pertinents que d'autres approches qui fournissent des modèles plus explicites d'un point de vue cognitif.

En résumé, les méthodes d'apprentissage à base d'exemples ont comme avantages de proposer une phase d'apprentissage très rapide et de pouvoir apprendre des fonctions de classement très complexes. Cependant elles présentent deux inconvénients majeurs : leur phase de généralisation est lente, d'une part, et, d'autre part, elles nécessitent que l'espace de représentation fasse ressortir une certaine organisation de la variable à prédire, aussi ces méthodes peuvent-elles assez facilement être perturbées par la présence de variables prédictives non pertinentes.

Le premier problème peut en partie être réglé à travers l'emploi d'arbres de recherche multidimensionnels ("*k-dimensional binary trees*" ou plus simplement "*k-d trees*"). Les arbres de recherche multidimensionnels représentent des bissections successives d'un espace de dimension k construites à partir de la densité observée, ce qui permet une structuration de l'espace de recherche. De la sorte, la complexité du calcul nécessaire à l'interrogation du modèle, c'est-à-dire la recherche incrémentale des meilleurs candidats à l'appariement, peut être réduite [WAD94].

Au sujet du problème des variables prédictives non pertinentes, de nombreuses approches existent, regroupées dans ce qui est appelé dans le monde anglophone sous le terme de "*feature selection*". Tout d'abord, certaines méthodes d'apprentissage intègrent directement un système de sélection des variables (comme c'est le cas pour la plupart des arbres de décision) et ne

sont donc pas sensibles aux variables non pertinentes. Certaines méthodes emploient un système de filtrage des variables (par exemple reposant sur des mesures de corrélation de chaque variable prédictive avec la variable à prédire), et procèdent soit par ajout progressif de variables prédictives pertinentes ("*forward selection*") soit par suppression de variables prédictives non pertinentes ("*backward elimination*"). D'autres méthodes procèdent suivant une approche « enveloppante » ("*wrapper approach*") en se basant sur l'algorithme d'induction en tant que fonction d'évaluation [JKP94], par exemple l'algorithme du plus proche voisin [AB95]. Enfin, il existe des méthodes de pondération des variables prédictives, pondération qui est effectuée suivant le degré de pertinence perçue de chaque variable, et ce procédé permet de moduler l'influence relative des variables prédictives les unes par rapport aux autres [BL97].

Pour conclure, nous indiquerons que, classiquement, les méthodes d'apprentissage à base d'exemples ne sont pas appropriées pour tous les problèmes d'apprentissage existants. Elles sont généralement bien adaptées dans le cas où le nombre de données d'apprentissage $n_a = Card(\Omega_a)$ est important et quand les variables prédictives sont peu nombreuses (par exemple $p < 20$).

1.3 Graphes de voisinage

1.3.1 Introduction

Nous avons décrit, dans la section précédente, les méthodes d'apprentissage à base d'exemples. Dans cette section, nous allons exposer le principe des graphes de voisinage, des outils issus de la géométrie computationnelle [PS88]. Par de nombreux points, les graphes de voisinage sont liés aux algorithmes d'apprentissage à base d'exemples. Avant de présenter de tels graphes, nous indiquerons un ensemble de définitions permettant de préciser les concepts que nous allons employer. De nombreux graphes de voisinage existent mais notre présentation se limitera à quatre graphes différents qui ont pour propriétés d'être symétriques et connexes, aussi écarterons-nous l'algorithme des k-plus proches voisins et ses diverses variantes.

1.3.2 Définitions

Nous reprenons ici le formalisme employé par Barthélemy et Guénoche en 1988 [BG88] et Sebban en 1996 [Seb96].

Définition 1.3.1 Graphe, sommet, arêtes. *Un graphe permet de représenter l'existence ou l'absence de liaisons entre des objets. Un graphe G est formé d'un ensemble de sommets notés Σ reliés entre eux par un ensemble d'arêtes (arcs non orientés) noté A. Nous indiquerons que le graphe G est décrit par le couple (Σ, A).*

Remarque 1.3.1 *Cette formulation est employée pour respecter l'usage en vigueur en théorie des graphes. Toutefois nous gardons à l'esprit que les différents sommets de l'ensemble Σ représentent les différents points projetés dans l'espace $\mathbb{R}^p$ correspondant à nos n individus de la population Ω. Ainsi, sauf indication contraire, le nombre de sommets de l'ensemble Σ sera égal à n (ou n_a), le nombre d'individus de la population Ω (ou Ω_a).*

Remarque 1.3.2 *Le nombre d'arêtes a de l'ensemble A reliant les différents sommets du graphe G sera au plus égal à la combinaison de tous les sommets du graphe :*

$$a \leqslant \frac{n(n-1)}{2} \tag{1.3.1}$$

Définition 1.3.2 Graphe valué, longueur, valuation. *Un graphe valué est un couple (G, L) formé d'un graphe $G = (\Sigma, A)$ et d'une fonction L, à valeurs réelles strictement positives, définies sur A. $L(\alpha, \beta)$ est appelée la longueur ou la valuation de l'arête (α, β).*

Remarque 1.3.3 *La valuation de l'arête (α, β) sera par exemple la distance euclidienne, notée $\delta(\alpha, \beta)$, séparant les points α et β (cf. équation 1.3.2).*

$$\delta(\alpha, \beta) = \sqrt{\sum_{i=1}^{p} f(\alpha_i, \beta i)} \tag{1.3.2}$$

où $f(\alpha_i, \beta_i) = (\alpha_i - \beta_i)^2$ dans le cas où les variables prédictives X_i sont numériques. Dans le cas où X_i est une variable prédictive booléenne ou catégorielle, f vaudra 1 si $\alpha_i = \beta_i$ et 0 sinon (cf. expression 1.2.3). Nous aurons donc une formule de calcul de la distance égale à l'opposé de l'expression de la similarité *(cf. équation 1.2.1). Ajoutons que le chapitre 2 sera consacré à une présentation plus complète des mesures de distance.*

Remarque 1.3.4 *La création d'un graphe de voisinage nécessitera le calcul de la matrice des distances entre chacun des n points. Cette étape est réalisée avec une complexité en $O(p \times n^2)$.*

Définition 1.3.3 Voisin. *Un point sera voisin d'un autre s'il est relié à ce dernier par une arête. L'ensemble des voisins d'un point α sera noté $V(\alpha)$.*

$V(\alpha) = \{\beta \in \Sigma | (\alpha, \beta) \in A\}$

Définition 1.3.4 Connexité. *Un graphe est dit connexe lorsqu'il existe au moins une suite d'arêtes entre chaque sommet de G. Ainsi, dans un graphe connexe, pour tout couple de points $\{\alpha, \beta\} \in \Sigma^2$, il existe une succession d'arêtes joignant α à β.*

Définition 1.3.5 Cycle. *Un cycle est un chemin qui, passant par un sommet donné, retourne à ce même sommet par un suite d'arêtes différentes.*

Définition 1.3.6 Arbre. *Un arbre est un graphe connexe sans cycle.*

Définition 1.3.7 Graphe de voisinage. *Un graphe de voisinage est un graphe obtenu par application d'une structure de voisinage sur ses sommets telle que les k-plus proches voisins, la structure de Gabriel, la structure des voisins relatifs ou des polyhèdres de Delaunay.*

Nous allons à présent détailler les structures de voisinage énoncées ci-dessus en nous limitant à celles qui permettent d'aboutir à des graphes de voisinage connexe (nous écarterons donc les k-plus proches voisins). Nous présenterons nos graphes sur le même jeu de données composés de plus d'une trentaine de points décrits par deux variables X_1 et X_2 (*cf.* figure 1.2). Certes, les graphes de voisinage peuvent s'appliquer à des espaces $\mathbb{R}^p$ mais, pour des commodités de représentation, nous limitons notre espace de description à deux dimensions.

1.3.3 Graphe des polyèdres de Delaunay

1.3.3.1 Introduction

Au cours du XX$^{\text{e}}$ siècle, de nombreux problèmes de mathématique et de physique ont amené les chercheurs à s'intéresser à un type particulier de cas géométrique : les diagrammes de proximité, notion formalisée par le mathématicien russe Voronoï. Ces travaux ont été poursuivi par son compatriote Delaunay qui a défini une triangulation à partir du diagramme de Voronoï. Nous donnons ces éléments à titre indicatif, les algorithmes et leurs complexités, trop coûteuses dans $\mathbb{R}^p$ pour être utilisables dans des cas concrets, pourront être trouvés par exemple dans [PS88].

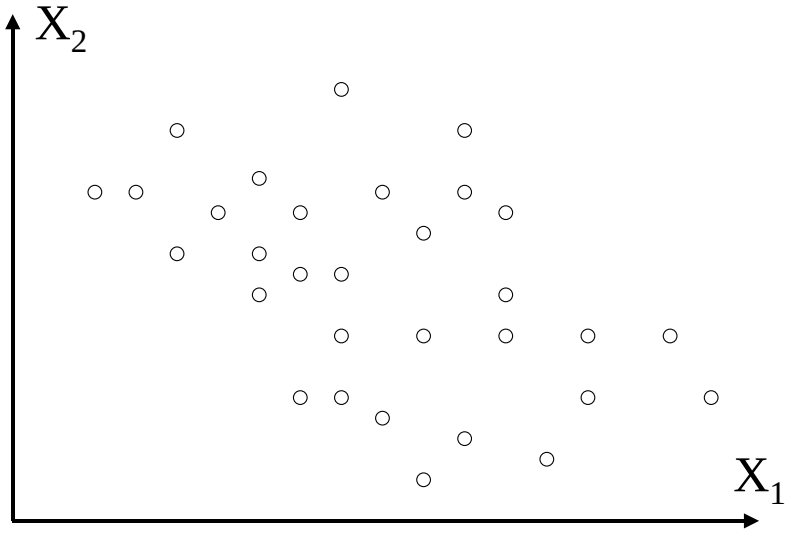

FIG. 1.2 – Données représentées dans un espace $\mathbb{R}^2$

1.3.3.2 Diagramme de Voronoï

Remarque 1.3.5 *Pour des raisons de simplicité, nous définissons dans un premier temps nos concepts dans* $\mathbb{R}^2$*. Ces derniers seront étendus à* $\mathbb{R}^p$ *par la suite.*

Remarque 1.3.6 *Ayant besoin de distinguer les points servant à la constitution du diagramme de Voronoï (et de la triangulation de Delaunay qui lui est duale) des autres points, nous considérons plus spécifiquement les* n_a *exemples de* $\Sigma = \Omega_a$*. Nous rappelons que* $\Omega_a \cup \Omega_t = \Omega$*.*

Définition 1.3.8 Région de Voronoï. *Une région de Voronoï associée à un point* $\alpha \in \Omega_a$*, et notée* Vor_α*, définit un espace qui indique que tout point* $\beta \notin \Omega_a$ *(par exemple* $\beta \in \Omega_t$*) est plus proche de* α *que tout autre point de* Ω_a*. L'équation 1.3.3 définit un tel espace qui se trouve être un polygone.*

$$Vor_\alpha = \{\beta \in \mathbb{R}^2,\ \delta(\alpha,\beta) \leqslant \delta(\alpha,\gamma),\ \forall \gamma \in \Omega_a - \{\alpha\}\} \tag{1.3.3}$$

Définition 1.3.9 Diagramme de Voronoï. *Un diagramme de Voronoï, noté*

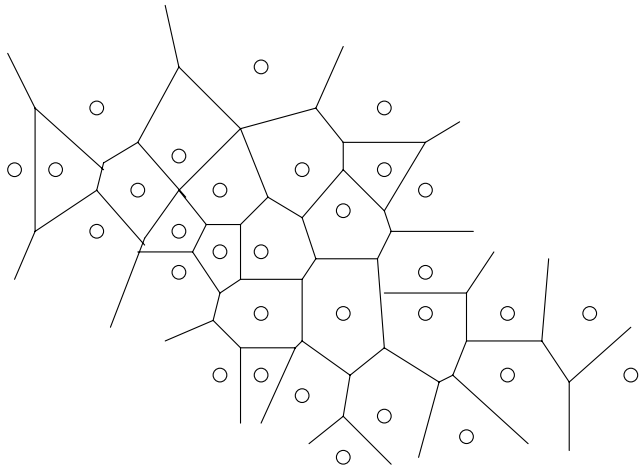

FIG. 1.3 – Diagramme de Voronoï

DV_{Ω_a}, est décrit comme étant l'union des régions de Voronoï de tous les points de Ω_a.

$$DV_{\Omega_a} = \bigcup_{\alpha \in \Omega_a} Vor_\alpha \tag{1.3.4}$$

Un diagramme de Voronoï, défini en 1.3.4, est représenté en figure 1.3. Dans ce diagramme, une arête sépare tout point de son plus proche voisin.

1.3.3.3 Triangulation de Delaunay

Définition 1.3.10 Triangulation de Delaunay (1). *On définit la triangulation de Delaunay d'un ensemble de points du plan comme étant le dual du diagramme de Voronoï correspondant.*

La construction de la triangulation de Delaunay est réalisée en reliant par un segment toutes les paires de points de Ω_a dont les régions de Voronoï correspondantes sont adjacentes (c'est-à-dire séparées par une arête de

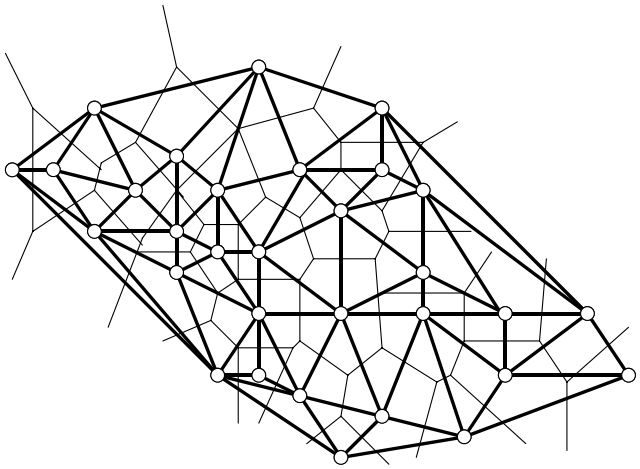

FIG. 1.4 – Passage du diagramme de Voronoï à la triangulation de Delaunay

Voronoï) (voir figure 1.4). Nous obtenons ainsi le graphe représenté sur la figure 1.5.

La triangulation de Delaunay présente plusieurs propriétés :
- elle est unique ;
- elle est complète ;
- les cercles passant par les trois sommets de chaque triangle ne contiennent aucun autre point en leur intérieur.

De la dernière propriété, nous pouvons déduire une autre manière de définir la triangulation de Delaunay à partir de chaque triangle :

Définition 1.3.11 Triangulation de Delaunay (2). *La triangulation de Delaunay est un graphe connexe dans lequel tout triplet de points* $\{\alpha, \beta, \gamma\} \in \Omega_a^3$ *sera considéré comme formant un triangle de Delaunay si et seulement si le cercle circonscrit à* $\{\alpha, \beta, \gamma\}$ *ne contient aucun autre point de* Ω_a.

Cette définition peut être étendue au cas où les points sont projetés dans un espace à p dimensions. Tout comme [Seb96], nous parlerons alors de « graphe des polyèdres de Delaunay ».

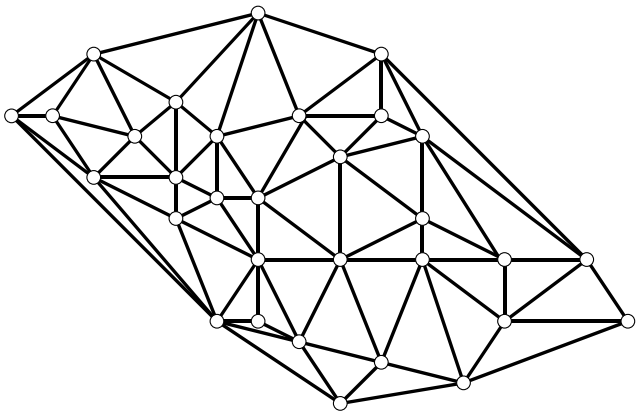

FIG. 1.5 – Triangulation de Delaunay

Définition 1.3.12 Graphe des polyèdres de Delaunay (GPD). *Dans* $\mathbb{R}^p$, $(p+1)$ *points définissent un polyèdre de Delaunay si et seulement si l'hypersphère circonscrite aux* $(p+1)$ *points ne contient aucun point. Le graphe des polyèdres de Delaunay sera défini comme étant l'union des polyèdres de Delaunay.*

En raison de la grande complexité algorithmique due au passage à p dimensions, la construction des polyèdres de Delaunay a donné lieu à peu de travaux. En effet, la construction de polyèdres de Delaunay avec n points dans un espace à p dimensions est en $O(\frac{p^4}{2} n^{\frac{p+2}{2}})$ (voir par exemple [Seb96]).

1.3.4 Arbre recouvrant minimal

Définition 1.3.13 Arbre recouvrant minimal (ARM). *Soit* (G, L) *un graphe valué connexe. Soient* H *un arbre recouvrant du graphe d'origine* G *et* $L(H)$ *la longueur de l'arbre* H *définie comme étant la somme des longueurs des arêtes qui le compose. L'arbre recouvrant* (H, L) *est dit minimal si la somme des longueurs* $L(H)$ *est minimale.*

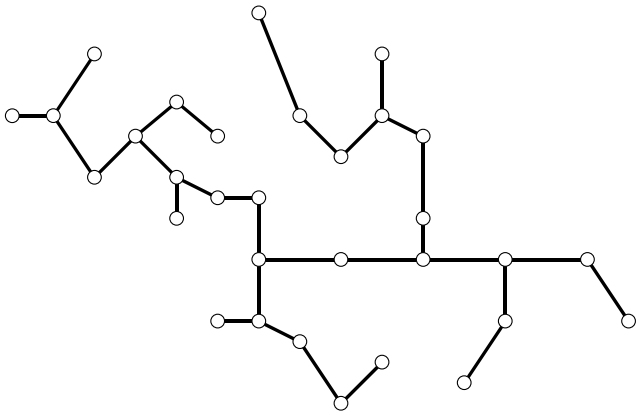

FIG. 1.6 – Arbre recouvrant minimal

L'arbre recouvrant minimal est un graphe sans cycle dont le nombre d'arêtes est égal à $(n-1)$. La démonstration de ce résultat est par exemple donnée dans [BG88].

Il existe différentes manières de construire un arbre recouvrant minimal tel que celui représenté en figure 1.6.

Un premier algorithme, développé par Prim [Pri57], est adapté aux matrices d'adjacence : il consiste à choisir un sommet initial de manière arbitraire et à rejoindre tous les sommets en ajoutant progressivement une arête de longueur minimale à l'arbre déjà construit. Dans l'algorithme 7 qui décrit cette méthode, la séquence de recherche de l'arête de longueur minimale est répétée $(n-1)$ fois, c'est-à-dire tant que l'arbre recouvrant minimal ne comporte pas les n sommets.

Une seconde manière, développée par Kruskal [Kru56], est adaptée aux listes de successeurs et graphes contenant peu d'arêtes. L'algorithme de Kruskal (voir algorithme 8) établit une liste des arêtes du graphe, ordonnée par valuation croissante, et les arêtes sont ajoutées à condition de ne pas créer de cycle, sachant qu'une nouvelle arête ne crée de cycle que si ses deux extrémités appartiennent à une même composante connexe.

Algorithme 7 Arbre recouvrant minimal, méthode de Prim

ARM $\leftarrow \varnothing$ {l'ensemble des arêtes de l'arbre recouvrant minimal est vide}
choisir un sommet quelconque $\sigma \in \Sigma$
Sommets_Utilisés $\leftarrow \{\sigma\}$
Sommets_Restants $\leftarrow \Sigma - \{\sigma\}$
tant que $Card$(ARM) $< (Card$(Sommets_Utilisés) $- 1)$ **faire**
  choisir l'arête $(\alpha, \beta) \in A$ avec $\delta(\alpha, \beta)$ minimal, $\alpha \in$ Sommets_Utilisés et $\beta \in$ Sommets_Restants
  Sommets_Utilisés $\leftarrow$ Sommets_Utilisés $\cup \{\beta\}$
  Sommets_Restants $\leftarrow$ Sommets_Restants $- \{\beta\}$
  ARM $\leftarrow$ ARM $\cup \{(\alpha, \beta)\}$
fin tant que

Algorithme 8 Arbre recouvrant minimal, méthode de Kruskal

ARM $\leftarrow \varnothing$ {l'ensemble des arêtes de l'arbre recouvrant minimal est vide}
$A' \leftarrow$ Tri(A, croissant)
pour $i \leftarrow 1$ **à** $Card(A')$ **faire**
  $(\alpha, \beta) \leftarrow$ l'arête numéro i dans A'
  si ARM $\cup \{(\alpha, \beta)\}$ ne crée pas de cycle **alors**
    ARM $\leftarrow$ ARM $\cup \{(\alpha, \beta)\}$
  fin si
fin pour

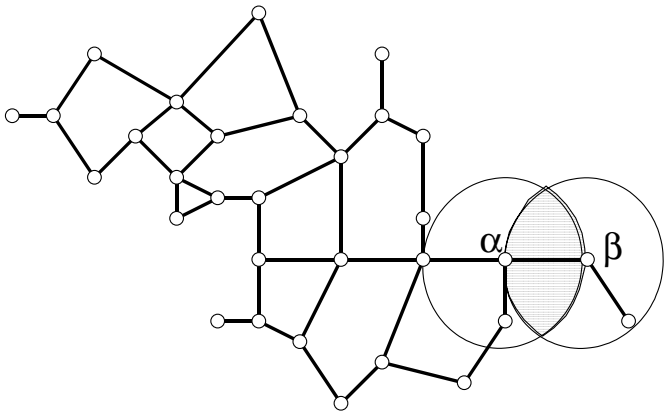

FIG. 1.7 – Graphe des voisins relatifs de Toussaint

1.3.5 Graphe des voisins relatifs

Définition 1.3.14 Graphe des voisins relatifs (GVR). *Le graphe des voisins relatifs défini par Toussaint [Tou80] d'un ensemble de sommets de Σ est un graphe où deux points α et β de Σ sont reliés par une arête si et seulement si il n'existe pas d'autre point γ de Σ plus proche à la fois de α et de β.*

La propriété énoncée ci-dessus se traduit aussi par l'équation 1.3.5.

$$\delta(\alpha,\beta) \leqslant \max_{\gamma \neq \alpha,\beta} (\delta(\alpha,\gamma), \delta(\beta,\gamma)) \tag{1.3.5}$$

L'équation 1.3.5 signifie que la « lunule » (en anglais "*lune*") obtenue par l'intersection des hypersphères de centres α et β et de rayon la longueur de l'arête (α,β) (soit $\delta(\alpha,\beta)$), ne contient aucun autre point de Σ.

Sur le graphe des voisins relatifs représenté en figure 1.7, la lunule définie entre les deux points α et β apparaît en gris clair. Comme cette lunule est vide, les points α et β sont reliés par une arête.

Algorithme 9 Graphe des voisins relatifs

```
GVR ← A      {le graphe des voisins relatifs prend toutes les arêtes}
pour α ← 1 à (n − 1) faire
  pour β ← (α + 1) à n faire
    γ ← 1
    poursuite_recherche ← vrai
    tant que (γ ⩽ n) et poursuite_recherche faire
      si (γ ≠ α) et (γ ≠ β) alors
        si δ(α, β) ⩽ max(δ(α, γ), δ(β, γ)) alors
          poursuite_recherche ← faux
          GVR ← GVR −{(α, β)}
        fin si
      fin si
      γ ← γ + 1
    fin tant que
  fin pour
fin pour
```

La construction d'un graphe des voisins relatifs est exposée dans l'algorithme 9. Au départ, le graphe prend toutes les arêtes de A, c'est-à-dire l'ensemble des arêtes reliant chaque sommet à tous les autres. Pour tout couple de points α et $\beta \in \Sigma^2$, on recherche la présence d'un point γ au sein de la lunule $\mathscr{L}_{(\alpha,\beta)}$. Si un tel point γ est trouvé dans $\mathscr{L}_{(\alpha,\beta)}$, on supprime l'arête (α, β) du graphe des voisins relatifs. Cet algorithme est d'une complexité en $O(n^3)$, avec n le nombre de sommets du graphe.

La complexité ne peut pas être diminuée dans le cas général mais il existe des améliorations de l'algorithme quand le nombre de variables prédictives p est inférieur à 3 [JT92].

1.3.6 Graphe de Gabriel

Définition 1.3.15 Graphe de Gabriel (GG). *Le graphe de Gabriel [GS69] est un graphe connexe dans lequel, si deux points α et β sont reliés par une arête, alors l'hypersphère de diamètre $\delta(\alpha, \beta)$ ne contient aucun point de Σ.*

Si nous appelons μ le centre de l'arête (α, β), les sommets α et β seront voisins au sens de Gabriel si et seulement si ils vérifient la propriété suivante :

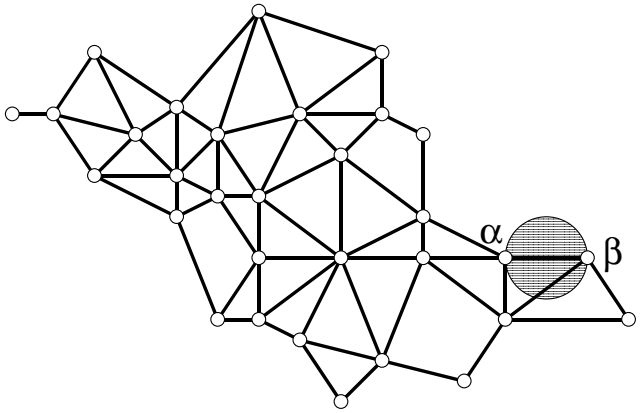

FIG. 1.8 – Graphe de Gabriel

$$\forall \gamma \in \Omega, \delta(\gamma, \mu) \geqslant \delta(\alpha, \mu) = \delta(\beta, \mu) = \frac{\delta(\alpha, \beta)}{2} \tag{1.3.6}$$

Nous avons représenté en figure 1.8 un graphe de Gabriel. Comme le cercle (en partie hachurée) qui a pour diamètre $\delta(\alpha, \beta)$ est vide, les deux points α et β sont reliés par une arête.

L'algorithme 10 décrit cette méthode. Tout comme l'algorithme de construction du graphe des voisins relatifs, la constitution du graphe de Gabriel est d'une complexité en $O(n^3)$.

1.3.7 Graphes de voisinage et apprentissage supervisé

1.3.7.1 Introduction

Nous rappelons que, dans le cadre de l'apprentissage supervisé, nous avons n_t individus $\omega_t \in \Omega_t$ décrits par p variables prédictives dont nous cherchons à prédire l'étiquette $\widehat{Y}(\omega_t)$ à partir d'un modèle φ. Le modèle φ est constitué par les n_a individus $\omega_a \in \Omega_a$ pour lesquels la valeur $Y(\omega_a)$ est

Algorithme 10 Graphe de Gabriel

```
GG ← A      {le graphe de Gabriel prend toutes les arêtes}
pour α ← 1 à (n − 1) faire
  pour β ← (α + 1) à n faire
    μ ← centre(α, β)
    γ ← 1
    poursuite_recherche ← vrai
    tant que (γ ⩽ n) et poursuite_recherche faire
      si (γ ≠ α) et (γ ≠ β) alors
        si δ(μ, α) ⩾ δ(μ, γ) alors
          poursuite_recherche ← faux
          GG ← GG −{(α, β)}
        fin si
      fin si
      γ ← γ + 1
    fin tant que
  fin pour
fin pour
```

connue, ainsi que les valeurs des variables prédictives $X_i(\omega_a)$ avec $i \in \{1, p\}$.

Dans le contexte des graphes de voisinage, comme dans celui des apprentissages à base d'exemples, les différentes variables prédictives $X_1, ..., Xp$ influent sur la région de l'hyperespace $\mathbb{R}^p$ où va se retrouver un point donné $\omega \in \Omega = \Omega_a \cup \Omega_t$. La prédiction de l'étiquette d'un point $\omega_t \in \Omega_t$ se fera à partir des étiquettes connues des n_a points $\omega_a \in \Omega_a$ qui lui sont similaires, et cette similarité sera estimée à travers la structure de voisinage ; deux points sont en effet voisins s'ils sont reliées par une arête dans le graphe de voisinage. Nous signalons que les points Ω_a ne sont pas nécessaires dans leur intégralité, un filtrage peut être réalisé afin de ne retenir que les exemples les plus caractéristiques. L'attribution de l'étiquette $\widehat{Y}$ au point ω_t relèvera d'un classement assez similaire à celui des k-plus proches voisins : l'étiquette attribuée à ω_t sera la plus fréquente parmi ses voisins (*cf.* algorithme 6).

1.3.7.2 Inclusions respectives des différents graphes de voisinage

Avant de décrire plus en avant comment un classement peut être réalisé à partir d'un graphe de voisinage, nous allons indiquer quels rapports, en particulier d'inclusion, existent entre les différentes structures de graphes de

voisinage présentées jusqu'ici.

L'arbre recouvrant minimal (figure 1.6) est, par définition, le graphe comprenant le moins d'arêtes possibles pour relier tous les sommets, ces arêtes donnant un arbre de longueur minimale. Il est par conséquent inclus dans tous les autres graphes de voisinage présentés précédemment.

La lunule $\mathscr{L}_{\alpha,\beta}$ entre deux points α et β définit une zone interdite à tout point γ pour que ces deux points α et β soient reliés par une arête dans un graphe des voisins relatifs. Cette zone comprend le cercle (dans $\mathbb{R}^2$) ou l'hypersphère (dans le cadre général de $\mathbb{R}^p$) de diamètre $\delta(\alpha,\beta)$, zone interdite à tout point γ pour que les points α et β soient reliés par une arête dans un graphe de Gabriel (voir les figures 1.7 et 1.8). Ainsi, le graphe des voisins relatifs est inclus dans le graphe de Gabriel.

Dans [PS88], Preparata et Shamos résument les inclusions relatives des différents graphes de voisinage de la manière suivante :

$$ARM \subseteq GVR \subseteq GG \subseteq GPD \tag{1.3.7}$$

Ces diverses inclusions apportent une information complémentaire sur la structure relative des différents graphes de voisinage. L'arbre recouvrant minimal est le graphe comprenant le moins d'arêtes, il représente le *squelette* des points dans l'espace $\mathbb{R}^p$. L'information de structure apportée par les autres graphes de voisinage est, quant à elle, plus globale.

1.3.7.3 Apprentissage et classement par graphe de voisinage

Tout comme les méthodes d'apprentissage à base d'exemples, les méthodes d'apprentissage par graphes de voisinage procèdent par le stockage des exemples vus dans une phase d'entraînement, avec éventuellement un filtrage qui peut être adapté des méthodes *IBL* de Aha *et al.* [AKA91] (par exemple l'algorithme *IB3* décrit en page 21).

Parmi les graphes de voisinage pouvant être utilisés en apprentissage supervisé, nous écartons le graphe des polyèdres de Delaunay (GPD). En raison de la complexité algorithmique de cette structure de voisinage lorsque les polyèdres de Delaunay sont constitués dans $\mathbb{R}^p$, une procédure d'apprentissage et de classement par ce genre de graphe semble prohibitive.

L'apprentissage supervisé à partir d'un arbre recouvrant minimal (ARM) est possible mais d'un intérêt réduit. En effet, un ARM se rapporte pour de nombreux points à la recherche du ou des 2 plus proches voisins pour une plus grande complexité. De plus, un ARM se construit par le choix des arêtes

les plus courtes permettant d'obtenir une structure connexe. Or, comme ce choix dépend des longueurs de l'ensemble des arêtes et que cet ensemble est différent à chaque fois que l'on introduit un nouveau point ω_t dans l'espace de recherche, il en résulte qu'il faudrait constituer un nouvel ARM à partir des $(n_a + 1)$ points $\Omega_a \cup \{\omega_t\}$ pour les n_t points de Ω_t.

Contrairement à l'arbre recouvrant minimal, l'apprentissage et le classement à partir de graphes de voisins relatifs (GVR) ou de Gabriel (GG) peuvent se faire globalement sur l'ensemble des points de Ω_t. Ce traitement global s'explique par le fait que pour le GVR ou le GG, la structure de voisinage d'un point peut être donnée de manière locale (*cf.* les algorithmes 9 et 10) : les points α et β ne sont reliés par une arête (c'est-à-dire qu'ils sont considérés comme « voisins ») qu'à condition qu'il n'y ait pas d'autre point γ soit dans la lunule $\mathscr{L}_{(\alpha,\beta)}$ soit dans l'hypersphère de diamètre $\delta(\alpha, \beta)$ pour, respectivement, le GVR ou le GG. Les différentes étapes de l'utilisation de ces deux graphes en tant que méthode prédictive sont indiquées dans l'algorithme 11.

1.4 Conclusion

Dans ce chapitre, nous avons présenté des graphes de voisinage et indiqué comment, à travers une utilisation analogue aux méthodes d'apprentissage à base d'exemples, ils pouvaient être employés dans le contexte de l'apprentissage supervisé.

Comme tout apprentissage à base d'exemples, l'intelligence des graphes de voisinage ne découle pas de l'abstraction de concepts sur les données mais de la manière dont des individus sont considérés similaires à d'autres. Étant donné que les graphes de voisinage ne produisent pas de connaissances générales extraites des différents exemples de l'échantillon d'apprentissage, on pourrait considérer qu'il s'agit d'outils assez peu appropriés dans le cadre de l'ECD ou dans celui des sciences cognitives. Dans le premier contexte, en effet, l'accent est mis sur la taille des bases de données d'où sont extraites les connaissances, et un modèle prédictif reposant sur le stockage, même limité par un filtrage, d'exemples de la base, semble donc vraiment inadéquat. Les graphes de voisinage semblent aussi éloignés du second contexte car il faut reconnaître que la construction de ces graphes est bien plus guidée par des propriétés issues de structures géométriques que par des modèles hypothétiques du système cognitif.

Or, si les graphes de voisinage, en raison de leur mode d'apprentissage

Algorithme 11 Classement par graphes de voisinage

$\Omega'_a \leftarrow$ filtrage(Ω_a) {filtrage éventuel pour retenir n'_a points sur les n_a}
GV $\leftarrow A_{(\Omega_t, \Omega'_a)}$ {le graphe de voisinage contient toutes les arêtes reliant les points de Ω_t à Ω'_a}
$\alpha \in \Omega_t, \beta \in \Omega'_a, \gamma \in \Omega'_a$
pour $\alpha \leftarrow 1$ **à** n_t **faire**
  $V(\alpha) \leftarrow \varnothing$ {la liste des voisins du point α est vide}
  $k \leftarrow 0$ {k indique le nombre de voisins du point α}
  pour $\beta \leftarrow 1$ **à** n'_a **faire**
    $\gamma \leftarrow 1$
    arête_entre_α_et_$\beta \leftarrow$ **vrai**
    tant que $(\gamma \leqslant n'_a)$ **et** arête_entre_α_et_β **faire**
      si $(\gamma \neq \beta)$ **alors**
        si la propriété 1.3.5 (dans le cas du GVR) ou la propriété 1.3.6 (dans le cas du GG) n'est pas vérifiée **alors**
          arête_entre_α_et_$\beta \leftarrow$ **faux**
          GV $\leftarrow$ GV $-\{(\alpha, \beta)\}$
        fin si
      fin si
      $\gamma \leftarrow \gamma + 1$
    fin tant que
    si arête_entre_α_et_β **alors**
      $V(\alpha) \leftarrow V(\alpha) \cup \{\beta\}$
      $k \leftarrow k + 1$
    fin si
  fin pour
  $\widehat{Y}(\alpha) \leftarrow \text{argmax}_{i=1}^{k}(Y(V(\alpha)_i))$
fin pour

« paresseux », apparaissent finalement limités en tant que modèle prédictif adapté aux perspectives cognitives et de l'ECD, ils n'en sont pas moins utiles dans ces deux approches par diverses propriétés :

- ils mettent en avant des relations de proximité et de distance entre les individus à partir de structure de voisinage, ce qui permet de retrouver des similarités entre les exemples suivant l'adage « qui se ressemble s'assemble » ;
- ils exploitent directement l'ensemble des variables prédictives, ce qui les rend certes sensibles aux variables prédictives non pertinentes mais qui, par ailleurs, fournit une information globale sur la manière dont se dispersent les données dans l'espace de représentation $\mathbb{R}^p$.

Par conséquent, l'intérêt que nous portons dans cette thèse aux graphes de voisinage va donc bien au-delà de leur seule utilisation en tant que système d'apprentissage supervisé. Dans les chapitres suivants, nous allons préciser les notions de similarité et de distance sur lesquelles se fondent les graphes de voisinage et exploiterons les propriétés de ces derniers à rendre compte de l'organisation de l'espace de représentation. Nous verrons notamment que cet outil géométrique nous permettra d'aborder différents problèmes rencontrés au sein de la démarche d'extraction des connaissances, en particulier ceux rencontrés au cours des phases préalables à celle de la fouille de données.

Nous achevons ce chapitre en faisant remarquer que nous travaillerons dorénavant de manière privilégiée avec le graphe des voisins relatifs de Toussaint. En effet, nous avons noté son avantage, avec le graphe de Gabriel, dans l'apprentissage à base de graphe de voisinage. En outre, dans le cas où nous ne connaîtrons pas toutes les valeurs des p variables prédictives mais où il nous faudra travailler avec la seule matrice de dissimilarité avec les distances $\delta(\alpha, \beta)$, $\forall \alpha, \beta \in \Sigma^2$, le graphe des voisins relatifs est plus facilement construit que le graphe de Gabriel puisque ce dernier demande de calculer la valeur du point μ, le centre du segment (α, β).

Mesures de distance et indices de similarité

Sommaire

Chapitre 2

Mesures de distance et indices de similarité

Résumé

Ce chapitre fait un état de l'art des diverses manières de calculer des mesures de distance entre des exemples avec des variables numériques, mais aussi booléennes ou catégorielles.

Nous exposons comment estimer la similarité entre des exemples, exploitant à cet effet les mesures de distance présentées, et nous indiquons comment construire une matrice de dissimilarité pouvant servir à la construction des graphes de voisinage.

2.1 Introduction

Les apprentissages à base d'exemples se fondent sur la similarité existant entre des individus pour attribuer une étiquette à des exemples lorsque cette étiquette n'est pas connue. Cette similarité ainsi que le principe de construction des graphes de voisinage sont associés à des notions de proximité et de distance entre les divers exemples quand ceux-ci se répartissent dans l'espace de représentation.

Dans ce chapitre, nous présenterons les principales *mesures de distance* entre objets existant dans la littérature. Nous nous limiterons aux seules distances entre objets et n'évoquerons pas les distances entre variables que l'on peut obtenir à partir d'indices de similarité comme le coefficient de corrélation pour les variables quantitatives ou les coefficients de Tschuprow

ou de Cramer lorsque les variables sont qualitatives[1].

Par ailleurs, nous affinerons la définition que nous donnons de la *similarité*. Pour cela, nous lui donnerons un mode de calcul rigoureux et général, faisant usage des notions de distance vues précédemment, tout en exposant des situations où la similarité entre individus peut être obtenue indépendamment de l'existence de variables prédictives.

2.2 Mesures de distance

2.2.1 Introduction

Pour caractériser la relation de proximité existant entre deux individus α et β projetés dans un espace de représentation multidimensionnel, il n'existe pas une distance unique mais un ensemble de distances possibles. Par conséquent, l'emploi d'une mesure de distance donnée sera guidé par la volonté de mettre en avant certaines propriétés de l'éloignement de ces objets dans l'espace de représentation ou par le souci de retenir un mode de calcul de la distance peu coûteux en complexité algorithmique. Il faut ajouter à cela que d'autres facteurs vont intervenir sur le choix de la mesure de distance, notamment la nature numérique ou catégorielle des variables prédictives à partir desquelles sont calculées les distances.

Les différentes distances que nous allons exposer dans cette section sont proposées par Chandon et Pinson [CP81].

2.2.2 Distances obtenues à partir de variables numériques

2.2.2.1 Introduction

Nous nous situons toujours le cadre de l'apprentissage supervisé, disposant d'un ensemble de n individus d'une population Ω décrits par un ensemble de p variables prédictives $(X_1, X_2, ..., X_p)$ (numériques, booléennes ou catégorielles) ainsi que d'une variable à prédire Y catégorielle. Dans un premier temps, ces variables prédictives sont supposées numériques.

[1] Nous notons néanmoins que, dans l'approche de l'extraction des connaissances à partir de données, les distances entre variables trouvent un champ d'application parmi les méthodes de pré-traitement. Elles sont en effet d'un grand intérêt pour le problème de la sélection des variables prédictives pertinentes.

2.2.2.2 Distance euclidienne

Distance euclidienne usuelle Cette distance, associée par tradition aux travaux du mathématicien de la Grèce antique Euclide, est la distance considérée classiquement comme la plus « naturelle ». Son mode de calcul est donné en équation 2.2.1 et revient, dans un espace de représentation orthonormé en deux dimensions ($p = 2$), au théorème de Pythagore.

$$\delta(\alpha, \beta) = \sqrt{\sum_{i=1}^{p} \left(X_i(\alpha) - X_i(\beta)\right)^2} \tag{2.2.1}$$

Distance euclidienne standardisée Lorsque les unités de mesure diffèrent d'une variable à une autre, comparée aux autres variables, la variable qui a la plus forte variance exerce un effet excessif dans la distance euclidienne calculée entre les points se répartissant dans l'espace de représentation. Par conséquent, il est souvent nécessaire de corriger cette distance en procédant à une *standardisation* des différentes valeurs des p variables prédictives X_i, c'est-à-dire en réalisant un centrage (soustraction de la valeur moyenne $\bar{X}_i$) et une réduction (division par l'écart-type σ_{X_i}), comme indiqué en équation 2.2.2. De cette manière, toutes les variables sont mesurées en unités identiques dépendant de l'écart-type.

$$\delta(\alpha, \beta) = \sqrt{\sum_{i=1}^{p} \left(X_i^{CR}(\alpha) - X_i^{CR}(\beta)\right)^2} \tag{2.2.2}$$

avec $X_i^{CR}(\omega) = \frac{X_i(\omega) - \bar{X}_i}{\sigma_{X_i}}$,
$\bar{X}_i = \frac{1}{n} \sum_{\omega=1}^{n} X_i(\omega)$
et $\sigma_{X_i} = \sqrt{\frac{1}{n} \sum_{\omega=1}^{n} \left(X_i(\omega) - \bar{X}_i\right)^2}$

Autres pondérations de la distance euclidienne La standardisation évoquée précédemment revient à pondérer la différence entre le carré des valeurs de $X_i(\alpha)$ et $X_i(\beta)$ par l'inverse de la variance $\sigma_{X_i}^2$ de chaque variable X_i, comme indiqué en équation 2.2.3.

$$\begin{aligned}
\delta(\alpha,\beta) &= \sqrt{\sum_{i=1}^{p}\left(X_i^{CR}(\alpha)-X_i^{CR}(\beta)\right)^2} \\
&= \sqrt{\sum_{i=1}^{p}\left(\frac{X_i(\alpha)-\bar{X_i}}{\sigma_{X_i}}-\frac{X_i(\beta)-\bar{X_i}}{\sigma_{X_i}}\right)^2} \\
&= \sqrt{\sum_{i=1}^{p}\left(\frac{\left(X_i(\alpha)-\bar{X_i}\right)-\left(X_i(\beta)-\bar{X_i}\right)}{\sigma_{X_i}}\right)^2} \\
&= \sqrt{\sum_{i=1}^{p}\left(\frac{X_i(\alpha)-X_i(\beta)}{\sigma_{X_i}}\right)^2} \\
&= \sqrt{\sum_{i=1}^{p}\frac{1}{\sigma_{X_i}^2}\left(X_i(\alpha)-X_i(\beta)\right)^2}
\end{aligned} \tag{2.2.3}$$

Ainsi, d'une façon générale, les différentes pondérations existant pour la distance euclidienne vont toucher soit la somme des différences dans leur globalité (w_1) soit chacune des différences entre $X_i(\alpha)$ et $X_i(\beta)$ avec un poids spécifique $w_2(i)$ pour chacune des variables X_i (*cf.* équation 2.2.4).

$$\delta(\alpha,\beta) = \sqrt{w_1\sum_{i=1}^{p} w_2(i)\left(X_i(\alpha)-X_i(\beta)\right)^2} \tag{2.2.4}$$

La distance euclidienne standardisée rentre ainsi dans le cadre général de l'équation 2.2.4 avec les pondérations $w_1 = 1$ et $w_2(i) = 1/(\sigma_{X_i})^2$.

Dans le cas où la distance euclidienne doit être comprise entre 0 et 1, il est possible d'appliquer la distance de Clark (*cf.* équation 2.2.5) qui consiste à pondérer la différence entre $X_i(\alpha)$ et $X_i(\beta)$ par l'inverse du carré de leur somme $\left(w_2(i) = 1/(X_i(\alpha)+X_i(\beta))^2\right)$. Quand à la somme globale, elle est pondérée par l'inverse du nombre de variables ($w_1 = 1/p$).

$$\delta(\alpha,\beta) = \sqrt{\frac{1}{p}\sum_{i=1}^{p}\frac{1}{\left(X_i(\alpha)+X_i(\beta)\right)^2}\left(X_i(\alpha)-X_i(\beta)\right)^2} \tag{2.2.5}$$

Enfin, nous signalons qu'il existe des modes de calcul particuliers de la distance euclidienne qui attribuent un poids particulier $w_2(i)$ à chacune de ces variables afin de tenir compte de la qualité ou de la fiabilité des mesures. Ajoutons à cela la *distance euclidienne moyenne* adaptée à la situation où les individus α et β sont caractérisés par un nombre différent de variables. Dans ce cas, tout comme pour la distance de Clark, la somme des différences est pondérée par l'inverse du nombre de variables $w_1 = (1/p)$ et chaque différence

peut être pondérée par un poids $w_2(i)$ donné. Nous écartons toutefois cette situation de notre cadre car nous supposons connues les valeurs $X_i(\omega)$ pour tous les individus $\omega \in \Omega$ et chacune des i variables prédictives avec $i \in \{1, p\}$, sans pouvoir attribuer a priori à une variable X_i une plus grande importante – et un plus grand poids $w_2(i)$ – qu'une variable $X_{i'}$.

2.2.2.3 Distance rectangulaire

Les distances euclidiennes, puisqu'elles dérivent du calcul du carré des différences entre les deux éléments α et β sur chacune des variables, ont tendance à donner une plus grande importance aux fortes différences plutôt qu'aux petites. De la sorte, elles mettent en valeur les objets atypiques. Dans le cas où ce sont les petites différences qui retiennent notre attention, il est préférable d'employer la distance rectangulaire.

Distance rectangulaire non pondérée La *distance rectangulaire non pondérée* est aussi connue sous le nom de *distance de Manhattan* ou "*city block*" car elle consisterait à calculer le chemin emprunté par une voiture pour se rendre d'un point à un autre dans un quartier d'une ville où toutes les rues se coupent à angle droit, alors que la distance euclidienne (usuelle) reviendrait à effectuer le même chemin à vol d'oiseau. La distance rectangulaire se calcule en faisant la somme des différences absolues des valeurs entre $X_i(\alpha)$ et $X_i(\beta)$, comme décrit par l'équation 2.2.6.

$$\delta(\alpha, \beta) = \sum_{i=1}^{p} |X_i(\alpha) - X_i(\beta)| \tag{2.2.6}$$

Pondérations de la distance rectangulaire Nous trouvons dans la littérature différentes manières de pondérer la distance rectangulaire : *pondération par l'inverse du total de la colonne* (équation 2.2.7), pondération par l'inverse des sommes des mesures appelée *distance de Canberra* (équation 2.2.8), la *distance de Bray et Curtis* où le poids est l'inverse des sommes des totaux des lignes de α et β (équation 2.2.9) ou enfin la *distance rectangulaire pondérée moyenne*, similaire au calcul de la distance euclidienne moyenne, applicable quand les individus α et β sont caractérisés par un nombre différent de variables (équation 2.2.10).

$$\delta(\alpha,\beta) = \sum_{i=1}^{p} \frac{1}{\sum_{j=1}^{n} X_j(\omega_j)} |X_i(\alpha) - X_i(\beta)| \tag{2.2.7}$$

$$\delta(\alpha,\beta) = \sum_{i=1}^{p} \frac{1}{X_i(\alpha) + X_i(\beta)} |X_i(\alpha) - X_i(\beta)| \tag{2.2.8}$$

$$\delta(\alpha,\beta) = \frac{1}{\sum_{j=1}^{p}(X_j(\alpha) + X_j(\beta))} \sum_{i=1}^{p} |X_i(\alpha) - X_i(\beta)| \tag{2.2.9}$$

$$\delta(\alpha,\beta) = \frac{1}{p} \sum_{i=1}^{p} w(i) |X_i(\alpha) - X_i(\beta)| \tag{2.2.10}$$

2.2.2.4 Distance de Chebychev

Contrairement aux distances rectangulaires qui cherchent à privilégier les petites différences, la *distance de Chebychev* ne tient compte que de la distance maximale (équation 2.2.11).

$$\delta(\alpha,\beta) = \max_{i=1}^{p} (X_i(\alpha) - X_i(\beta)) \tag{2.2.11}$$

2.2.2.5 Distance de Minkowski

Les différentes distances présentées jusqu'à présent peuvent être obtenue par la formule générale de la *distance de Minkowski*. Dans cette formule, décrite par l'équation 2.2.12, lorsque le coefficient q est égal à 2, la distance de Minkowski revient à la distance euclidienne ; quand q est égal à 1, il s'agit d'une distance rectangulaire ; le résultat est égal à la distance de Chebychev lorsque quand le coefficient q tend vers l'infini.

$$\delta(\alpha,\beta) = \left(\sum_{i=1}^{p} w(i) (X_i(\alpha) - X_i(\beta))^q \right)^{\frac{1}{q}} \tag{2.2.12}$$

2.2.2.6 Distance de Mahalanobis

Les distances présentées supposent que les p variables prédictives sur lesquelles sont calculées les différences sont indépendantes. Mais si des variables

$\alpha \backslash \beta$	1	0
1	$\sum_{1\leftrightarrow 1}$	$\sum_{1\leftrightarrow 0}$
0	$\sum_{0\leftrightarrow 1}$	$\sum_{0\leftrightarrow 0}$

TAB. 2.1 – Notation des appariements et différences de valeurs entre α et β

sont fortement corrélées entre elles, la dimension mesurée est similaire et ressort de manière exagérée dans le calcul de la distance. Parmi les méthodes qui permettent de corriger ce problème, il en est une, appelée la *distance de Mahalanobis* (équation 2.2.13), où chaque paire de variables X_i et $X_{i'}$ est pondérée par l'inverse de leur covariance.

$$\delta(\alpha,\beta) = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{p} w_{i,j}\left(X_i(\alpha) - X_i(\beta)\right)\left(X_j(\alpha) - X_j(\beta)\right)} \qquad (2.2.13)$$

avec $w_{i,j}$ l'élément inverse de la matrice de covariance.

L'utilisation de la distance de Mahalanobis corrige la corrélation existant entre les différentes variables prédictives et présente d'autres propriétés intéressantes pour la représentation des données et la classification mais elle est d'une assez grande complexité calculatoire, en particulier lorsque le nombre de variables prédictives p est important.

2.2.3 Distances obtenues à partir de variables booléennes

2.2.3.1 Introduction

Lorsque les variables prédictives X_i sont booléennes, les valeurs pour tout individu $\omega \in \Omega$ sont soit *Vrai* (ou 1) soit *Faux* (ou 0). Les distances obtenues à partir de variables booléennes résultent ainsi du comptage du nombre de caractères communs ($1 \leftrightarrow 1$ et $0 \leftrightarrow 0$) ou différents pour les valeurs 1 et 0 et il est d'usage de procéder à leurs décomptes. Le tableau 2.1 indique notre façon de noter le nombre de caractères communs et différents sachant que la somme de ces quantités donne p, le nombre total de variables prédictives binaires ($p = \sum_{1\leftrightarrow 1} + \sum_{1\leftrightarrow 0} + \sum_{0\leftrightarrow 1} + \sum_{0\leftrightarrow 0}$).

2.2.3.2 Distances binaires et forme disjonctive complète

Lorsque l'on s'intéresse à la similarité existant entre deux objets et que pour cela on emploie des métriques plaçant ces objets dans un espace de re-

présentation à p dimensions, il est bien moins aisé de travailler avec des variables prédictives catégorielles que des variables booléennes ou numériques. Aussi est-il fréquent de transformer les variables prédictives qualitatives sous forme disjonctive complète, comme nous l'expliquerons dans la sous-section 2.2.4.3 (page 58).

Nous signalons qu'il faut toutefois traiter des variables booléennes issues d'une transformation de variables catégorielles sous forme disjonctive complète avec précaution car cette transformation passe par le remplacement d'une variable qualitative par un ensemble de variables booléennes prédictives, ce qui a pour effet de donner des variables booléennes essentiellement creuses, c'est-à-dire pourvues de valeurs très majoritairement fausses. De ce fait, le nombre d'appariements de valeurs fausses entre α et β sera anormalement élevé et une variable prédictive catégorielle avec beaucoup de modalités risque d'avoir plus d'influence qu'une variable binaire ou catégorielle pourvue de peu de modalités.

2.2.3.3 Distance de Hamming

La *distance de Hamming* indique l'éloignement de deux individus α et β en fonction du nombre de traits qu'ils n'ont pas en commun, comme cela est indiqué en équation 2.2.14. Il s'agit d'une distance très utilisée en codage informatique afin de rendre compte du nombre de bits séparant deux mots.

$$\delta(\alpha, \beta) = \sum_{i=1}^{p} T(\alpha_i \neq \beta_i) \tag{2.2.14}$$

où la fonction $T(u)$ retourne la valeur numérique 1 dans le cas où l'expression booléenne u est vraie et 0 dans le cas où u est une expression fausse.

Dans le cadre indiqué précédemment, cette distance revient à faire la somme des valeurs différentes, comme indiqué en équation 2.2.15 [EMTB00].

$$\delta(\alpha, \beta) = \sum_{1 \leftrightarrow 0} + \sum_{0 \leftrightarrow 1} \tag{2.2.15}$$

2.2.3.4 Distance euclidienne binaire

La *distance euclidienne binaire* est une simple adaptation de la distance euclidienne usuelle au cas des variables booléennes. Son expression en équation 2.2.16 indique qu'elle ne revient finalement qu'à calculer la racine carrée

de la distance de Hamming [EMTB00].

$$\delta(\alpha,\beta) = \sqrt{\sum_{1\leftrightarrow 0} + \sum_{0\leftrightarrow 1}} \tag{2.2.16}$$

2.2.3.5 Indices de similarité appliqués à des données binaires

Nous notons que de nombreux indices appliqués aux variables binaires connus dans la littérature sont, plutôt que des mesures de distance, des indicateurs de proximité entre les objets α et β. Afin de ne pas les confondre avec les mesures de distances pouvant servir d'indices de dissimilarité, nous notons ι ces indices de similarité que nous allons présenter et dont nous pouvons trouver une description chez Lerman [Ler70], Chandon et Pinson [CP81] ou Esposito, Malerba, Tamma et Bock [EMTB00]. Nous donnerons dans la section 2.3 plus de précisions sur les manières de passer d'un indice de similarité à un indice de dissimilarité et réciproquement.

Indice de Russel et Rao L'*indice de Russel et Rao*, décrit en équation 2.2.17, est la proportion de traits positifs communs [Ler70, CP81, EMTB00].

$$\iota(\alpha,\beta) = \frac{\sum_{1\leftrightarrow 1}}{p} \tag{2.2.17}$$

Coefficient de simple concordance Ce coefficient, aussi connu sous le nom d'*indice de Sokal et Michener* (1958), consiste à calculer la proportion de valeurs communes entre les individus α et β. Sa formule de calcul est indiquée en équation 2.2.18 [EMTB00].

$$\iota(\alpha,\beta) = \frac{\sum_{1\leftrightarrow 1} + \sum_{0\leftrightarrow 0}}{p} \tag{2.2.18}$$

Coefficient de communauté Le coefficient précédent présente le désavantage de tenir compte aussi bien au numérateur qu'au dénominateur du nombre de fois où les valeurs de α et β sont toutes deux fausses ($\sum_{0\leftrightarrow 0}$), ce qui donne une mauvaise idée de ι quand il faut davantage prendre en compte la valeur *Vrai* que *Faux*. Ce cas se rencontre en particulier lorsque les variables booléennes sont le résultat d'une transformation de variables prédictives initialement catégorielles sous forme disjonctive complète (*cf.* sous-section 2.2.4.3). Il est alors préférable d'utiliser le *coefficient de communauté*,

appelé aussi *indice de Jaccard* (1908) comme indiqué en équation 2.2.19 [Ler70, EMTB00].

$$\iota(\alpha,\beta) = \frac{\sum_{1\leftrightarrow 1}}{\sum_{1\leftrightarrow 1} + \sum_{1\leftrightarrow 0} + \sum_{0\leftrightarrow 1}} \tag{2.2.19}$$

Indice d'Ochiai L'*indice d'Ochiai* donne une mesure de la similarité existant entre α et β à travers le rapport existant entre leurs valeurs positives similaires et la racine carrée du produit de la somme des valeurs positives similaires avec l'une ou l'autre forme de leurs différences (*cf.* équation 2.2.20) [Ler70]. Le carré de l'indice d'Ochiai est aussi connu sous le nom d'*indice d'équivalence*.

$$\iota(\alpha,\beta) = \frac{\sum_{1\leftrightarrow 1}}{\sqrt{(\sum_{1\leftrightarrow 1} + \sum_{0\leftrightarrow 1}) \times (\sum_{1\leftrightarrow 1} + \sum_{1\leftrightarrow 0})}} \tag{2.2.20}$$

Indice de Czekanowski-Dice Cet indice, dû à *Czekanowski* en 1913 et à *Dice* en 1945, attribue un poids deux fois plus important au nombre de valeurs vraies communes aux individus α et β par rapport aux valeurs de α et β qui sont différentes (*cf.* équation 2.2.21) [EMTB00].

$$\iota(\alpha,\beta) = \frac{2 \times \sum_{1\leftrightarrow 1}}{2 \times \sum_{1\leftrightarrow 1} + \sum_{1\leftrightarrow 0} + \sum_{0\leftrightarrow 1}} \tag{2.2.21}$$

Indice de Sokal et Sneath Contrairement à l'indice de Czekanowski-Dice, l'*indice de Sokal et Sneath* (1963) attribue un poids deux fois plus important pour les valeurs des individus α et β qui sont différentes (*cf.* équation 2.2.22) [Ler70, EMTB00].

$$\iota(\alpha,\beta) = \frac{\sum_{1\leftrightarrow 1}}{\sum_{1\leftrightarrow 1} + 2 \times (\sum_{1\leftrightarrow 0} + \sum_{0\leftrightarrow 1})} \tag{2.2.22}$$

Indice de Rogers et Tanimoto L'*indice de Rogers et Tanimoto* (1960), par rapport à l'indice de Sokal et Sneath, tient compte dans sa formule de calcul (*cf.* équation 2.2.23) du nombre de fois où les individus α et β ont tout deux des valeurs fausses [EMTB00].

$$\iota(\alpha,\beta) = \frac{\sum_{1\leftrightarrow 1} + \sum_{0\leftrightarrow 0}}{\sum_{1\leftrightarrow 1} + \sum_{0\leftrightarrow 0} + 2 \times (\sum_{1\leftrightarrow 0} + \sum_{0\leftrightarrow 1})} \tag{2.2.23}$$

Q de Yule Alors que les indices de similarité précédents étaient positifs, le quotient *Q de Yule* et l'indice suivant prennent leurs valeurs entre -1 à $+1$. Le mode de calcul du Q de Yule est donné par la formule 2.2.24 [Ler70, CP81, EMTB00].

$$\iota(\alpha,\beta) = \frac{(\sum_{1\leftrightarrow 1} \times \sum_{0\leftrightarrow 0}) - (\sum_{1\leftrightarrow 0} \times \sum_{0\leftrightarrow 1})}{(\sum_{1\leftrightarrow 1} \times \sum_{0\leftrightarrow 0}) + (\sum_{1\leftrightarrow 0} \times \sum_{0\leftrightarrow 1})} \tag{2.2.24}$$

Indice d'Haman L'*indice d'Haman* est égal à la différence entre les valeurs communes à α et β et leurs valeurs différentes rapportée à l'ensemble des valeurs, comme indiqué par l'équation 2.2.25 [CP81].

$$\iota(\alpha,\beta) = \frac{(\sum_{1\leftrightarrow 1} + \sum_{0\leftrightarrow 0}) - (\sum_{1\leftrightarrow 0} + \sum_{0\leftrightarrow 1})}{p} \tag{2.2.25}$$

2.2.3.6 Bilan : quel indicateur de distance binaire choisir ?

Parmi la pléthore de mesures de distances binaires et d'indicateurs de similarité dont nous venons de donner une liste – non exhaustive ! –, quelle est la formule de calcul la plus adaptée pour rendre compte des distances entre individus décrits par des variables booléennes ?

La réponse n'est pas triviale. Nous conseillons, dans le cadre de notre problématique où les variables prédictives caractérisant notre ensemble de données peuvent être à la fois des variables numériques centrées et réduites, des variables booléennes originelles ou des variables binaires issues d'une transformation de variables qualitatives sous forme disjonctive complète, d'employer la *distance euclidienne binaire*, et ceci pour trois raisons.

Tout d'abord, la distance euclidienne est d'un calcul simple et s'applique de manière équivalente aux variables prédictives, que celles-ci se trouvent sous forme binaire ou sous forme numérique, centrées et réduites.

Ensuite, la distance euclidienne, comme nous l'avons déjà souligné un peu plus haut, rend compte de propriétés de distances « naturelles », en particulier dans un espace de représentation limité à deux ou à trois dimensions.

Enfin, la distance euclidienne binaire ne tient pas compte dans son mode de calcul du nombre de fois où les individus ont tout deux des valeurs fausses ($\sum_{0\leftrightarrow 0}$), nombre qui est peu pertinent dans le cas où les variables booléennes sont issues d'une transformation de variables catégorielles sous forme disjonctive complète (*cf.* sous-section 2.2.4.3).

2.2.4 Distances obtenues à partir de variables catégorielles

2.2.4.1 Introduction

Obtenir une information de distance est un procédé qui est naturel avec des données numériques, qui peut le sembler un peu moins à partir de données binaires, et qui, de prime abord, a l'air vraiment artificiel lorsque ces données sont qualitatives.

Des techniques existent pourtant pour prendre en compte des variables prédictives qualitatives afin de considérer leurs influences pour exprimer le caractère semblable ou différent entre deux individus caractérisés par l'ensemble des variables prédictives. Nous nous proposons de présenter les principales méthodes existant dans la littérature pour obtenir des mesures de distance à partir de variables catégorielles.

2.2.4.2 Généralisation des indices appliqués initialement aux variables booléennes

Lorsque les individus sont décrits par des variables prédictives catégorielles, il est parfois possible d'adapter les indices de similarité appliqués à des variables binaires. Ainsi, le coefficient de simple concordance de Sokal et Michener (*cf.* équation 2.2.18) peut se généraliser à la proportion des variables catégorielles qui sont appariées, chaque variable pouvant être pondérée en fonction de leurs nombres de modalités catégorielles.

2.2.4.3 Transformation sous forme disjonctive complète

La généralisation proposée précédemment peut être appliquée aux indices de similarité sur des variables binaires mais n'est pas nécessairement adaptée dans le cas de mesures de distance. Une méthode plus globale permet de réécrire des variables catégorielles en variables booléennes : la *transformation sous forme disjonctive complète.*

Ce procédé consiste à remplacer chaque variable prédictive qualitative X_i contenant η modalités différentes en un ensemble de η variables binaires $X_{i,1}, X_{i,2}, ..., X_{i,\eta}$ où, pour un individu $\omega \in \Omega$, dont la valeur originelle $X_i(\omega) = \mathscr{M}$, seule la valeur de la variable $X_{i,\mathscr{M}}$ est vraie (ou à 1), les valeurs des $(\eta - 1)$ autres variables étant fausses (ou à 0). Toutes les variables prédictives qualitatives sont ainsi transformées sous forme disjonctive complète afin de former un nouvel ensemble de $p^\star$ variables prédictives.

Étant donné que les variables booléennes issues d'une transformation de variables qualitatives sous forme disjonctive complète sont plus nombreuses et comportent principalement des valeurs fausses, une attention particulière doit être portée à l'usage du mode de calcul de la distance effectuée sur les variables binaires résultant de cette transformation. En effet, dans l'ensemble des $p^\star$ variables, une variable catégorielle originelle X_i, pourvue de η modalités différentes, a été transformée en un ensemble $X_{i,1}, X_{i,2}, ..., X_{i,\eta}$ variables binaires, ce qui a tendance à sur-représenter les variables catégorielles comprenant un grand nombre de modalités différentes, variables dont les valeurs seront très majoritairement fausses (équivalentes à des matrices creuses). Ainsi, pour deux individus α et β, la somme des valeurs fausses communes $\sum_{0 \leftrightarrow 0}$ sera artificiellement très importante comparée aux valeurs vraies communes $\sum_{1 \leftrightarrow 1}$. Les mesures de distances appliquées à des variables binaires issues d'une transformation de variables qualitatives sous forme disjonctive complète devront par conséquent tenir compte de ce phénomène.

2.2.4.4 Mélange de données numériques, booléennes et catégorielles

Une façon plus directe de traiter aussi bien des données numériques que booléennes ou catégorielles consiste à procéder comme le font Aha, Kibler et Albert [AKA91]. Nous rappelons brièvement la manière dont ces auteurs procèdent pour calculer leur distance, celle-ci étant déjà détaillée dans la section 1.2.4 (page 17) du premier chapitre.

Lorsque les variables prédictives sont numériques, Aha *et al.* utilisent une distance euclidienne sur les valeurs centrées et réduites. Pour des variables prédictives booléennes ou catégorielles, ils attribuent une valeur de différence maximale égale à 1 quand les deux individus ont des modalités différentes et cette différence est nulle lorsque les modalités sont identiques.

Toutefois, cette manière de procéder, même si elle garde le même nombre de variables prédictives qualitatives – contrairement à la transformation sous forme disjonctive complète – présente quelques désavantages. En effet, ce mode de calcul de la distance à tendance à être excessif, attribuant trop facilement une valeur de distance maximale quand deux modalités des individus α et β diffèrent.

2.2.4.5 Métrique de différences de valeurs

Les diverses façons de traiter des variables prédictives catégorielles énoncées jusqu'alors considèrent que toutes les distances entre des modalités différentes d'une variable prédictive donnée sont égales. Cette manière de procéder n'est toutefois pas toujours judicieuse dans le cadre de l'apprentissage supervisé. Par exemple, imaginons que nous ayons à traiter d'un problème concernant les crèmes solaires. Nous devons apprendre quelle est la capacité de résistance de personnes s'exposant au soleil pendant une durée déterminée (estimée à travers la présence ou l'absence de coups de soleil) en fonction d'un ensemble de paramètres (âge, sexe, couleurs des yeux, des cheveux, de la peau). Pour la variable prédictive « couleur des cheveux », il serait inadéquat de considérer que la modalité « blond » est aussi distante de la modalité « brun » qu'elle l'est des modalités « roux » ou « châtain » car la peau des bruns est généralement plus résistante au soleil que celle des personnes à cheveux blonds ou roux.

Cependant, comment savoir a priori que, pour une variable prédictive catégorielle donnée, certaines modalités sont plus proches que d'autres ? De plus, comment établir une telle distance entre les modalités d'une variable prédictive catégorielle sachant que si cette variable est utilisée dans une autre problématique d'apprentissage, les distances préalablement définies entre modalités ne seront peut-être plus adaptées ?

Stanfill et Waltz [SW86] ont proposé une réponse à ces questions en imaginant une méthode permettant d'obtenir une information de distance *contextuelle* à partir de variables prédictives catégorielles. Cette mesure de distance, connue sous le nom de « métrique de différences de valeurs » (“*value difference metric*”, ou “*VDM*”), introduit dans son mode de calcul la distribution des étiquettes de la variable à prédire. Pour une variable prédictive qualitative X_i donnée, la métrique de différences de valeurs pour deux individus α et β sera calculée selon la formule 2.2.26.

$$\delta^q_{X_i}(\alpha,\beta) = \sum_{e=1}^{r} \left| \frac{nb_{X_i}(\alpha,e)}{nb_{X_i}(\alpha)} - \frac{nb_{X_i}(\beta,e)}{nb_{X_i}(\beta)} \right|^q \qquad (2.2.26)$$

où :

- e est une étiquette de Y ;
- r est le nombre total d'étiquettes de Y ;
- $nb_{X_i}(\gamma)$ est le nombre d'individus $\omega \in \Omega$ (ou Ω_a) qui ont la même modalité que γ pour la variable X_i ;

Ω	X_1	X_2	Y
ω_1	$\mathscr{A}$	$\mathscr{D}$	*Vrai*
ω_2	$\mathscr{A}$	$\mathscr{D}$	*Vrai*
ω_3	$\mathscr{A}$	$\mathscr{E}$	*Vrai*
ω_4	$\mathscr{B}$	$\mathscr{E}$	*Vrai*
ω_5	$\mathscr{B}$	$\mathscr{E}$	*Vrai*
ω_6	$\mathscr{B}$	$\mathscr{D}$	*Faux*
ω_7	$\mathscr{B}$	$\mathscr{D}$	*Faux*
ω_8	$\mathscr{C}$	$\mathscr{E}$	*Faux*
ω_9	$\mathscr{C}$	$\mathscr{D}$	*Faux*
ω_{10}	$\mathscr{C}$	$\mathscr{E}$	*Faux*

TAB. 2.2 – Base de données illustrant le calcul de la métrique *VDM*

$X_1 \backslash Y$	*Vrai*	*Faux*	Total
$\mathscr{A}$	3	0	3
$\mathscr{B}$	2	2	4
$\mathscr{C}$	0	3	3

TAB. 2.3 – Répartition des individus suivant les modalités de X_1

- $nb_{X_i}(\gamma, e)$ est le nombre d'individus $\omega \in \Omega$ (ou Ω_a) qui ont la même modalité que γ pour la variable prédictive X_i et qui ont pour étiquette $e = Y(\omega)$;
- q est un coefficient égal à 1 pour une distance rectangulaire ou égal à 2 pour une distance euclidienne.

L'équation 2.2.26 indique que, pour la métrique de différences de valeurs (*VDM*), deux modalités catégorielles d'une variable prédictive sont considérées d'autant plus proches que la distribution des étiquettes suivant ces modalités est similaire. Nous allons illustrer le calcul de la métrique *VDM* à travers un exemple décrit dans le tableau 2.2. Il s'agit d'une base d'apprentissage comprenant un échantillon d'une dizaine d'individus, avec deux variables prédictives (X_1, dont les modalités sont $\mathscr{A}$, $\mathscr{B}$ et $\mathscr{C}$, et X_2, dont les modalités sont $\mathscr{D}$ et $\mathscr{E}$) et dont la variable à prédire Y est booléenne.

Pour chaque variable prédictive, nous effectuons un comptage du nombre d'individus présents pour une modalité suivant la répartition des étiquettes. Les effectifs de ces tableaux de contingence sont présentés en tableau 2.3 pour la variable X_1 et en tableau 2.4 pour la variable X_2.

Si l'on souhaite comparer les individus ω_1 et ω_4 du tableau 2.2 dont les

$X_2 \backslash Y$	*Vrai*	*Faux*	Total
$\mathscr{D}$	2	3	5
$\mathscr{E}$	3	2	5

TAB. 2.4 – Répartition des individus suivant les modalités de X_2

profils sont respectivement $\{\mathscr{A}, \mathscr{D}\}$ et $\{\mathscr{B}, \mathscr{E}\}$, nous aurons comme valeur de différence $\delta^2_{X_1}(\omega_1, \omega_4) = 1/2$ comme indiqué par le calcul en 2.2.27 (en choisissant une valeur de $q = 2$).

$$\begin{aligned}
\delta^2_{X_1}(\omega_1, \omega_4) = & \sum_{e=1}^{r=2} \left| \frac{nb_{X_1}(\omega_1, e)}{nb_{X_1}(\omega_1)} - \frac{nb_{X_1}(\omega_2, e)}{nb_{X_1}(\omega_2)} \right|^2 \\
= & \left| \frac{nb_{X_1}(\omega_1, Vrai)}{nb_{X_1}(\omega_1)} - \frac{nb_{X_1}(\omega_2, Vrai)}{nb_{X_1}(\omega_2)} \right|^2 \\
+ & \left| \frac{nb_{X_1}(\omega_1, Faux)}{nb_{X_1}(\omega_1)} - \frac{nb_{X_1}(\omega_2, Faux)}{nb_{X_1}(\omega_2)} \right|^2 \\
= & \left| \frac{nb(\mathscr{A}, Vrai)}{nb(\mathscr{A})} - \frac{nb(\mathscr{B}, Vrai)}{nb(\mathscr{B})} \right|^2 + \left| \frac{nb(\mathscr{A}, Faux)}{nb_{\mathscr{A}}} - \frac{nb(\mathscr{B}, Faux)}{nb(\mathscr{B})} \right|^2 \\
= & \left| \frac{3}{3} - \frac{2}{4} \right|^2 + \left| \frac{0}{3} - \frac{2}{4} \right|^2 \\
= & \frac{1}{2}
\end{aligned} \tag{2.2.27}$$

En procédant de la même manière, nous obtenons $\delta^2_{X_2}(\omega_1, \omega_4) = 2/25$, d'où une différence globale $\delta(\omega_1, \omega_4) = \sqrt{29/50} \simeq 0,761$.

Si nous regardons à présent la distance existant entre l'individu ω_1 dont les modalités sont $\mathscr{A}$ et $\mathscr{D}$ et l'individu ω_8 dont les modalités sont $\mathscr{C}$ et $\mathscr{E}$, avec la métrique de différences de valeurs, nous obtenons cette fois-ci une distance $\delta(\omega_1, \omega_8) \simeq 1,442$.

Ainsi, alors que ω_4 et ω_8 sont tout deux des individus ayant des modalités différentes de l'exemple ω_1, nous observons que la *VDM* calculée entre ω_8 et ω_1 est bien plus importante que celle existant entre ω_4 et ω_1. Cette distance reflète mieux la différence de distribution selon la valeur de la variable à prédire : les étiquettes $Y(\omega_4)$ et $Y(\omega_1)$ sont identiques alors que l'étiquette $Y(\omega_8)$ diffère de $Y(\omega_1)$.

Nous signalons enfin que l'on peut trouver diverses adaptations de la métrique de différences de valeurs. Cost et Salzberg [CS93] proposent une pondération de cette mesure appelée *MVDM* pour "*Modified Value Difference Metric*". Le poids utilisé est propre à chaque individu et intervient dans la somme des différences sur chaque variable prédictive.

D'autres améliorations ont été apportées à la *VDM* par Wilson et Marti-

nez [WM97]. Ces auteurs ont défini en particulier une métrique de différences de valeurs applicable à des variables prédictives hétérogènes, appelée *HVDM* (*Heterogeneous Value Difference Metric*), qui résulte en fait d'une hybridation de la *VDM* de Stanfill et Waltz [SW86] et du calcul de distance sur des variables aussi bien numériques que catégorielles d'Aha *et al.* [AKA91].

Une deuxième métrique élaborée par Wilson et Martinez, appelée *IVDM* (*Interpolated Value Difference Metric*), procède par interpolation, les variables prédictives continues étant discrétisées en un ensemble d'intervalles.

Enfin, ces auteurs proposent aussi une métrique de différences de valeurs par fenêtrage, appelée *WVDM* (*Windowed Value Difference Metric*), qui est une variante de la *VDM* dans le cas où l'ordre des données dans l'échantillon a de l'importance (par exemple une suite de lettres dans un texte). Ainsi, au lieu de considérer l'échantillon de données de façon globale pour estimer la distribution des étiquettes, seule une fenêtre d'un nombre déterminé d'exemples, centrée sur l'individu spécifique α ou β, est prise en compte.

2.3 Similarité et dissimilarité entre individus

2.3.1 La similarité entre individus vue comme une fonction de la distance

2.3.1.1 Introduction

Un indice de similarité, que nous noterons Ⓢ, doit rendre compte de la ressemblance entre deux individus, propriété qui est estimée à travers la relation de proximité que ces deux individus entretiennent au sein de l'espace de représentation dans lequel ils se trouvent projetés. Dans le chapitre premier, nous avons donné en équation 1.2.1 (page 18) l'écriture d'une mesure de similarité entre deux individus α et β proposée par Aha, Kibler et Albert [AKA91] dans le cadre des algorithmes *IBL* d'apprentissage à base d'exemples. La formule de similarité était calculée comme étant l'opposée d'une mesure de distance tenant à la fois de la distance euclidienne standardisée pour les variables prédictives numériques et de la distance de Hamming pour les variables prédictives catégorielles.

Toutefois une telle mesure de similarité présente certains inconvénients. En effet, en dehors du cas de deux individus identiques où la distance, et donc la similarité, est nulle, la similarité calculée selon la formule 1.2.1 est toujours une valeur négative, la distance étant toujours une mesure positive. Il est ainsi plus judicieux de considérer une mesure de similarité donnant une

valeur maximale (voire infinie) pour deux mêmes individus, et une valeur nulle dans le cas où ces deux individus se situent à une distance maximale (voire infinie) au sein l'espace de représentation.

Nous observons ainsi qu'un ensemble de caractéristiques peuvent être recherchées pour avoir un indice de similarité Ⓢ adapté à certaines situations. Nous allons à présent indiquer quelles sont ces propriétés.

2.3.1.2 Propriétés d'un indice de similarité

Non-négativité Comme nous venons de l'indiquer, un indice de similarité se doit d'être un nombre non négatif :

$$Ⓢ(\alpha, \beta) \geqslant 0 \; \forall \alpha, \beta \in \Omega$$

Nous notons toutefois que dans le cas particulier où un indice de similarité est obtenu à travers des variables booléennes (ou catégorielles). Par exemple, l'indice *Q de Yule*, présenté dans la sous-section 2.2.3.5, peut avoir des valeurs négatives. Ainsi, une valeur de $Q_{\alpha,\beta} = -1$ indique une dissimilarité totale entre les individus α et β, c'est-à-dire qu'à chaque modalité positive de α correspond à une modalité négative de β et *vice versa*.

Symétrie Une autre propriété importante de l'indice de similarité est la symétrie : la valeur de Ⓢ ne doit pas dépendre de l'ordre de présentation des individus.

$$Ⓢ(\alpha, \beta) = Ⓢ(\beta, \alpha) \; \forall \alpha, \beta \in \Omega$$

Si cette propriété n'est pas vérifiée, il est éventuellement possible de la corriger en attribuant à Ⓢ la moyenne des indices de la similarité initiale non symétrique :

$$Ⓢ(\alpha, \beta) = \frac{1}{2} \left(\text{similarité}(\alpha, \beta) + \text{similarité}(\beta, \alpha) \right)$$

Normalisation La propriété de normalisation fixe une valeur maximale à l'indice de similarité Ⓢ. En règle générale, l'indice de similarité atteint son maximum, fixé arbitrairement à 1, lorsqu'un individu est comparé à lui-même :

$$Ⓢ(\alpha, \beta) = 1 \Leftrightarrow \alpha = \beta \; \forall \alpha, \beta \in \Omega$$

En outre, lorsque la similarité est maximale, la distance entre α et β est nulle :

$$\delta(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta \; \forall \alpha, \beta \in \Omega$$

Inégalités de l'indice de dissimilarité L'indice de dissimilarité, que nous considérons comme équivalent à une mesure de distance métrique, est caractérisé de plus par des propriétés d'inégalités. Nous notons cependant que ces propriétés d'inégalités ne sont pas vérifiées par un indice de similarité et que, de plus, il n'est généralement pas possible de transformer sans distorsion une matrice de similarité en une matrice de distance métrique.

Les inégalités, sous leur forme générale, sont données par l'équation 2.3.1.

$$\delta(\alpha,\beta) \leqslant \left[(\delta(\alpha,\gamma))^q + (\delta(\beta,\gamma))^q\right]^{\frac{1}{q}} \tag{2.3.1}$$

où q est un coefficient positif. Ainsi nous avons :

- l'inégalité triangulaire pour $q = 1$: $\delta(\alpha,\beta) \leqslant \delta(\alpha,\gamma) + \delta(\beta,\gamma)$;
- l'inégalité ultramétrique pour $q = \infty$: $\delta(\alpha,\beta) \leqslant max\left(\delta(\alpha,\gamma), \delta(\beta,\gamma)\right)$.

Distance arborée Une mesure de dissimilarité δ peut être une *distance arborée* dans le cas où elle satisfait à la condition des quatre points, appelée aussi *inégalité de Buneman*, décrite en équation 2.3.2.

$$\delta(\alpha,\beta) + \delta(\gamma,\epsilon) \leqslant \max\left(\delta(\alpha,\gamma) + \delta(\beta,\epsilon), \delta(\alpha,\epsilon) + \delta(\beta,\gamma)\right) \ \forall \alpha,\beta,\gamma,\epsilon \in \Omega \tag{2.3.2}$$

Le fait qu'une distance soit arborée est une propriété tout particulièrement recherchée dans des situations où il est souhaitable de pouvoir représenter les individus $\omega \in \Omega$ dans un espace plan, par exemple à travers la construction d'un arbre phylogénétique [BG88] ou pour représenter des distances subjectives en psychologie cognitive.

2.3.1.3 Indices de similarité et indices de dissimilarité

Lorsque l'on considère que la mesure de distance joue le rôle d'un indice de dissimilarité, il y a plusieurs façons d'obtenir un indice de similarité.

Considérer l'indice de similarité comme étant simplement l'opposé de la mesure de distance $\textcircled{S}(\alpha,\beta) = -\delta(\alpha,\beta)$ présente le désavantage de donner un indice n'ayant ni la propriété de non-négativité ni la propriété de normalisation.

Nous conseillons par conséquent d'utiliser la formule de calcul de l'indice de similarité donnée par l'équation 2.3.3 :

$$Ⓢ(\alpha, \beta) = \frac{1}{1 + \delta(\alpha, \beta)} \tag{2.3.3}$$

Avec cette formule, l'indice de similarité respecte les propriétés de non-négativité, de symétrie et de normalisation. Comme la distance δ prend ses valeurs entre 0 et $+\infty$, l'indice de similarité Ⓢ prend ses valeurs entre 0 (quand la distance tend vers l'infini) et 1 (quand la distance est nulle).

2.3.2 Matrice de dissimilarité

À partir des distances calculées entre les n_a individus de l'échantillon d'apprentissage, suivant l'une ou l'autre des métriques indiquées précédemment, il est possible de ranger les individus dans des tableaux de distance que nous noterons **D**. Ces matrices de distance ou de dissimilarité sont des matrices carrées symétriques d'ordre n_a dont la diagonale est nulle.

En fait, seule la matrice **D** nous est nécessaire lorsque nous souhaitons construire un graphe de voisinage comme le graphe des voisins relatifs de Toussaint [Tou80]. Cette propriété est très intéressante car, dans certaines situations, il arrive que nous puissions travailler à partir d'une matrice de dissimilarité (ou une matrice de similarité) mais que nous n'ayons pas à notre disposition toutes les informations de la matrice des données **X**.

Ces situations sont en effet assez fréquentes dans le domaine des sciences humaines et sociales. Il existe ainsi, en économie, les tableaux d'échanges interindustriels (TEI) de Leontief [Leo66]. Ces tableaux d'entrées-sorties décrivent l'interdépendance entre les différents secteurs de production et relient les facteurs de production (flux d'entrée) au produit (flux de sortie). La lecture du TEI en ligne indique ce que les secteurs vendent aux autres secteurs et la lecture en colonne ce que les différents secteurs ont acheté. Les informations présentes dans un tel tableau, une fois symétrisé, indiquent la force des échanges effectués, ce qui peut être considéré comme un indicateur de similarité entre les divers secteurs de production.

Nous pouvons également procéder de la sorte à partir de tableaux de données en psychologie ou en sociologie. De tels tableaux, assimilables à des matrices de dissimilarité ou de similarité selon les cas, se rencontrent notamment lorsque l'on cherche à modéliser et traiter les relations d'affinité liées à l'attirance ou à la répulsion qui peuvent se développer à l'intérieur d'un groupe de personnes comme des enfants à l'école ou des collègues dans une entreprise. Ces données peuvent être obtenues à partir d'études expérimentales, en demandant aux sujets d'indiquer quels sont leurs camarades de

classes ou leurs collègues les plus (ou les moins) appréciés suivant une échelle de valeur. Les matrices contenant ces scores doivent ensuite être symétrisées et adaptées (un score d'affinité pouvant être interprété comme un indicateur de similarité) pour obtenir des tableaux de dissimilarité.

2.4 Conclusion

Au cours de ce chapitre, nous avons présenté les principales manières de calculer des distances entre objets à partir de variables prédictives alors que celles-ci sont quantitatives, booléennes ou qualitatives. Outre les propriétés de ces mesures de distance, nous avons indiqué comment décrire la similarité – ou dissimilarité – existant entre des individus au moyen de ces distances calculées à partir des valeurs prises par leurs variables prédictives.

Par ailleurs, nous avons vu que des situations pouvaient se rencontrer où nous étions amenés à travailler à partir de matrices de dissimilarité sans avoir accès aux informations issues d'un ensemble de variables prédictives. Ainsi, à partir des matrices de dissimilarité, il est possible d'extraire de la connaissance sur la manière dont se comportent les individus, obtenant des groupes d'individus avec des méthodes de classification ou même des modèles prédictifs en associant les matrices de dissimilarité à une variable à prédire.

Dans les chapitres suivants, lorsqu'une mesure de distance sera employée (par exemple pour construire un graphe de voisinage), nous utiliserons la distance euclidienne standardisée, les variables qualitatives étant transformées sous forme disjonctive complète. Le choix d'une telle métrique de distance est motivé par le souci de disposer d'une distance facilement calculable et disposant de bonnes propriétés spatiales. Quant à la transformation des variables prédictives qualitatives sous forme disjonctive complète, elle est préférée au calcul de la distance *VDM* malgré toutes les qualités de cette dernière car nous présenterons des travaux donnant a priori des informations sur la séparabilité des étiquettes dans l'espace de représentation. Or, comme le calcul de la distance *VDM* est justement réalisé en tenant compte a priori de la répartition des étiquettes, cela aurait pour effet de biaiser nos résultats en les présentant sous un jour trop favorable.

Séparabilité des étiquettes et traitement des *outliers*

Sommaire

Chapitre 3

Séparabilité des étiquettes et traitement des *outliers*

Résumé

Nous présentons dans ce chapitre un test de séparabilité des étiquettes permettant d'estimer la qualité de l'espace de représentation lors de l'apprentissage supervisé d'une variable catégorielle. Ce test procède par l'étude de la distribution des étiquettes sur un graphe de voisinage et indique si le poids des arêtes qu'il faut couper dans ce graphe pour séparer des sommets d'étiquettes différentes relève ou non d'une répartition aléatoire.

Nous plaçant dans la problématique des bases d'apprentissage présentant des *outliers* dus à des erreurs d'étiquetage, nous proposons aussi dans ce chapitre une version locale de ce test afin d'identifier ces derniers pour les retirer des bases ou pour leur attribuer une nouvelle étiquette afin d'améliorer les performances en généralisation des méthodes d'apprentissage qui utilisent ces bases.

3.1 Introduction

Les méthodes d'apprentissage supervisé sont essentielles lors de la phase de fouille de données, étape elle-même fondamentale dans la démarche globale de l'extraction des connaissances à partir de données. Nous avons indiqué que ces méthodes essaient de construire un modèle de prédiction φ à partir d'un échantillon d'apprentissage Ω_a. Cependant, en raison de leurs modes de construction, ces modèles sont plus ou moins fiables.

La fiabilité des modèles prédictifs est généralement évaluée a posteriori sur un échantillon test Ω_t. Différents facteurs vont jouer sur cette fiabilité : la sélection des données de l'échantillon d'apprentissage, les hypothèses statistiques sous-jacentes aux modèles d'apprentissage, ainsi que les outils mathématiques employés pour réaliser l'apprentissage. Il arrive cependant qu'aucune méthode d'apprentissage ne puisse produire un modèle de prédiction fiable, en particulier lorsque les différentes étiquettes de la variables à prédire Y ne sont pas séparables dans l'espace de représentation $\mathbb{R}^p$.

Il est par conséquent important de disposer d'un indicateur capable de caractériser le degré de séparabilité des étiquettes d'une base d'apprentissage à partir d'un échantillon de celle-ci. Utilisant les propriétés des graphes de voisinage, nous fournissons un tel indicateur de séparabilité des étiquettes, appelé le « test du poids des arêtes coupées ». Cet indicateur va nous renseigner sur la facilité de séparer les différentes étiquettes d'un ensemble de données dans l'espace de représentation $\mathbb{R}^p$ et, par voie de conséquence, apportera une information a priori sur l'aptitude qu'auront des méthodes d'apprentissage supervisé à fournir un modèle fiable de la variable à prédire dans la problématique d'apprentissage étudiée.

Par ailleurs, lorsque les étiquettes d'une base de données sont bien séparables, des individus ne doivent normalement pas se retrouver dans des régions de l'espace de représentation présentant d'autres étiquettes que la leur. Nous proposons ainsi une manière originale d'utiliser les propriétés locales du voisinage et du poids des arêtes coupées d'un exemple de l'échantillon d'apprentissage pour repérer les *outliers*. Par *outliers*, nous entendons des exemples de la base de données situés hors de leur place attendue au sein de l'espace de représentation $\mathbb{R}^p$.

Nous situant dans le cas particulier où l'échantillon d'apprentissage est supposé contenir du bruit sur la variable à prédire, c'est-à-dire quand des erreurs d'étiquetage portent sur des individus de la base, nous proposons une méthode de filtrage ainsi qu'une méthode de réétiquetage/suppression reposant sur la détection et le traitement de ces *outliers*.

3.2 Séparabilité des étiquettes et poids des arêtes coupées

3.2.1 Introduction

Nous avons présenté plus haut l'intérêt de disposer d'un indicateur de séparabilité des étiquettes. Un tel indicateur n'est pas sans lien avec l'apprenabilité d'un échantillon d'apprentissage. Il est vrai que des mesures d'apprenabilité existent déjà par ailleurs, telles que la « VC-dimension » fournie par la théorie de l'apprentissage statistique [Vap98]. Cependant la VC-dimension est difficile à calculer dans de nombreux cas.

Le problème de l'apprenabilité a également été étudié à travers une approche statistique par Rao [Rao65]. Dans le cas d'une distribution normale des étiquettes, Rao a mesuré le degré d'apprenabilité à travers un test reposant sur l'homogénéité de la population. De façon assez similaire, Kruskall et Wallis ont défini un test non paramétrique se fondant sur l'hypothèse d'égalité entre les paramètres d'échelle [AEM86].

Plus récemment, Sebban et Zighed [Seb96, SZ96] ont proposé un test lié au nombre d'arêtes reliant des exemples d'étiquettes différentes dans un graphe de voisinage. Ce test nécessite la construction d'une structure de voisinage telle que le graphe des voisins relatifs de Toussaint [Tou80]. Certaines arêtes du graphe sont coupées afin de n'avoir plus que des amas de points de la même étiquette. Les auteurs ont établi la proportion d'arêtes qui doivent être retirées sous l'hypothèse nulle H_0 d'une distribution aléatoire des étiquettes. Avec cette loi, ils ont été en mesure d'indiquer si les étiquettes sont séparables ou non à partir de la probabilité d'avoir une valeur calculée aussi importante que celle observée sous l'hypothèse nulle.

Nous avons poursuivi ces travaux en proposant un cadre théorique plus général dans lequel nous avons développé une statistique de test non paramétrique qui tienne compte des poids des arêtes coupées [ZLM01, ZLM02]. Pour réaliser ceci, nous nous sommes inspirés des travaux sur l'auto-corrélation spatiale, en particulier la statistique "*join-counts*" présentée par Cliff et Ord [CO86].

3.2.2 Graphe de voisinage et amas

La capacité d'apprentissage d'une méthode est fortement associée au degré de séparabilité des étiquettes. Or les étiquettes sont d'autant plus facilement séparables que, d'une part, les exemples d'une même étiquette

apparaissent sous forme de groupes dans une région donnée de l'espace de représentation $\mathbb{R}^p$, que, d'autre part, le nombre de ces groupes est petit (ce nombre est au minimum est égal à r, le nombre d'étiquettes différentes de la variable à prédire Y), et enfin que les frontières entre ces divers groupes sont simples.

Pour rendre compte de la proximité entre les exemples, nous utilisons les graphes de voisinage que nous avons présenté dans le premier chapitre, plus particulièrement le graphe des voisins relatifs de Toussaint [Tou80].

Suivant Sebban [Seb96], nous définissons la notion d'amas (*cf.* équation 3.2.1) pour exprimer le fait qu'un groupe de points voisins sont de même étiquette.

Définition 3.2.1 Amas. *Un amas* $\aleph$ *est un sous-graphe connexe du graphe de voisinage* G *dont tous les sommets* $\Sigma_\aleph$ *(avec* $\Sigma_\aleph \in \Sigma$*) sont de la même étiquette.*

$$\aleph \textit{ est un amas } \Leftrightarrow \{\{\aleph \textit{ est connexe}\} \wedge \{\forall(\omega_i, \omega_j) \in \Sigma_\aleph^2, Y(\omega_i) = Y(\omega_j)\}\} \tag{3.2.1}$$

La construction des amas qui nous permettra de caractériser la structure des points dans l'espace $\mathbb{R}^p$ est donc réalisée en deux temps : nous effectuons d'abord la construction d'un graphe de voisinage puis nous procédons à la coupure des arêtes reliant des points d'étiquettes différentes.

Sur la figure 3.1(a), nous devons couper cinq arêtes pour séparer les points qui ont une étiquette blanche de ceux qui ont une étiquette grise. La coupure produit alors trois amas représentés sur la figure 3.1(b).

Le nombre d'amas $N_\aleph$ obtenus par ce procédé nous renseigne partiellement sur la séparabilité des étiquettes de la base d'apprentissage. Si ce nombre a tendance à être faible, proche du nombre total d'étiquettes différentes r, les étiquettes sont bien séparables et nous pouvons supposer qu'il est possible de trouver une méthode d'apprentissage capable de rendre compte du modèle sous-tendant l'organisation particulière de ces données. Au contraire, si le nombre d'amas est élevé, comme cela est le cas dans la situation d'une distribution aléatoire des étiquettes, il n'est pas possible de trouver de modèle capable de rendre compte de la structure des données.

Ce constat est toutefois à nuancer car il existe des situations où le nombre d'amas ne permet pas de caractériser certaines organisations particulières des données qui semblent pourtant intuitivement bien différentes. En effet, un

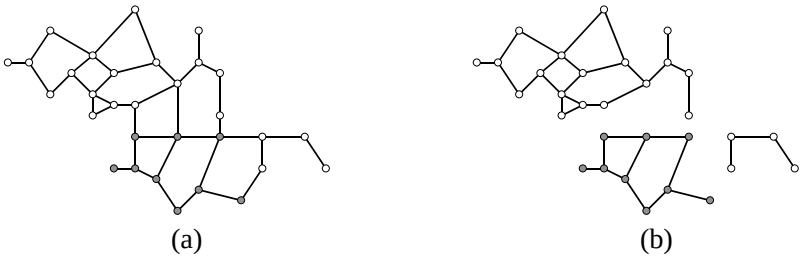

FIG. 3.1 – Coupure d'arêtes et construction des amas

même nombre d'amas peut être obtenu aussi bien dans des situations où les différents amas sont bien distants les uns des autres et se retrouvent isolés dans l'espace de représentation (donc les étiquettes sont aisément séparables) que dans des situations où les amas d'étiquettes différentes sont très proches. Par conséquent, nous nous intéresserons davantage au nombre d'arêtes qu'il sera nécessaire de couper pour isoler les amas, arêtes auxquelles nous attribuerons un poids qui sera défini par la suite.

3.2.3 Cadre statistique

Pour faire un parallèle entre l'apprentissage supervisé et l'analyse spatiale, nous considérons un graphe de contiguïté spatiale jouant le rôle d'un graphe de voisinage [CO86]. Suivant Cliff et Ord, les arêtes du graphe prennent une couleur donnée en fonction de leurs étiquettes. Nous noterons r le nombre de couleurs possibles, c'est-à-dire le nombre d'étiquettes différentes. Nous allons à la fois chercher à décrire le lien existant entre les sommets d'un graphe qui sont de la même couleur et chercher à tester l'hypothèse de non significativité. Procédant de cette manière, cette recherche se traduit par le test de l'hypothèse d'absence d'auto-corrélation spatiale entre les valeurs prises par une variable catégorielle à travers des unités spatiales.

Dans le cas d'un graphe de voisinage, cela revient à tester l'hypothèse que la variable à prédire Y ne peut pas être apprise au moyen de méthodes d'apprentissage à base d'exemples.

Notations	Définitions	Poids : connexion simple
$\sum_2 w_{i,j}$	$\sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} w_{i,j}$	$2a$
S_0	$\sum_2 w_{i,j}$	$2a$
S_1	$\frac{1}{2} \sum_2 (w_{i,j} + w_{j,i})^2$	$4a$
S_2	$\sum_{i=1}^{n} (w_{i+} + w_{+i})^2$	$4 \sum_{i=1}^{n} v_{i+}^2$

TAB. 3.1 – Simplifications proposées par Cliff et Ord

3.2.3.1 Notations et abréviations

Soient i, j, k ou l des sommets du graphe de voisinage G, n le nombre de sommets de G ($n = Card(\Sigma)$) et a le nombre d'arêtes du graphe ($a = Card(A)$).

Nous appelons V la matrice de connexité entre les différents points de G. La matrice $V = \{v_{i,j}, \forall \{i,j\} \in \Sigma^2\}$ est booléenne et prend ses valeurs dans $\{0\,;1\}$, plus précisément $v_{i,j}$ prend la valeur 1 si l'arête $(i,j) \in A$ et 0 sinon.

Nous notons W la matrice des poids, $W = \{w_{i,j}, \forall \{i,j\} \in \Sigma^2\}$ où $w_{i,j}$ est le poids de l'arête (i,j).

Nous notons w_{i+} la somme de la ligne i et w_{+j} la somme de colonne j.

Nous supposons que la matrice W est symétrique. Dans le cas contraire ($w_{i,j} \neq w_{j,i}$), nous appelons W' cette matrice de poids. Nous obtenons alors une matrice de poids W symétrique à partir de W' sans perte de généralité en calculant $w_{i,j} = \frac{1}{2}\left(w'_{i,j} + w'_{j,i}\right)$.

Nous appelons π_e la proportion des sommets ayant l'étiquette e parmi les r étiquettes possibles.

Suivant Cliff et Ord [CO86], nous adoptons les notations simplificatrices évoquées en tableau 3.1. Dans ce tableau, en troisième colonne, nous indiquons le cas particulier où la pondération de la matrice W se résume à la simple connexité : $w_{i,j} = v_{i,j}$.

3.2.3.2 Définition de la statistique du poids des arêtes coupées

Pour prendre en compte une éventuelle pondération des arêtes, nous travaillons avec la matrice des poids symétriques W. Nous notons différents types de poids pour W :

- la connexion simple : $w_{i,j} = v_{i,j}$, soit 1 quand les points i et j sont reliés, et 0 sinon ;
- la distance : $w_{i,j} = (1 + \delta_{i,j})^{-1}$;

– le rang : $w'_{i,j} = \frac{1}{rang(j)}$, où $rang(j)$ est le numéro du point j dans le voisinage du point i.

Nous faisons remarquer que les deux premiers poids sont symétriques. Quant au troisième, il est rendu symétrique en calculant $w_{i,j} = \frac{1}{2}\left(w'_{i,j} + w'_{j,i}\right)$.

Les arêtes reliant des points de même étiquette sont distinguées des arêtes retirées lors de la constitution des amas. Nous notons donc I_e la somme des poids des arêtes reliant des sommets de même étiquette e et $J_{e,e'}$ la somme des poids des arêtes reliant des sommets d'étiquette e reliées aux sommets d'étiquette e'. Les statistiques du poids des arêtes non coupées I et du poids des arêtes coupées J sont définies respectivement par les équations 3.2.2 et 3.2.3.

$$I = \sum_{e=1}^{r} I_e \tag{3.2.2}$$

$$J = \sum_{e=1}^{r-1} \sum_{e'=e+1}^{r} J_{e,e'} \tag{3.2.3}$$

Dans la mesure où les statistiques I et J sont reliées à travers la propriété $I + J = \frac{1}{2}S_0$, il n'est utile d'étudier que la statistique J ou sa forme normalisée $\frac{J}{I+J} = \frac{2J}{S_0}$. Les deux valeurs sont en effet identiques après centrage et réduction.

Tout comme Jain et Dubes [JD88], nous considérons un échantillon binomial dans lequel l'hypothèse nulle H_0 est définie de la manière suivante : les sommets du graphe sont étiquetés indépendamment l'un de l'autre, avec la même probabilité de distribution π_e, où π_e indique la probabilité d'avoir l'étiquette e parmi les r étiquettes différentes. Nous pouvons également considérer un échantillon hypergéométrique par ajout à l'hypothèse nulle de la contrainte d'avoir n_e sommets ayant l'étiquette e.

Le rejet de l'hypothèse nulle signifie soit que les étiquettes ne sont pas distribuées de manière indépendante sur les sommets du graphe soit que la probabilité de distribution des étiquettes n'est pas la même pour les différents sommets.

3.2.3.3 Distribution de I et de J sous l'hypothèse nulle

Afin de tester l'hypothèse nulle H_0 avec la statistique J, nous utilisons un test bilatéral pour indiquer l'obtention, pour J, soit d'une valeur anormale-

ment petite (signe d'une grande séparabilité des étiquettes) soit d'une valeur anormalement grande (signe de la présence d'un motif ou d'une structuration particulière).

Nous établissons la distribution de J sous l'hypothèse nulle H_0 en calculant le risque critique (ou "*p-value*") associé à la valeur observée J et si celui-ci est inférieur à un seuil de significativité fixé α_0 (que nous avons choisi égal à 5%), nous rejetterons l'hypothèse nulle compte tenu de cette observation. Ce calcul peut également être réalisé par simulation ou par approximation normale. Dans ce dernier cas, nous devons calculer la moyenne et la variance de J sous H_0.

3.2.3.4 Cas booléen

Dans le cas particulier d'une situation booléenne, les deux étiquettes définies par Y sont notée 1 et 2. D'après Moran [Mor48], $U_i = 1$ si l'étiquette du $i^{\text{ème}}$ sommet est 1 et $U_i = 0$ si cette étiquette est 2, avec $i = 1, 2, ..., n$.

Nous appelons π_1 la proportion de points ayant l'étiquette 1 et π_2 la proportion de points d'étiquette 2. La valeur de J est donnée par la formule 3.2.4.

$$J_{1,2} = \frac{1}{2} \sum_{2} w_{i,j} \left(U_i - U_j\right)^2 = \frac{1}{2} \sum_{2} w_{i,j} Z_{i,j} \tag{3.2.4}$$

Dans l'équation 3.2.4, les U_i sont indépendamment distribués selon une loi de Bernoulli de paramètre π_1, notée $B(1, \pi_1)$.

Nous pouvons remarquer que les variables $Z_{i,j} = \left(U_i - U_j\right)^2$ se répartissent selon la distribution $B(1, \pi_1\pi_2)$ mais ne sont pas indépendantes. En fait, les covariances $Cov(Z_{i,j}, Z_{k,l})$ sont nulles seulement si les quatre indices sont différents. S'il y a au moins deux indices identiques, nous obtenons l'expression indiquée en équation 3.2.5.

$$Cov(Z_{i,j}, Z_{i,l}) = \pi_1\pi_2(1 - 4\pi_1\pi_2) \tag{3.2.5}$$

Le tableau 3.2 résume les différents résultats concernant les espérances et variances associées à la statistique $J_{1,2}$.

Le risque critique de $J_{1,2}$ est calculé à partir de la distribution normale standardisée par centrage et réduction de la valeur observée. Les valeurs critiques de $J_{1,2}$ au seuil de significativité α_0 sont données par les formules algébriques 3.2.6 et 3.2.7.

Variable	Espérance	Variance
U_i	π_1	$\pi_1\pi_2$
$Z_{i,j} = (U_i - U_j)^2$	$2\pi_1\pi_2$	$2\pi_1\pi_2\left(1 - 2\pi_1\pi_2\right)$
$J_{1,2}$	$S_0\pi_1\pi_2$	$S_1\pi_1^2\pi_2^2 + S_2\pi_1\pi_2\left(\frac{1}{4} - \pi_1\pi_2\right)$
$J_{1,2}$ si $w_{i,j} = v_{i,j}$	$2a\pi_1\pi_2$	$4a\pi_1^2\pi_2^2 + \pi_1\pi_2\left(1 - 4\pi_1\pi_2\right)\sum_{i=1}^{n} v_{i+}^2$

TAB. 3.2 – Espérances et variances de la statistique $J_{1,2}$

$$J_{1,2;\alpha_{0/2}} = S_0\pi_1\pi_2 - u_{1-\alpha_{0/2}}\sqrt{S_1\pi_1^2\pi_2^2 + S_2\pi_1\pi_2\left(\frac{1}{4} - \pi_1\pi_2\right)} \tag{3.2.6}$$

$$J_{1,2;1-\alpha_{0/2}} = S_0\pi_1\pi_2 + u_{1-\alpha_{0/2}}\sqrt{S_1\pi_1^2\pi_2^2 + S_2\pi_1\pi_2\left(\frac{1}{4} - \pi_1\pi_2\right)} \tag{3.2.7}$$

3.2.3.5 Cas de plusieurs étiquettes

Pour généraliser ces résultats au cas où la variable Y présente r et non plus seulement deux étiquettes, suivant Cliff et Ord [CO86], nous raisonnons à partir des statistiques I et J définies précédemment.

Ces statistiques sont données par les équations 3.2.8 et 3.2.9.

$$I = \sum_{e=1}^{r} I_e = \frac{1}{2}\sum_{2} w_{i,j}T_{i,j} \tag{3.2.8}$$

$$J = \sum_{e=1}^{r-1}\sum_{e'=e+1}^{r} J_{e,e'} = \frac{1}{2}\sum_{2} w_{i,j}Z_{i,j} \tag{3.2.9}$$

où, dans formules algébriques 3.2.8 et 3.2.9, $T_{i,j}$ et $Z_{i,j}$ sont deux variables aléatoires booléennes indiquant si les sommets i et j ont la même étiquette ($T_{i,j}$) ou non ($Z_{i,j}$).

Des résultats précédents, nous déduisons aisément les espérances des statistiques I et J, notées respectivement E_I et E_J, et dont le calcul est indiqué par les formules 3.2.10 et 3.2.11.

$$E_I = \frac{1}{2}S_0\sum_{e=1}^{r}\pi_e^2 \tag{3.2.10}$$

$$E_J = S_0 \sum_{e=1}^{r-1} \sum_{e'=e+1}^{r} \pi_e \pi_{e'} \tag{3.2.11}$$

Comme I et J sont reliées par la relation $I + J = \frac{1}{2} S_0$, ces deux variables ont la même variance, notée $\sigma^2 = Var(I) = Var(J)$. Nous faisons remarquer qu'obtenir la valeur de la variance σ^2 est plus complexe dans ce cas car il est nécessaire de prendre en considération les covariances dans la formule de calcul. En s'aidant de la démarche suivie par Cliff et Ord [CO86], nous obtenons pour le schéma binomial le résultat présenté en équation 3.2.12.

$$4\sigma^2 = \left\{ \begin{array}{lll} & S_2 \sum_{e=1}^{r-1} \sum_{e'=e+1}^{r} \pi_e \pi_{e'} & \\ + & (2S_1 - 5S_2) \sum_{e=1}^{r-2} \sum_{e'=e+1}^{r-1} \sum_{e''=e'+1}^{r} \pi_e \pi_{e'} \pi_{e''} & \\ + & 4\,(S_1 - S_2) \left[\begin{array}{ll} & \sum_{e=1}^{r-1} \sum_{e'=e+1}^{r} \pi_e^2 \pi_{e'}^2 \\ - & 2 \sum_{e=1}^{r-3} \sum_{e'=e+1}^{r-2} \sum_{e''=e'+1}^{r-1} \sum_{e'''=e''+1}^{r} \pi_e \pi_{e'} \pi_{e''} \pi_{e'''} \end{array} \right] & \end{array} \right. \tag{3.2.12}$$

3.2.4 Complexité algorithmique du test

Différentes étapes sont à prendre en compte pour l'évaluation de la complexité algorithmique du test. Tout d'abord, le calcul de la matrice de distance (nécessaire à la création du graphe de voisinage) est réalisé en $O(p \times n^2)$, avec n le nombre d'exemples (ou nombre de sommets du graphe) et p le nombre de variables prédictives. Ensuite, la construction du graphe de voisinage dans $\mathbb{R}^p$ se fait en $O(n^3)$ pour le graphe des voisins relatifs de Toussaint [Tou80]. Enfin, le calcul des espérances et variances des statistiques de I et J se déroule en $O(n^2)$ (pour le calcul de S_0 ou S_1 par exemple) ou en $O(r^3)$, avec r représentant le nombre d'étiquettes différentes pour Y (élément nécessaire au calcul de la variance donnée en équation 3.2.12). Étant donné que le nombre de variables prédictives p et que le nombre d'étiquettes r sont généralement très petits comparés au nombre d'exemples n, le test est globalement en $O(n^3)$.

Par ailleurs, il important de savoir que la base d'apprentissage complète n'est pas nécessaire pour la réalisation du test. Un échantillon représentatif de cette base peut suffire pour fournir une bonne idée de la séparabilité des étiquettes de cette base de données.

Nom de la base	n	p	r	$N_{\aleph}$	a
Breast cancer Wisconsin	683	9	2	10	7562
Bupa liver disorders	345	6	2	50	581
Glass Identification	214	9	6	52	275
Haberman's survival	306	3	2	47	517
Image segmentation	210	19	7	27	268
Ionosphere	351	34	2	43	402
Iris plants	150	4	3	6	196
Iris (Bezdek)	150	4	3	6	189
Musk "Clean 1"	476	166	2	14	810
Pima Indians diabetes	768	8	2	82	1416
Waveform (Breiman)	1000	21	3	49	2443
Wine recognition	178	13	3	9	281
Yeast (protein localization)	1484	8	10	401	2805

TAB. 3.3 – Informations générales sur les 13 bases

3.2.5 Évaluation expérimentale de la statistique du poids des arêtes coupées

3.2.5.1 Expérimentations principales

Nous avons étudié la statistique du poids des arêtes coupées sur 13 jeux d'essai du site d'apprentissage automatique de l'Université de Californie à Irvine (*UCI Machine Learning Repository*) [BM98]. Les bases de données utilisées ont été choisies afin de n'avoir que des variables prédictives numériques et une variable à prédire catégorielle. Pour chaque base, nous construisons un graphe de voisinage (le graphe des voisins relatifs de Toussaint) sur les n individus de la base. Sur le tableau 3.3, n indique le nombre d'individus de la base, p le nombre de variables prédictives et r le nombre d'étiquettes différentes de la variable à prédire Y, $N_{\aleph}$ indique le nombre d'amas obtenus après avoir coupé les arêtes reliant des sommets d'étiquettes différentes et a le nombre d'arêtes obtenues pour la construction du graphe.

Sur le tableau 3.4, pour chacune des bases, nous indiquons la valeur relative du poids des arêtes coupées $\frac{J}{I+J}$, la valeur centrée et réduite de la statistique du poids des arêtes coupées J^{CR} ainsi que son risque critique (ou *p-value*) pour chacune des trois pondérations :

1. lorsque le poids $w_{i,j}$ se résume à la simple connexion, c'est-à-dire qu'il vaut 1 si les sommets i et j sont reliés par une arête ;

Base	Poids = Connexion			Poids = Distance			Poids = Rang		
	$\frac{J}{I+J}$	J^{CR}	*p-value*	$\frac{J}{I+J}$	J^{CR}	*p-value*	$\frac{J}{I+J}$	J^{CR}	*p-value*
Breast	0,008	-25,29	0	0,003	-24,38	0	0,014	-25,02	0
BUPA	0,401	-3,89	$1{,}01\text{E}^{-4}$	0,385	-4,33	$1{,}48\text{E}^{-05}$	0,394	-4,08	$4{,}58\text{E}^{-05}$
Glass	0,356	-12,63	0	0,315	-12,90	0	0,342	-12,93	0
Haberman	0,331	-1,92	0,0544	0,321	-2,20	0,0276	0,331	-1,90	0,0578
Image	0,224	-29,63	0	0,141	-29,31	0	0,201	-29,88	0
Ionosphere	0,137	-11,34	0	0,046	-11,07	0	0,136	-11,33	0
Iris plants	0,087	-17,22	0	0,074	-17,41	0	0,076	-17,14	0
Iris Bezdek	0,090	-16,82	0	0,077	-17,01	0	0,078	-16,78	0
Musk 1	0,167	-17,53	0	0,115	-7,69	$1{,}51\text{E}^{-14}$	0,143	-18,10	0
Pima	0,310	-8,74	$2{,}38\text{E}^{-18}$	0,282	-9,86	0	0,305	-8,93	$4{,}34\text{E}^{-19}$
Waveform	0,255	-42,75	0	0,248	-42,55	0	0,248	-42,55	0
Wine	0,093	-19,32	0	0,054	-19,40	0	0,074	-19,27	0
Yeast	0,524	-27,03	0	0,512	-27,18	0	0,509	-28,06	0

TAB. 3.4 – Valeurs de la statistique de test sur les 13 bases

2. lorsque le poids $w_{i,j}$ est associé à la distance, c'est-à-dire l'inverse de la longueur de l'arête (i, j) ;
3. lorsque le poids $w_{i,j}$ est associé au rang du sommet i ou j dans le voisinage de l'autre sommet.

Pour chacune des bases et chacun des modes de pondération, nous remarquons dans le tableau 3.4 que les risques critiques sont très faibles, ce qui indique que l'hypothèse nulle d'une distribution aléatoire des étiquettes sur les sommets d'un graphe de voisinage est très forte.

À titre indicatif, avec un PC à 450 MHz, le temps CPU mis pour le calcul des tests (soit le calcul de la matrice de distance, la construction du graphe, la procédure de coupure des arêtes et le calcul des statistiques de test) est compris entre une seconde (pour les bases *Iris* comprenant 150 individus) et 200 secondes (pour la base *Yeast* avec 1 500 individus). Nous signalons que les tests présentés ne concernent que le graphe des voisins relatifs de Toussaint, les résultats obtenus avec un graphe de Gabriel ou un arbre recouvrant minimal sont assez proches de ceux indiqués pour le graphe des voisins relatifs.

3.2.5.2 Sensibilité du test au bruit sur l'étiquette

Afin d'étudier la sensibilité du test du poids des arêtes coupées au bruit, nous avons procédé à un retournement aléatoire d'étiquettes d'une des 13 bases présentées : le jeu de données *Iris Plants Database*, une base de référence facile à apprendre.

Les résultats reportés en figure 3.2 présentent les valeurs obtenues en changeant progressivement l'étiquette initiale des exemples de la base des Iris pour les trois modes de pondération de la statistique de test : sur le

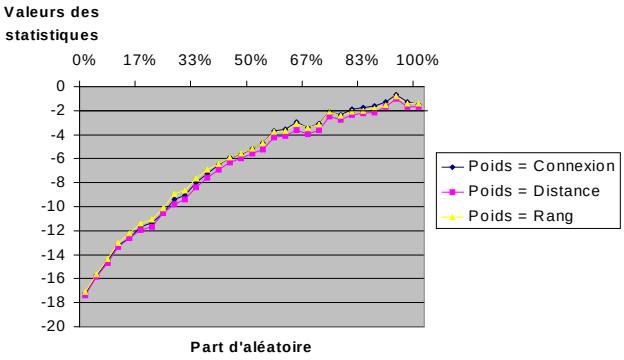

FIG. 3.2 – Sensibilité de la statistique de test au bruit sur l'étiquette

graphique, 0% correspond à la base Iris originelle et 100% correspond à une base entièrement aléatoire.

À partir de la statistique du poids des arêtes coupées, nous avons aussi pu calculer le risque critique de l'hypothèse nulle (répartition aléatoire des données). L'évolution du risque critique en fonction de l'aléatoire introduit dans la base d'apprentissage est représentée en figure 3.3. Le risque critique passe la barre du seuil de significativité, suivant les tests, entre 20 et 30 étiquettes retournées sur les 150 étiquettes totales, soit 80 à 90% d'aléatoire. Le test où la pondération revient simplement à la connexion est le plus sensible à la présence de bruit parmi les trois modes de pondération, le test des arêtes pondérées par la distance étant le moins sensible. Les trois tests présentent une forte robustesse au bruit car le risque critique garde une valeur nulle ou infime en-deçà de deux tiers de retournements d'étiquettes. Ce phénomène s'explique par le fait que l'hypothèse de répartition aléatoire des étiquettes sur les sommets d'un graphe de voisinage connexe est très forte.

3.2.5.3 Poids des arêtes coupées et taux d'erreur en apprentissage

Les 13 jeux d'essai ont été testés sur les méthodes d'apprentissage supervisé suivantes :

- une méthode d'apprentissage à base d'exemples : le *plus proche voisin* (PPV) ;
- un arbre de décision : *C4.5* ;

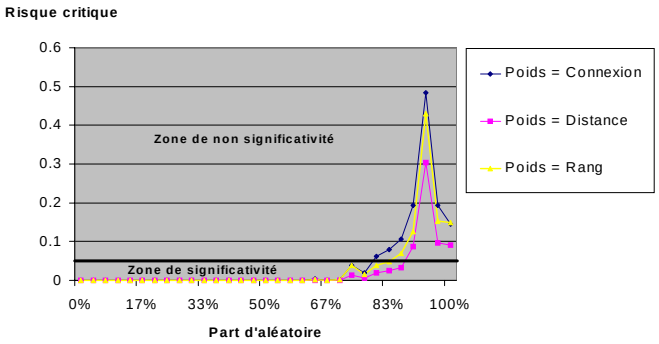

FIG. 3.3 – Significativité du test en fonction du bruit sur l'étiquette

Base	$\frac{J}{I+J}$	J^{CR}	*p-value*	PPV	C4.5	Sipina	Perc.	MLP	B. Naïf	Moyenne
Breast	0,008	-25,29	0	0,041	0,059	0,050	0,032	0,032	0,026	0,040
BUPA	0,401	-3,89	0,0001	0,363	0,369	0,347	0,305	0,322	0,380	0,348
Glass	0,356	-12,63	0	0,317	0,289	0,304	0,350	0,448	0,401	0,352
Haberman	0,331	-1,92	0,0544	0,326	0,310	0,294	0,241	0,275	0,284	0,288
Image	0,224	-29,63	0	0,124	0,124	0,152	0,119	0,114	0,605	0,206
Ionosphere	0,137	-11,34	0	0,140	0,074	0,114	0,128	0,131	0,160	0,124
Iris plants	0,087	-17,22	0	0,060	0,033	0,053	0,067	0,040	0,080	0,056
Iris (Bezdek)	0,090	-16,82	0	0,053	0,060	0,067	0,060	0,053	0,087	0,063
Musk 1	0,167	-17,53	0	0,065	0,162	0,232	0,187	0,113	0,227	0,164
Pima	0,310	-8,74	$2{,}4\text{E}^{-18}$	0,288	0,283	0,270	0,231	0,266	0,259	0,266
Waveform	0,255	-42,75	0	0,186	0,260	0,251	0,173	0,169	0,243	0,214
Wine	0,093	-19,32	0	0,039	0,062	0,073	0,011	0,017	0,186	0,065
Yeast	0,524	-27,03	0	0,455	0,445	0,437	0,447	0,446	0,435	0,444

TAB. 3.5 – Taux d'erreur et valeurs de la statistique de test sur les 13 bases

- un graphe d'induction : *Sipina* ;
- un réseau de neurones simple : le *Perceptron* (Perc.) ;
- un réseau de neurones multicouche avec 10 neurones en couche cachée (MLP) ;
- un modèle *bayésien naïf* (B. Naïf).

Les ouvrages auxquels nous pourrons nous référer pour ces différentes méthodes d'apprentissage sont celui de Quinlan [Qui93b] pour la méthode *C4.5*, celui de Zighed et Rakotomalala [ZR00] pour la méthode *Sipina* et celui de Mitchell [Mit97] pour les autres méthodes d'apprentissage.

Nous indiquons dans le tableau 3.5 les taux d'erreur obtenus pour nos 13 jeux d'essai avec ces méthodes à partir d'une validation croisée en 10 parties (les $9/10^{\text{èmes}}$ de la base sont retenus pour l'apprentissage et le test est effectué sur le dernier dixième, l'opération étant répétée pour chacune des 10 parties) et les informations affichées dans le tableau 3.5 sont les moyennes de ces 10

	PPV	C4.5	Sipina	Perc.	MLP	B. Naïf	Moyenne
Moyenne	0,189	0,195	0,203	0,181	0,187	0,259	0,202
$R^2(\frac{J}{I+J}$; tx err)	0,933	0,934	0,937	0,912	0,877	0,528	0,979
$R^2(J^{CR}$; tx err)	0,076	0,020	0,019	0,036	0,063	0,005	0,026

TAB. 3.6 – Corrélation entre les taux d'erreur et la statistique de test

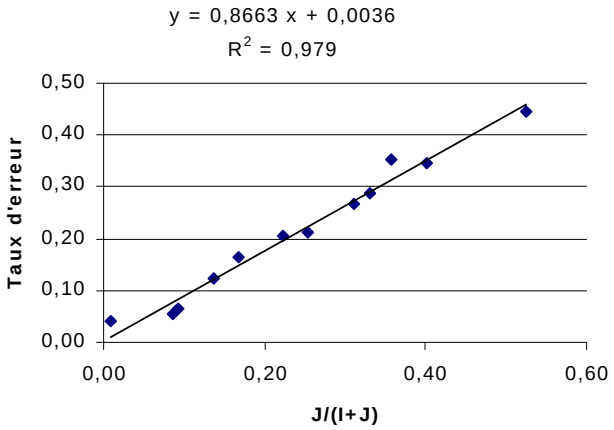

FIG. 3.4 – Statistique du poids des arêtes coupées relative et taux d'erreur

taux d'erreur. Sur le tableau 3.6, nous observons que les taux d'erreurs des différentes méthodes, et plus particulièrement de la moyenne du taux d'erreur de chacune des méthodes, sont bien corrélés avec la statistique relative du poids des arêtes coupées. Cette corrélation est également mise en évidence sur la figure 3.4 qui indique la relation linéaire existant entre la statistique du poids des arêtes coupées relative $J/(I + J)$ et la moyenne des taux d'erreur pour les 13 bases d'apprentissage.

3.2.5.4 Statistique pour des variables prédictives catégorielles

Afin de montrer qu'il est également possible de traiter des bases de données dont certaines variables prédictives, voire toutes, sont de nature catégorielle, nous avons appliqué le test du poids des arêtes coupées sur la base *Flag* également présente sur le site de l'UCI [BM98]. Les différentes variables prédictives ont été réécrites sous forme de variables booléennes par une trans-

n	p	r	$N_{\aleph}$	a	$\frac{J}{I+J}$	J^{CR}	*p-value*
194	67	6	46	327	0,489	-13,91	0

PPV	C4.5	Sipina	Perc.	MLP	B. Naïf	Moyenne
0,366	0,346	0,371	0,310	0,428	0,340	0,360

TAB. 3.7 – Statistique de test et taux d'erreur pour la base *Flag*

n	p	r	$N_{\aleph}$	a	$\frac{J}{I+J}$	J^{CR}	*p-value*
20	21	3	6	25	0,400	-0,44	$6{,}64\text{E}^{-01}$
50	21	3	11	72	0,375	-4,05	$5{,}02\text{E}^{-05}$
100	21	3	12	156	0,301	-8,44	$3{,}30\text{E}^{-17}$
1000	21	3	49	2443	0,255	-42,75	0

TAB. 3.8 – Statistique de test pour différentes tailles de la base *Waves*

formation sous forme disjonctive complète (*cf.* sous-section 2.2.4.3, page 58). Le graphe de voisinage a été construit à partir des variables centrées et réduites obtenues à partir de ces variables binaires. Sur le tableau 3.7, le test indique que les étiquettes de la base *Flag* sont séparables (*p-value* < 0.05), ce qui peut être rapporté au taux d'erreur moyen de cette base qui est de 36% pour l'apprentissage d'une variable présentant 6 étiquettes différentes.

3.2.5.5 Effet de la taille de la base de données

Nous faisons remarquer que la statistique du poids des arêtes coupées normalisée J^{CR} et que son risque critique associé dépendent fortement de la taille de la base d'apprentissage. La même variance observée sous l'hypothèse nulle est ainsi plus significative en raison de la taille de la base d'apprentissage. Cet effet est mis en évidence sur le tableau 3.8 qui présente la base d'apprentissage *Waves* avec différentes tailles (n = 20, 50, 100, 1000). Le risque critique n'est pas significatif pour $n = 20$ (car supérieur à 0.05) mais il le devient très rapidement lorsque la taille de l'échantillon augmente, devenant de plus en plus significatif lorsque le nombre d'individus n dans la base croît.

Les taux d'erreur des méthodes d'apprentissage, non présentés dans ce tableau en raison de leur grande variabilité lorsque la base est de petite taille, diminuent également en fonction du nombre d'individus n présents. Par ailleurs, le poids des arêtes coupées relatif $\frac{J}{I+J}$ diminue également, proportionnellement au taux d'erreur.

3.2.6 Bilan de la statistique du poids des arêtes coupées

Le test du poids des arêtes coupées, qui poursuit les travaux de Sebban et Zighed [SZ96], est ici défini dans un cadre statistique strict qui permet de prendre en considération le poids des arêtes pour des variables prédictives aussi bien numériques que catégorielles.

Notre test repose sur la construction d'un graphe de voisinage. Nous avons déjà indiqué dans les chapitres précédents que pour obtenir un tel graphe, seule la matrice de dissimilarité sur les n points est demandée. Cette propriété donne à notre approche une dimension générale afin d'estimer la séparabilité des étiquettes, que la représentation des exemples de la base soit connue ou non.

En outre, les résultats de notre indicateur de séparabilité des étiquettes semblent donner une bonne estimation de la possibilité d'obtenir un modèle fiable d'une base de données résultant de méthodes d'apprentissage.

Dans la section suivante, nous allons voir que nous pouvons adapter ce test de manière locale afin de détecter les *outliers*. De la sorte, nous pourrons traiter ces exemples hors norme afin d'obtenir un échantillon d'apprentissage plus approprié à la constitution d'un modèle prédictif.

3.3 Filtrage des *outliers*

3.3.1 Introduction

La statistique de test du poids des arêtes coupées que nous avons présentée semble être un indicateur pertinent de la séparabilité des étiquettes d'une base d'apprentissage. Or, si dans une base où les étiquettes sont globalement bien séparables, nous nous trouvons en présence de quelques individus dont les voisins sont exceptionnellement d'une étiquette différente de la leur, nous sommes en droit de nous demander s'il est pertinent de conserver ces exemples dans notre échantillon d'apprentissage. En effet, de telles exceptions risquent de perturber les capacités de généralisation du modèle. Ce phénomène prend encore d'avantage de sens dans le cas où la base d'apprentissage est susceptible de contenir des exemples mal étiquetés.

Nous proposons ainsi une nouvelle méthode améliorant la performance en généralisation d'un algorithme d'apprentissage supervisé dans la situation où des exemples de la base d'apprentissage présentent du bruit sur la variable à prédire [MLZ02].

Nous nous situons toujours dans le cadre de l'apprentissage d'une variable catégorielle Y décrite par p variables prédictives numériques $X_1, X_2, ..., X_p$. Nous supposons que certains exemples de la base d'apprentissage ont été mal étiquetés et constituent ainsi une catégorie particulière de points « hors place », plus couramment appelés "*outliers*". La stratégie de filtrage que nous proposons identifie les exemples suspects et les retire de la base d'apprentissage afin d'améliorer les performances en généralisation de l'algorithme d'apprentissage supervisé. L'originalité de notre méthode [MLZ02] réside dans le fait que la détection d'exemples suspectés d'être mal étiquetés se fait sur la base de la statistique du poids des arêtes coupées.

3.3.2 Le problème des *outliers*

3.3.2.1 Définition

La recherche des *outliers* est une étape importante de tout processus d'extraction des connaissances à partir de données [BC83].

Sous le terme *outliers*, nous entendons des exemples « hors place » dont le caractère exceptionnel fait qu'ils perturbent la généralisation. Par la suite, nous reprendrons la définition des *outliers* donnée par Barnett et Lewis [BL84] :

Définition 3.3.1 Outlier. *Observation, ou sous-ensemble d'observations, qui apparaît contradictoire avec le reste de l'ensemble de données dont il est issu.*

3.3.2.2 *Outliers* dans la problématique de l'apprentissage

En apprentissage supervisé, le *bruit sur la variable à prédire* est distingué du *bruit sur les variables prédictives* [Qui86]. Dans ce dernier cas, Quinlan a montré que lorsque le niveau de bruit augmente, le fait d'ôter le bruit des variables prédictives diminue la performance en généralisation si le même bruit apparaît par la suite dans les exemples à tester. Cependant le problème se pose différemment dans le cas du bruit sur la variable à prédire car il concerne exclusivement l'ensemble d'apprentissage. C'est ainsi que Brodley et Friedl [BF96, BF99] ont montré que, quelle que soit la base ou la stratégie de filtrage expérimentée, le fait d'identifier ces exemples et de les retirer de l'échantillon d'apprentissage améliore notablement la performance en généralisation tant que le niveau de bruit sur la variable à prédire ne dépasse pas 20%, voire 30 ou 40% dans certains cas.

Nous rappelons que notre objectif n'est pas de réduire la taille de la base de données mais bien de filtrer l'ensemble d'apprentissage, aussi allons-nous évoquer des travaux suivant cette optique, à savoir ceux de Wilson [Wil72], de John [Joh95], ainsi que de Brodley et Friedl [BF96, BF99].

La règle *EkNN* (*Edited k Nearest Neighbor rule*) proposée par Wilson [Wil72] utilise les k-plus proches voisins (avec $k = 3$) pour filtrer l'ensemble d'apprentissage avant de procéder à une prédiction par le plus proche voisin. Dans cette approche, un exemple n'est retenu que s'il est correctement prédit à la majorité par ses k-plus proches voisins. Cette règle permet d'éliminer les exemples mal étiquetés, ainsi que ceux qui se trouvent trop près du bord des groupes d'exemples de même étiquette, lissant les frontières, tout en retenant les points intérieurs aux groupes. La règle *EkNN* peut être appliquée de manière répétée jusqu'à ce que tous les exemples conservés aient la majorité de leurs voisins qui présentent la même étiquette. Une variante, introduite par Tomek [Tom76], se propose de répéter la règle *EkNN* pour des valeurs croissantes de k. Cette procédure a été intégrée par Wilson et Martinez [WM00] dans différentes techniques de sélection d'exemples (telles que *DROP3* dans le cas du filtrage) dans le cadre des algorithmes d'apprentissage à partir d'exemples.

Pour des méthodes d'apprentissage procédant par arbres de décision, John [Joh95] a proposé un système de filtrage afin de construire des arbres robustes à travers la méthode *C4.5*. Après avoir construit l'arbre complet par éclatements successifs des différents nœuds, la méthode *C4.5* opère un élagage qui réexamine la pertinence des différents nœuds et opère des regroupements (*cf.* Quinlan [Qui93b]) pour éviter un sur-ajustement aux exemples bruités. John parle d'« exemples déroutants » pour les individus dont l'arbre final effectue un classement incorrect après élagage et les retire de l'ensemble d'apprentissage avant de relancer *C4.5*. À partir de ses expérimentations, l'auteur indique avoir obtenu une diminution significative (bien que réduite) de l'erreur moyenne et de la variabilité du taux d'erreur, ainsi qu'une diminution massive de la taille de l'arbre.

Brodley et Friedl [BF96, BF99] ont proposé une procédure de filtrage reposant sur l'emploi conjugué de diverses méthodes d'apprentissage. En utilisant m (avec $m = 3$) algorithmes d'apprentissage automatique, appelés « algorithmes de filtrage », ils repèrent les exemples de l'ensemble d'apprentissage mal étiquetés à l'issue d'une validation croisée en 10 parties. Un exemple est considéré comme mal étiqueté s'il est mal prédit par au moins deux des trois algorithmes d'apprentissage. Les méthodes d'apprentissage qui ont servi d'algorithme de filtrage à Brodley et Friedl sont l'arbre de dé-

cision avec élagage *C4.5*, le plus proche voisin et la méthode *LM* ("*Linear Machine*") qui associe les fonctions linéaires discriminantes relatives à chaque étiquette. Ensuite, un algorithme de prédiction tel que le plus proche voisin ou un arbre de décision est appliqué à l'ensemble d'apprentissage traité afin d'évaluer la fiabilité d'un modèle élaboré sur un tel échantillon de données. Les expérimentations menées sur cinq bases en introduisant un niveau de bruit contrôlé sur l'étiquette (de 0 à 40%) ont montré que la méthode permettait de maintenir l'erreur de base (celle qui correspond à 0% de bruit) jusqu'à 20% de bruit, pour toutes les bases, voire jusqu'à 30% pour deux des bases.

3.3.3 Méthode de détection et suppression des *outliers*

Dans la partie de filtrage de notre méthode, un exemple donné de l'échantillon d'apprentissage sera sujet à caution si l'étiquette de cet exemple est différente de celle des exemples de son voisinage géométrique. Cette propriété sera d'autant plus vraie que les étiquettes sont discernables dans l'espace de représentation $\mathbb{R}^p$, information qui peut être fournie à partir de notre test du poids des arêtes coupées.

Afin de repérer un exemple pourvu d'une étiquette suspecte, nous proposons de calculer la somme des poids des arêtes coupées partant de cet exemple, c'est-à-dire la somme des poids des arêtes qui relient cet individu aux exemples de son voisinage géométrique dont les étiquettes sont différentes de la sienne.

Un exemple $i \in \Omega_a$, dont l'étiquette est $e = Y(i)$ de proportion globale π_i, est considéré comme un « bon exemple » si, dans son voisinage, la proportion d'exemples de la même étiquette e est significativement plus grande que π_i.

Nous appelons H_0 l'hypothèse nulle correspondante. C'est ainsi que pour un « bon exemple » le poids des arêtes coupées est significativement plus petit que sa valeur attendue sous H_0.

Sous H_0, la probabilité pour qu'un exemple du voisinage de i ne soit pas de la même étiquette que i est $(1 - \pi_e)$. Nous notons par n_i le nombre d'exemples figurant dans le voisinage de i, $w_{i,j}$ le poids de l'arête reliant les points i et j, et J_i, calculé selon l'expression 3.3.1, le poids absolu des arêtes coupées partant de i. Les $I_i(j)$ sont des variables aléatoires indépendantes et identiquement distribuées suivant la loi alternative de paramètre $(1-\pi_e)$, sous H_0.

$$J_i = \sum_{j=1}^{n_i} w_{i,j} I_i(j) \tag{3.3.1}$$

L'espérance E et la variance Var de J_i sous H_0 sont données par les équations 3.3.2 et 3.3.3 où $e = Y(i)$.

$$E(J_i/H_0) = (1 - \pi_e) \sum_{j=1}^{n_i} w_{i,j} \tag{3.3.2}$$

$$Var(J_i/H_0) = \pi_e \, (1 - \pi_e) \sum_{j=1}^{n_i} w_{i,j}^2 \tag{3.3.3}$$

Nous proposons de considérer les exemples i, $i = 1, 2, ..., n$, en fonction de α, le risque critique unilatéral à gauche de J_i, sous H_0, définissant ainsi idéalement trois catégories d'exemples : les *bons exemples*, situés dans la région critique à gauche (risque critique α très faible), les *exemples incertains*, et enfin les *mauvais exemples* situés dans la région critique à droite (risque critique α très fort). Si n_i est assez grand et les poids pas trop déséquilibrés, nous pouvons utiliser une approximation normale et détecter les exemples directement à partir de J_i^{CR} (c'est-à-dire J_i centré et réduit).

Toutefois, pour notre problème, nous pouvons nous ramener à deux seules catégories d'exemples, ceux que nous garderons dans la base d'apprentissage (les « bons exemples ») et ceux que nous retirerons (à savoir les individus suspects : « mauvais » voire « incertains »). Pour contrôler la coupure entre les deux catégories, nous proposons un test unilatéral à gauche dont le risque α_0 peut aller de 0 (pour ne retirer aucun exemple de la base) à 1 (tous les exemples sont retirés), la valeur 1/2 correspondant à la réalisation stricte de H_0 dans le voisinage de l'exemple i.

Ce risque ne reflète qu'imparfaitement le risque réel dans la mesure où l'approximation normale est parfois forcée et où nous travaillons non pas sur un seul exemple mais sur un ensemble d'exemples.

Un exemple i d'étiquette $e = Y(i)$ sera dit « mal étiqueté » et retiré de l'ensemble d'apprentissage si nous ne pouvons pas refuser H_0 au risque unilatéral à gauche α_0 choisi. Ainsi l'exemple i sera considéré « mal étiqueté » si $J_i^{CR} \geq u_{\alpha_0}$, où u_{α_0} est la valeur en-dessous de laquelle se réalise une loi normale centrée réduite avec la probabilité α_0, comme l'indique l'équation 3.3.4.

$$J_i \geq (1-\pi_e)\sum_{j=1}^{n_i} w_{i,j} - u_{1-\alpha_0}\sqrt{\pi_e\,(1-\pi_e)\sum_{j=1}^{n_i} w_{i,j}^2} \tag{3.3.4}$$

3.3.4 Évaluation expérimentale de la méthode de filtrage

Afin d'évaluer notre procédure de filtrage, nous appliquons un algorithme de prédiction à l'ensemble d'apprentissage traité par notre méthode. Nous présentons par la suite les résultats obtenus en choisissant comme algorithme le classement par le plus proche voisin (décrit en algorithme 6, page 23). Le choix de cette méthode prédictive est simplement motivé par le souci de rester cohérent avec notre procédure d'identification, fondée elle aussi sur la distance.

La méthode de filtrage des *outliers* que nous proposons a été testée sur une dizaine de bases d'apprentissage automatique du site de l'Université de Californie à Irvine [BM98]. Tout comme pour notre évaluation précédente, les jeux d'essai choisis ne comportent que des variables prédictives numériques et une variable à prédire qualitative. Pour des raisons liées à l'influence des échelles de mesure lors de la construction du graphe de voisinage, les valeurs numériques de chaque variable prédictive ont été centrées et réduites.

La procédure de test opérée sur chacune des bases, dont les différentes étapes sont schématisée en figure 3.5, est réalisée ainsi :

- tout d'abord, nous séparons aléatoirement notre base en deux échantillons de taille égale, réservant le premier échantillon à la création du modèle et le second à la phase de test par généralisation du modèle ainsi conçu (figure 3.5(1)) ;
- sur l'échantillon d'apprentissage, nous sélectionnons aléatoirement un certain nombre d'individus (de 0 à 20%) que nous bruitons en leur attribuant une étiquette différente de la leur, cette autre étiquette étant choisie de manière aléatoire (figure 3.5(2)) ;
- avec ces données d'apprentissage plus ou moins bruitées, nous construisons un graphe de voisinage géométrique, ici le graphe des voisins relatifs [Tou80] (figure 3.5(3)) ;
- à partir de ce graphe de voisinage géométrique, nous coupons les arêtes entre sommets d'étiquettes différentes (figure 3.5(4)) et appliquons notre test pour ne retenir que les points dont la somme des poids des arêtes coupées J_i est inférieure à la valeur critique définie précédemment pour un risque $\alpha_0 = 0.10$ (figure 3.5(5)) ;

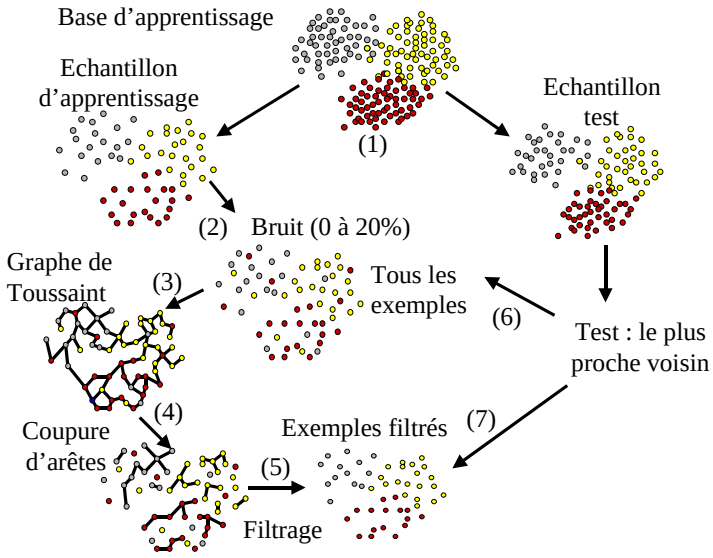

FIG. 3.5 – Représentation schématique de la procédure de filtrage des *outliers*

- les performances du modèle d'apprentissage à partir de l'échantillon sans filtrage (« tous » les exemples, figure 3.5(6)) ou après le filtrage (exemples « filtrés », figure 3.5(7)) sont estimées sur l'échantillon test ;
- cette phase de simulation est répétée 25 fois avec, à chaque fois, un nouveau partitionnement des données en échantillon d'apprentissage et échantillon test.

Les résultats obtenus suivant ce mode opératoire (avec *tous* les exemples ou lorsque les exemples sont *filtrés*) sont représentés sur les figures 3.6 et 3.7 avec en abscisse le taux de bruit introduit dans la base et en ordonnée le taux d'erreur moyen obtenu en généralisation ainsi que son écart-type, les résultats ayant été répétés 25 fois. Le pourcentage d'exemples retirés dans chaque base d'apprentissage par la procédure de filtrage est indiqué sur le tableau 3.9 pour l'introduction des dix premiers pourcentages de bruit.

Nous avons choisi de conduire notre expérimentation sur des bases pour lesquelles le taux d'erreur en généralisation n'est pas trop élevé (moins de 30%) parce qu'au-delà d'un tel taux d'erreur, une base se prête mal à un apprentissage supervisé dans la vie réelle.

Comme cela est clairement mis en évidence sur les figures 3.6 et 3.7, sur les 10 bases testées, 9 ont montré une réelle efficacité de la méthode. À partir

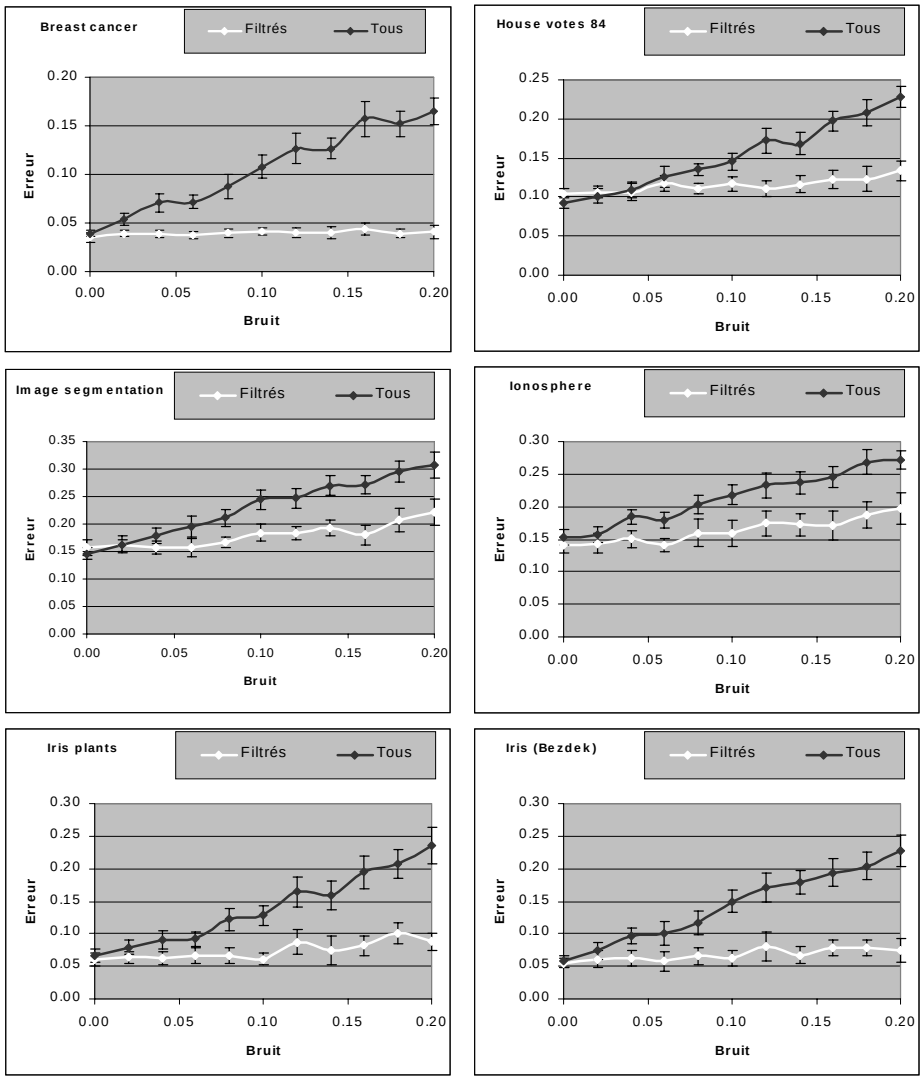

FIG. 3.6 – Taux d'erreur sur les bases *Breast cancer*, *House vote 84*, *Image segmentation*, *Ionosphere*, *Iris plants* et *Iris Bezdek*

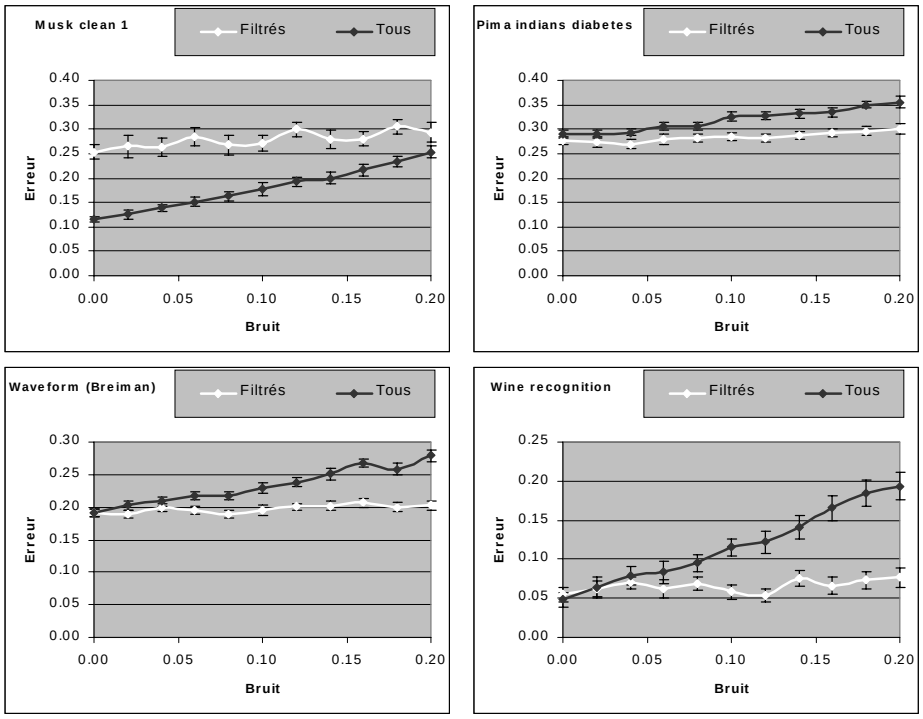

FIG. 3.7 – Taux d'erreur sur les bases *Musk "clean 1"*, *Pima Indians diabetes*, *Waveform* et *Wine recognition*

Base	n	0%	2%	4%	6%	8%	10%
Breast cancer	341	19%	20%	25%	29%	31%	34%
House votes 84	217	28%	33%	38%	41%	43%	48%
Image segmentation	105	16%	18%	22%	24%	26%	29%
Ionosphere	175	67%	70%	73%	76%	76%	78%
Iris plants	75	19%	25%	27%	28%	34%	39%
Iris (Bezdek)	75	19%	25%	27%	28%	34%	39%
Musk clean 1	238	60%	62%	66%	67%	70%	70%
Pima Indians diabetes	384	71%	72%	75%	75%	76%	78%
Waveform (Breiman)	500	32%	36%	39%	42%	44%	46%
Wine recognition	89	23%	23%	29%	31%	35%	39%

TAB. 3.9 – Pourcentage d'individus supprimés de l'échantillon à travers le filtrage pour l'introduction de 0 à 10% de bruit

de 4% de bruit, nous constatons que le filtrage est préférable 9 fois sur 10, ce qui est très significatif, et qu'il était déjà préférable 6 fois sur 10 à 0% de bruit. Le pourcentage d'exemples restant dans la base après avoir procédé au filtrage est très variable suivant les bases, mais il décroît linéairement en fonction du niveau de bruit (R^2 compris entre 0.95 et 1 pour chaque base), avec une pente quasiment proportionnelle au pourcentage d'exemples retenus à 0% de bruit.

En revanche, notre méthode ne semble pas du tout adaptée au cas de la base *Musk Clean 1*. Plusieurs raisons expliquent cette particularité. Même si la base *Musk Clean 1* n'est pas très difficile à apprendre (*cf.* tableau 3.5), celle-ci compte un très grand nombre de variables (166) pour peu d'individus (238). Le taux d'erreur résultant de notre méthode ne se détériore pas lorsque le niveau de bruit augmente, mais il commence beaucoup plus haut que le taux d'erreur sur la base non filtrée (à 0% de bruit). Ce phénomène est à rattacher au fait que pour cette base, à 0% de bruit, il ne reste que 40% des exemples (*cf.* tableau 3.9). Cette dernière considération est toutefois à nuancer car, pour la base *Pima Indians diabetes*, il ne reste que 27% des exemples à 0% de bruit alors que le filtrage est tout de suite efficace.

Nous retrouvons ainsi le problème attendu pour les bases dont les étiquettes sont peu séparables. Il apparaît donc que ce qui peut perturber la méthode est le fait que trop d'exemples soient retirés dès 0% de bruit. Pour corriger cet effet, il faut modifier la valeur du risque α_0. Après avoir procédé à un tel changement sur la base *Musk 1*, nous avons effectivement pu obtenir des résultats plus satisfaisants.

3.3.5 Bilan de la méthode de filtrage

Les résultats que nous avons obtenus avec notre méthode de filtrage se sont révélés satisfaisants. Si lors de l'application de la méthode sur des jeux d'essai, le nombre d'exemples retirés à 0% de bruit est trop important, il faut augmenter la valeur de α_0. Le test de séparabilité des étiquettes à partir du poids des arêtes coupées que nous avons décrit précédemment peut alors s'avérer fort utile dans cette situation afin de régler la valeur de α_0 suivant le résultat de ce test.

L'utilisation locale du poids des arêtes coupées nous a permis de détecter les individus d'un échantillon dont l'étiquette est hors norme vis-à-vis de l'étiquette de leurs voisins. Dans notre méthode de filtrage, ces individus sont supprimés de l'échantillon d'apprentissage. Dans la section suivante, nous proposons une extension de cette méthode qui va chercher non plus

seulement à supprimer des individus de la base mais à retrouver leur véritable étiquette.

3.4 Réétiquetage des *outliers*

3.4.1 Introduction

Il peut sembler à première vue curieux de chercher à changer la valeur de la variable à prédire de certaines observations d'une base sur laquelle il est prévu de procéder à un apprentissage automatique supervisé. Imaginons que cette base soit un ensemble de données médicales où la variable à prédire est l'état « survivant » ou « mort » du patient en fonction d'un ensemble de signes cliniques. Procéder à un réétiquetage de certains individus, dans notre ensemble de données, aurait pour effet de faire mourir des patients survivants ou ressusciter des patients décédés !

Il existe cependant des situations où un réétiquetage peut s'avérer pertinent. De telles situations peuvent se rencontrer lorsque, dans la base d'apprentissage, certaines valeurs de la variable à prédire sont sujettes à caution, soit par exemple parce que l'étiquette résulte de l'avis donné par un expert dont la fiabilité n'est pas certaine, soit que l'étiquette est l'opinion d'une personne interrogée à l'occasion d'un sondage et que l'on est en droit de s'interroger pour savoir si cette personne a menti ou non, soit enfin, comme cela est assez souvent le cas lorsque les données proviennent du domaine des sciences humaines et sociales, parce qu'il n'y a pas de valeur véritablement absolue pour l'étiquette d'une observation donnée.

Proche de ce dernier cas, nous indiquons le domaine de l'indexation manuelle de texte, un domaine qui met en œuvre toute la subtilité de l'interprétation humaine et où le réétiquetage peut aussi être particulièrement adapté. En effet, lorsque des textes doivent être indexés, il n'est pas possible de fournir des procédures automatiques et il n'est pas toujours évident d'indiquer quel est le sujet principal auquel se rattache un texte donné. Avant de procéder à système d'apprentissage, il serait ainsi très intéressant de parvenir à retrouver l'étiquette qualifiant au mieux le texte telle que l'auraient donnée une majorité de spécialistes plutôt que celle, non fausse mais peut-être pas la plus appropriée, donnée par un seul expert.

Un réétiquetage des données peut également être intéressant lorsque l'on sait qu'il existe des erreurs particulières spécifiquement sur l'étiquette. Ce cas peut se présenter dans le cadre de la reconnaissance de formes sur des

images (en noir et blanc, pour simplifier) quand l'étiquette est la valeur d'un pixel et que l'image a subi des dégradations (un pixel blanc devient noir). De tels travaux ont déjà été réalisés dans le cadre de la relaxation que nous allons préciser par la suite.

3.4.2 Réétiquetage par relaxation

L'utilisation de graphes de voisinage dans une procédure apparentée à la relaxation a déjà été proposée par Zighed, Tounissoux, Auray et Largeron [ZTAL90]. Ces auteurs ont employé le graphe de Gabriel (*cf.* algorithme 10, page 39) afin de fournir un critère tenant compte de la structure locale de chaque exemple dans un graphe de voisinage pour pouvoir procéder au réétiquetage.

Précisons maintenant ce que nous entendons par « relaxation ». Pour cela, nous reprenons la définition de la relaxation donnée par Largeron dans le cadre de la reconnaissance de formes [Lar91].

Définition 3.4.1 Relaxation. *Ensemble d'algorithmes itératifs qui prennent en compte l'information contextuelle pour améliorer les résultats de procédures de reconnaissance de forme.*

Ces techniques reposent sur deux hypothèses, la *localité* et la *cohérence*.

Définition 3.4.2 Localité. *Selon le principe de localité, l'étiquette d'un individu donné n'est influencée que par les étiquettes d'un nombre limité d'exemples voisins.*

Définition 3.4.3 Cohérence. *Selon le principe de cohérence, les étiquettes des différents exemples doivent être compatibles entre elles.*

Largeron [Lar91] indique que le schéma itératif réexamine à chaque étape l'étiquetage de chaque exemple en fonction des étiquettes de ses voisins et d'un critère de cohérence à partir de ces deux principes. Ce critère de cohérence rend compte de l'adéquation de l'étiquetage effectué.

Le réétiquetage itératif procède à partir de l'introduction d'une fonction d'étiquetage. Pour une variable à prédire présentant r étiquettes différentes, la fonction d'étiquetage associe à chaque exemple un vecteur de dimension r dont les composantes sont des poids normalisés compris entre 0 et 1. Une matrice $\mathbf{P}$, de dimensions n (nombre d'exemples) et r (nombre d'étiquettes),

rassemblera les valeurs de la fonction d'étiquetage avec en ligne les vecteurs-étiquettes de chaque exemple. Le processus itératif va chercher à rendre l'étiquetage non ambigu, c'est-à-dire atteindre l'état où chaque vecteur-étiquette n'a qu'une seule composante non nulle.

Différentes heuristiques ont été proposées à cette fin [RHZ76] ainsi que des méthodes de recherche de l'optimum local (technique du gradient avec contraintes [KI85]) ou global [Lar91]. Nous préciserons plus particulièrement ce dernier cas, le plus proche de notre méthode de réétiquetage [LMZ02a].

La méthode proposée par Largeron [Lar91] consiste à minimiser un critère global F, somme d'un indicateur de cohérence F_1 (assimilable à un poids d'arêtes coupées dans une structure de voisinage) et d'un indicateur de fidélité F_2 (nombre de modifications d'étiquettes opérées). Le critère optimisé dans la relaxation $F = F_1 + \alpha F_2$, avec α un paramètre réel strictement positif, est un compromis à trouver entre ces deux critères dont le comportement dépend de la valeur de α. Au départ, le réétiquetage par relaxation s'attaque à un problème booléen : l'étiquetage est non ambigu, chaque vecteur-étiquette a une seule composante qui vaut 1, les autres valant 0. Ce problème booléen est transformé en un problème continu lors de la phase d'optimisation avec modification itérative des composantes des différents vecteurs-étiquettes. Lorsque le processus de relaxation est achevé, les éléments continus sont ramenés à une solution booléenne en prenant pour chaque exemple l'étiquette correspondant à la plus forte composante.

3.4.3 Traitement des *outliers* : réétiquetage/suppression

La méthode que nous proposons [LMZ02a] est une poursuite du travail effectué dans le cadre du filtrage des *outliers* [MLZ02]. Elle consiste à modifier les étiquettes des exemples qui contribuent le plus à la statistique du poids des arêtes coupées [ZLM01, ZLM02].

Plutôt que de procéder à un système itératif, nous allons rechercher directement, à l'aide de notre test, les exemples douteux (ceux pour lesquels le voisinage ne semble pas cohérent, estimé à travers un poids d'arêtes coupées important). Suivant la valeur du poids d'arêtes coupées, nous nous trouverons dans une situation où l'exemple détecté est soit visiblement un individu dont l'étiquette est erronée (poids d'arêtes coupées très élevé) soit un individu douteux (poids d'arêtes coupées moyen ou élevé) dont l'étiquette est vraisemblablement fausse mais dont l'attribution d'une nouvelle étiquette ne peut se faire de façon certaine. Les premiers exemples douteux seront ainsi réétiquetés en remplaçant la valeur de leur étiquette par celle de la

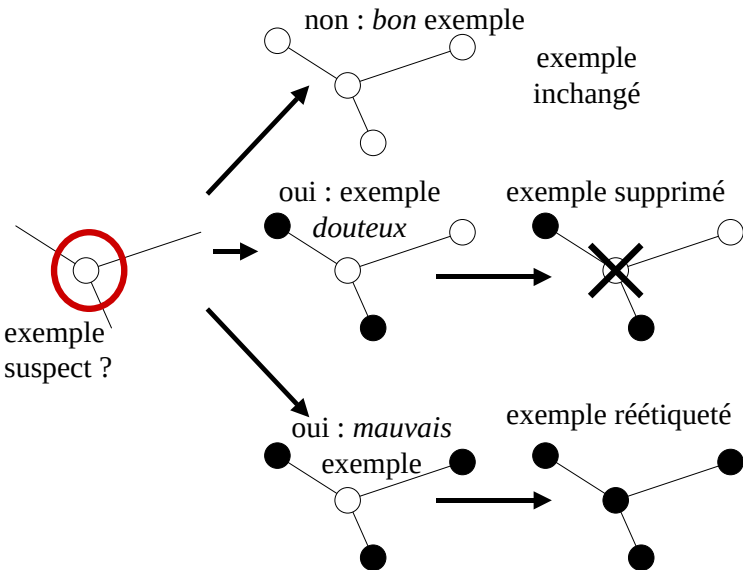

FIG. 3.8 – Représentation schématique de la procédure de suppression et réétiquetage des *outliers*

majorité de leurs voisins, les seconds seront simplement retirés de l'échantillon d'apprentissage, à la manière de notre méthode de filtrage présentée précédemment.

Nous rappelons que l'étiquette $e = Y(i)$ d'un exemple i est sujet à caution si elle est différente de celles des exemples de son voisinage. De tels exemples sont considérés comme suspects. Nous affinons à présent notre méthode, schématisée en figure 3.8 en distinguant dans l'ensemble d'apprentissage Ω_a trois types d'exemples à partir de deux risques critiques θ_1 et θ_2 (compris entre 0 et 1) :

- les « bons » exemples, qui sont laissés dans l'échantillon sans changement d'étiquette, se trouvent à gauche du risque critique θ_1 ;
- les exemples « douteux », entre les risques θ_1 et θ_2, sont supprimés de l'échantillon d'apprentissage ;
- les « mauvais » exemples, à droite du risque critique θ_2, sont réétiquetés.

Un exemple i est considéré comme *bon*, *douteux* ou *mauvais* suivant, respectivement, que le risque critique de son poids centré et réduit d'arêtes coupées J_i^{CR} est inférieur à θ_1, compris entre θ_1 et θ_2, ou est supérieur à θ_2.

En modifiant la valeur des paramètres θ_1 et θ_2, nous pouvons moduler la définition que nous donnons de chacun des types d'exemples présents dans Ω_a. Ainsi plus θ_1 est proche de 0, plus nous sommes sévères sur la définition des *bons* exemples. Plus θ_2 est proche de 1, plus nous sommes sévères sur la définition des *mauvais* exemples. Enfin, plus θ_1 et θ_2 sont semblables, moins il y a d'exemples considérés comme *douteux*. La méthode de filtrage que nous avons présentée dans la section précédente peut ainsi être reconsidérée dans ce contexte où le paramètre θ_1 joue le même rôle que le risque critique α, le paramètre θ_2 étant fixé à 1 afin de ne procéder au réétiquetage d'aucun exemple.

Après avoir procédé à plusieurs tests sur des jeux d'essai, nous avons cependant introduit une variante dans notre procédure de traitement par réétiquetage/suppression. Les tests préliminaires que nous avions réalisés afin d'observer expérimentalement sur des bases d'apprentissage quelles étaient les valeurs de θ_1 et θ_2 les plus adaptées à notre méthode ont indiqué des résultats empiriques non concluants. En effet, la définition que nous donnions des « mauvais » exemples était trop sévère : nous détections bien des individus candidats à la suppression (les exemples « douteux ») mais nous nous retrouvions privés d'exemples candidats au réétiquetage, ce qui ne permettait pas de distinguer la méthode de réétiquetage/suppression de la méthode antérieure de filtrage.

Par conséquent, nous avons modifié l'algorithme de notre méthode de réétiquetage/suppression afin de ne plus manipuler qu'un seul paramètre $\theta = \theta_1 = \theta_2$. Pour nos expériences, nous avons choisi $\theta = 0.1$, valeur que nous avions également choisie lors des tests effectués avec la méthode de filtrage.

Agissant ainsi, nous ne considérons plus aucun exemple suspect comme « douteux » et proposons tous ces exemples comme « mauvais » et donc susceptibles d'être réétiquetés. Toutefois pour qu'un exemple suspect i, d'étiquette $e = Y(i)$, prenne la nouvelle étiquette $e' = \hat{Y}(V(i))$ de son voisinage, il faut que cet exemple remplisse les conditions suivantes :

- l'exemple i doit avoir un poids d'arêtes coupées J_i significativement important (par définition) ;
- la grosse majorité des k voisins $V(i)$ de i doit avoir l'étiquette $e' = argmax_{j=1}^{k}(Y(V_j(i)))$;
- les voisins de i ne doivent pas être eux-mêmes considérés comme des points suspects.

Si les critères de cette condition ne sont pas remplis, l'exemple i est

rejeté de l'échantillon d'apprentissage, à la manière de la suppression simple effectuée dans la méthode de filtrage.

3.4.4 Évaluation expérimentale de la méthode de réétiquetage/suppression

Pour évaluer notre procédure de réétiquetage/suppression d'un échantillon d'apprentissage, nous avons repris le protocole expérimental de la méthode de filtrage avec 10 bases issues du site de l'UCI [BM98] : la base de données est séparée aléatoirement en deux échantillons de taille égale, du bruit est introduit dans l'échantillon d'apprentissage, la méthode de réétiquetage/suppression est appliquée sur cet échantillon, la prédiction est effectuée à partir de la base ainsi traitée sur l'échantillon test au moyen du plus proche voisin, et cette procédure est répétée 25 fois.

Dans les tableaux 3.10 et 3.11 nous indiquons les pourcentages, respectivement, d'exemples supprimés de l'échantillon et d'exemples réétiquetés alors que de 0 à 10% de bruit ont été introduits dans ces bases de test. Nous présentons en figure 3.9 et 3.10 les taux d'erreur et écarts-type en généralisation obtenus avec trois modes de traitements appliqués sur les exemples de la base : la méthode de réétiquetage/suppression (R ou S), la méthode de filtrage vue précédemment (Filtrés) et aucun traitement, c'est-à-dire quand tous les exemples de la base d'apprentissage sont utilisés pour produire un modèle (Tous). Les résultats sont représentés pour l'ensemble des bases avec, en abscisse, le taux de bruit introduit dans la base.

Nous remarquons que lorsqu'aucun bruit n'est introduit dans la base, les méthodes de réétiquetage/suppression et de filtrage sont meilleures que lorsque les données ne sont pas traitées pour 6 bases sur 10. À partir de 4% de bruit, nos deux méthodes sont meilleures pour 9 bases sur les 10 (seule la base *Musc clean 1* cause problème).

En dehors d'une base, les résultats sont assez étonnamment meilleurs dans le cas où les exemples sont simplement supprimés (méthode de filtrage classique) que lorsqu'un réétiquetage est effectué. En effet, la base *Breast cancer* est la seule pour laquelle la méthode de réétiquetage/suppression donne un taux d'erreur plus faible que dans le cas de la méthode de filtrage. Les tableaux 3.4 et 3.5 (page 82) nous fournissent des aides à l'interprétation utiles car il s'agit du jeu d'essai qui présente la plus grande facilité d'apprentissage sur diverses méthodes et dont l'indicateur $\frac{J}{I+J}$ de séparabilité des étiquettes est le plus faible.

Par ailleurs, l'observation des tableaux 3.10 et 3.11 nous apprend que,

Base	n	0%	2%	4%	6%	8%	10%
Breast cancer	341	13%	12%	15%	17%	18%	21%
House votes 84	217	13%	17%	20%	22%	24%	27%
Image segmentation	105	8%	10%	13%	15%	16%	19%
Ionosphere	175	61%	64%	67%	71%	71%	74%
Iris plants	75	12%	18%	19%	21%	26%	31%
Iris (Bezdek)	75	13%	18%	20%	21%	26%	31%
Musk clean 1	238	45%	49%	53%	54%	58%	59%
Pima Indians diabetes	384	58%	60%	63%	64%	65%	68%
Waveform (Breiman)	500	14%	16%	18%	19%	22%	24%
Wine recognition	89	14%	13%	18%	20%	23%	27%

TAB. 3.10 – Pourcentage d'individus supprimés de l'échantillon à travers la méthode de réétiquetage/suppression lors de l'introduction de 0 à 10% de bruit

Base	n	0%	2%	4%	6%	8%	10%
Breast cancer	341	4%	4%	4%	4%	4%	4%
House votes 84	217	10%	10%	8%	8%	6%	7%
Image segmentation	105	6%	6%	6%	6%	5%	5%
Ionosphere	175	2%	2%	2%	1%	2%	1%
Iris plants	75	6%	6%	6%	6%	5%	5%
Iris (Bezdek)	75	6%	7%	6%	5%	5%	5%
Musk clean 1	238	10%	10%	9%	8%	7%	6%
Pima Indians diabetes	384	8%	6%	6%	6%	5%	5%
Waveform (Breiman)	500	13%	13%	13%	12%	12%	12%
Wine recognition	89	2%	2%	2%	2%	2%	2%

TAB. 3.11 – Pourcentage d'individus dont l'étiquette a été changée à travers la méthode de réétiquetage/suppression lors de l'introduction de 0 à 10% de bruit

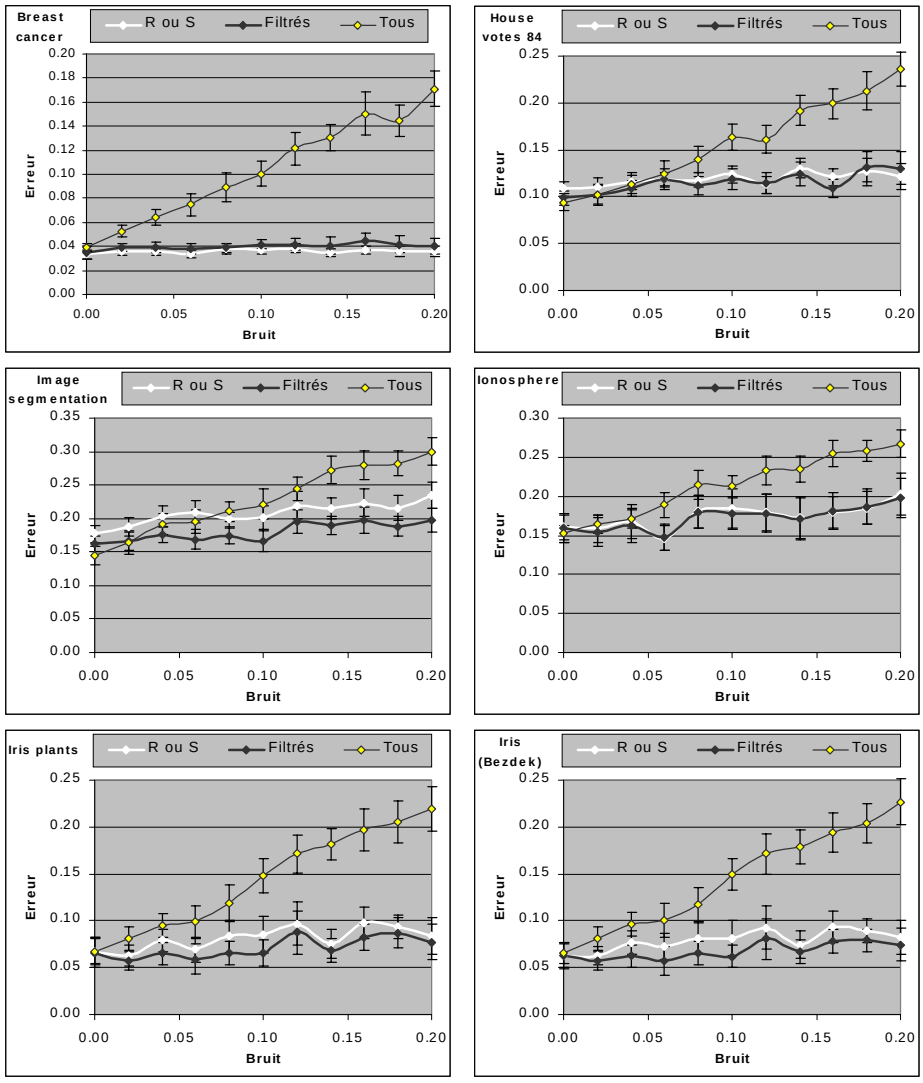

FIG. 3.9 – Taux d'erreur sur les bases *Breast cancer*, *House vote 84*, *Image segmentation*, *Ionosphere*, *Iris plants* et *Iris Bezdek*

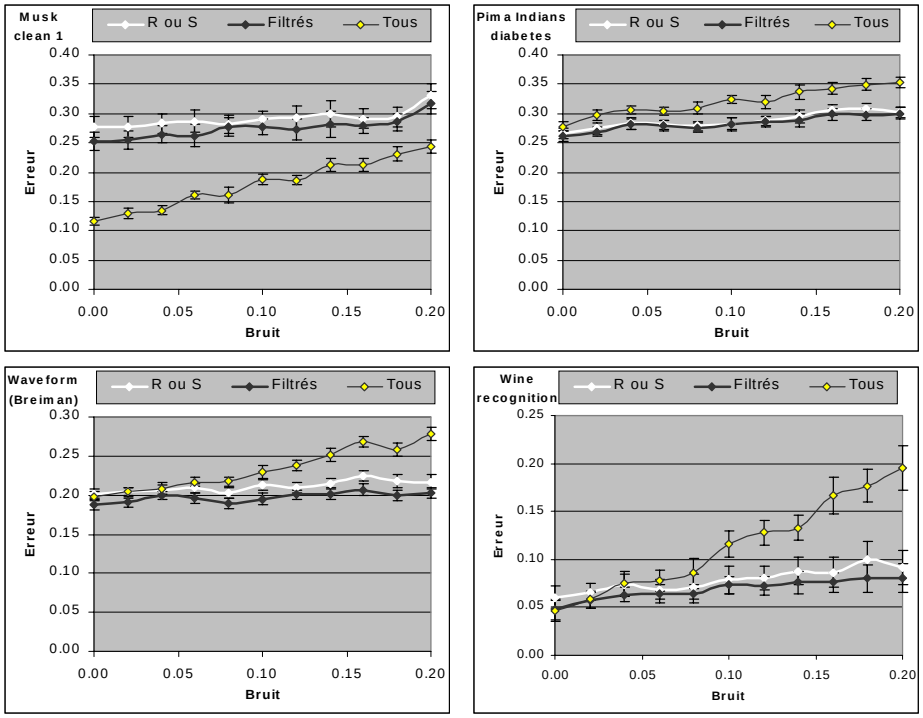

FIG. 3.10 – Taux d'erreur sur les bases *Musk "clean 1"*, *Pima Indians diabetes*, *Waveform* et *Wine recognition*

dans la méthode de réétiquetage/suppression, le pourcentage d'exemples supprimés est bien plus grand que celui d'exemples réétiquetés. De plus, le nombre d'individus supprimés tend à s'accroître en fonction du bruit introduit alors que la proportion des individus réétiquetés reste constante (ou diminue dans quelques cas). Ce phénomène peut s'expliquer par le fait que seuls sont réétiquetés les exemples dont le voisinage est considéré comme bien étiqueté. Quand les voisins d'un exemple suspect sont de diverses étiquettes, le réétiquetage ne peut être effectué et l'exemple en question est supprimé.

3.4.5 Méthode de réétiquetage/suppression et relaxation

La manière dont procède la méthode que nous venons de présenter, avec un réétiquetage ou une suppression de certains exemples, peut être interprétée suivant un mode de recherche du critère minimal dans le schéma de relaxation.

Tout d'abord, par le réétiquetage des exemples les plus différents de leurs voisins, nous suivons l'hypothèse de localité, et l'hypothèse de cohérence est assurée par le réétiquetage des individus en fonction de la majorité des voisins. De plus, notre réétiquetage assure la minimisation directe du critère de cohérence locale, puisqu'il repose sur la statistique du poids des arêtes coupées, tout en minimisant l'indicateur de fidélité, puisque seules les étiquettes des exemples les plus inattendus sont modifiées en raison de la suppression des exemples sur lesquels une décision de réétiquetage ne peut être prise. Par ailleurs, l'arbitrage entre les critères de cohérence et de fidélité se fait, dans notre méthode, dans le choix du paramètre θ qui définit les exemples inattendus.

Cependant, une grande différence entre notre méthode et la relaxation est que, d'une part, nous ne procédons pas de manière itérative mais procédons directement au réétiquetage à partir d'un test statistique et que, d'autre part, nous supprimons des exemples alors que dans le cadre de la relaxation, les exemples sont toujours présents dans l'échantillon mais prennent la valeur de la composante la plus forte du vecteur-étiquette (étiquette majoritaire).

3.4.6 Bilan de la méthode de réétiquetage/suppression

La méthode de réétiquetage/suppression que nous proposons ne semble finalement présenter qu'un intérêt mineur comparé à la méthode de filtrage. Nous décelons en effet deux problèmes principaux :

– le premier est le réétiquetage erroné de quelques exemples de l'échan-

tillon. Ainsi l'étiquette de certains individus est changée par la méthode de réétiquetage/suppression alors qu'aucun bruit n'est introduit dans la base, or ce mauvais réétiquetage est plus grave que de simplement supprimer l'individu en question de la base comme nous le faisons dans le cas de la méthode de filtrage ;

– le second problème de la méthode est que des individus dont l'étiquette a réellement été bruitée ne sont pas considérés comme des candidats au réétiquetage. En effet, nous observons dans les tableaux 3.10 et 3.11 que la proportion d'individus réétiquetés n'augmente pas avec l'introduction du bruit : les individus bruités sont supprimés et non réétiquetés.

Le problème essentiel de la méthode de réétiquetage/suppression prend ses sources dans la difficulté que nous avons eu à définir ce qu'est un « mauvais » exemple. Nous avons en effet indiqué que les exemples considérés comme suspects en raison de l'importance du poids de leurs arêtes coupées étaient *tous* candidats à la phase de réétiquetage. Seule la condition portant sur le voisinage d'un exemple suspect (à savoir posséder une majorité de voisins d'une étiquette donnée, ces voisins étant eux-mêmes non suspects) limite le réétiquetage à tort d'un tel exemple.

Par conséquent, deux voies d'amélioration de la méthode de réétiquetage sont encore à explorer. La première consisterait à adapter notre méthode pour introduire un système itératif, à la manière du schéma de relaxation. De la sorte, nous ne déciderions plus brutalement de considérer un exemple comme « bon » ou « mauvais » mais ferions ressortir progressivement les *outliers*, ce qui aurait l'avantage de garder des voisinages stables et favorables à l'attribution d'une bonne étiquette.

La seconde voie d'amélioration envisagée serait de mieux définir ce qu'est un « mauvais exemple » ainsi que le mode de réétiquetage employé. En effet, en changeant la structure de voisinage et proposant, au lieu du graphe des voisins relatifs de Toussaint, le graphe de Gabriel, le voisinage d'un *outlier* serait sans doute plus à même de proposer une étiquette correcte, car un tel graphe fournit plus d'arêtes, et par voie de conséquence de voisins, que le graphe des voisins relatifs. En outre, le mode de réétiquetage peut également être optimisé en proposant des critères d'attribution de la nouvelle étiquette à travers un vote pondéré ou en modifiant la condition qui autorise ou non le réétiquetage d'un exemple suspect.

3.5 Conclusion

Au cours de ce chapitre, nous avons présenté un test statistique qui donne une information sur la séparabilité des étiquettes d'une base d'apprentissage, information qui peut se révéler précieuse dans une perspective d'ECD, en particulier avant de débuter un travail de fouille de données. Nous avons ensuite exploité les propriétés locales de ce test pour proposer une méthode de détection des *outliers* dans une base d'apprentissage et ceci afin de procéder au filtrage ou au réétiquetage des points mal étiquetés.

Même si l'objectif initial de notre méthode de détection des *outliers* est de trouver les exemples présentant une étiquette bruitée, il n'en faut pas moins souligner que le filtrage effectue quand même une certaine réduction du nombre de données de la base d'apprentissage. Or cette réduction du nombre d'exemples peut, d'une certaine manière, s'apparenter à une sélection de prototypes. Par conséquent, l'opération de filtrage peut être employée en tant que phase de pré-traitement dans le cadre des apprentissages à base d'exemples, présentés dans le chapitre premier, méthodes où le nombre d'individus dans la base d'apprentissage joue beaucoup sur la rapidité de classement en plus d'influer la qualité du modèle prédictif.

Certes, nos méthodes de filtrage et de réétiquetage cherchent plutôt à conserver les exemples et ne retirent que ceux considérés comme étant des *outliers*, alors que les méthodes de sélection de prototypes essaient davantage de ne garder qu'un minimum d'exemples capables de rendre compte de l'ordonnancement général de la base d'apprentissage. Cependant, comme le filtrage, par son traitement local des données bruitées, va supprimer les cas non généralisables afin d'homogénéiser l'espace de représentation, il peut être considéré comme la première étape d'une méthode plus générale de sélection de prototypes. Nous envisageons en effet de poursuivre la méthode en ce sens par l'ajout d'une deuxième étape qui procéderait à la construction d'un nouveau graphe de voisinage à partir des individus issus de ce premier filtrage et qui ne conserverait finalement que les seuls exemples présents sur les frontières, c'est-à-dire ceux qui sont reliés par une arête mais dont l'étiquette est différente.

Généralisation à l'apprentissage d'une variable numérique

Sommaire

Chapitre 4

Généralisation à l'apprentissage d'une variable numérique

Résumé

Ce chapitre est consacré à une généralisation du test évaluant la qualité de la représentation donnée par un ensemble de variables prédictives lorsque la variable à apprendre est numérique. Nous indiquons les travaux réalisés dans le cadre de l'analyse spatiale et présentons un nouveau test de structure appliqué au cas de l'apprentissage supervisé d'une variable numérique en adaptant l'autocorrélation spatiale au voisinage fourni par des graphes construits dans un espace multidimensionnel $\mathbb{R}^p$.

Nous illustrons ensuite la manière de procéder à la recherche d'*outliers* numériques avec des exemples réalisés dans le domaine du traitement d'image.

4.1 Introduction

Au cours du précédent chapitre, nous avons exposé des travaux que nous avons menés dans le cadre de l'apprentissage supervisé d'une variable catégorielle. Considérons à présent le cadre plus général de l'apprentissage supervisé en nous intéressant au cas de l'apprentissage d'une variable numérique.

Dans cette situation, l'objectif n'est pas de chercher à retrouver une étiquette donnée mais d'obtenir une valeur numérique en fonction des valeurs

des variables prédictives. Nous présentons ci-dessous plusieurs familles de méthodes aboutissant à un tel résultat, méthodes qui peuvent éventuellement se combiner entre elles [Qui93a].

Régression multivariée linéaire La régression [AMS97, UG99], et plus particulièrement la *régression multivariée linéaire*, est une des méthodes les plus communément employées dans le cas de l'apprentissage d'une variable numérique. Son principe est décrit par l'équation 4.1.1.

$$\widehat{Y}(\omega) = \mu_0 + \sum_{i=1}^{p} \mu_i \times X_i(\omega) \tag{4.1.1}$$

Pour ce problème, qu'il est possible de résoudre à travers des inversions de matrices, il faut retrouver les $(p+1)$ coefficients μ qui minimisent la somme des carrés des écarts entre les valeurs réelles et celles prédites pour les n_a exemples d'apprentissage. Nous notons que les variables prédictives X_i n'ont pas nécessairement besoin d'être numériques, des adaptations sont en effet éventuellement possibles avec des variables prédictives catégorielles en ordonnant les diverses modalités de celles-ci et en prenant comme valeur le rang de chaque modalité.

Apprentissage à base d'exemples Au cours du chapitre premier, nous avons indiqué comment était réalisé le classement pour les algorithmes *IBL* tels que ceux décrits par Aha, Kibler et Albert [AKA91]. Dans le cas où la variable à prédire est numérique, au lieu d'attribuer une étiquette semblable à celle des exemples voisins, la valeur prédite est simplement la moyenne (pondérée ou non) des valeurs de ces voisins. Par ailleurs, au lieu de calculer une moyenne, nous signalons qu'il est aussi possible de procéder à une régression pondérée localement.

Arbres de décision Les apprentissages à base de modèle sont également adaptés, pour certains, à fournir des résultats lorsque la variable à prédire est numérique. Citons notamment l'arbre de décision *CART* (abréviation de "*Classification And Regression Trees*") de Breiman, Friedman, Olshen et Stone [BFOS84] et la méthode *M5* de Quinlan [Qui92].

Réseaux de neurones Les réseaux de neurones formels peuvent aussi servir à l'apprentissage d'une variable numérique [RM86]. En effet, il suffit

pour cela d'avoir en sortie, au lieu d'une fonction à seuil fournissant comme résultat des modalités catégorielles, une valeur résultant d'un calcul similaire au modèle de régression (*cf.* équation 4.1.1) avec :

- pour X_i : les connexions des neurones provenant de la dernière couche cachée ;
- pour p : le nombre de connexions arrivant au neurone de sortie ;
- pour μ : les poids des connexions.

4.2 Test de structure pour la prédiction des variables numériques

4.2.1 Introduction

Dans le chapitre précédent, nous avons présenté des travaux évaluant la qualité de la représentation lorsque la variable à prédire est catégorielle, indiquant notamment à travers notre test du *poids des arêtes coupées* si les étiquettes d'une base d'apprentissage sont séparables ou non. Ainsi, lorsque les exemples d'une base se répartissent dans l'espace de représentation suivant p variables prédictives avec des étiquettes bien séparables, nous pouvons supposer que ces exemples peuvent être aisément appris par des méthodes d'apprentissage fondées sur les exemples telles que celles présentées dans le premier chapitre.

Quand une variable à prédire est numérique, la qualité de la représentation ne peut plus être évaluée à travers l'étude de la distribution des étiquettes, aussi devons-nous trouver un moyen d'estimer la façon dont se répartissent les diverses valeurs numériques de la variable à prédire. Agissant de manière analogue au cas de l'apprentissage d'une variable catégorielle, nous supposons qu'une variable numérique pourra être apprise si les valeurs de cette variable se répartissent suivant une certaine structure dans l'espace de représentation, les petites valeurs étant présentes à un endroit donné de l'espace et distantes des grandes valeurs.

Afin de tester la présence de structure, nous réalisons un graphe de voisinage sur les exemples de la base d'apprentissage mais il ne nous est plus possible de nous intéresser au poids des arêtes coupées [ZLM01] comme nous l'avons fait pour l'apprentissage d'une variable catégorielle en raison d'absence d'étiquettes, similaires ou différentes, qui nous avaient permis de procéder à la suppression de certaines arêtes du graphe. À la place, nous utilisons le coefficient de Moran [Mor48] et les outils développés dans le domaine de

c de Geary	I de Moran	Interprétation
$0 < c < 1$	$I > 0$	Similarité, valeurs régionalisées
$c = 1$	$I = 0$	Indépendance, structure aléatoire
$c > 1$	$I < 0$	Dissimilarité, contraste, structure en damier

TAB. 4.1 – Interprétation des coefficients de Geary et de Moran

l'autocorrélation spatiale [LMZ02b].

4.2.2 Structure et autocorrélation spatiale

Les graphes spatiaux sont des outils notamment employés en géographie. L'analyse spatiale considère des sites en deux dimensions, voire trois, décrits par une variable Y et cherche à rendre compte du fait que les données à caractère spatial qui ont des valeurs similaires pour cette variable Y sont proches dans le plan ou dans l'espace. Nous nous intéressons donc à l'autocorrélation de Y, c'est-à-dire la corrélation existant entre des paires d'observations réalisées à partir de cette même variable, mesure qui est classiquement opérée à travers les coefficients de Moran [Mor48] ou de Geary [Gea54]. Ces mesures d'autocorrélation spatiale reposent sur l'hypothèse que ce qui se produit en un lieu géographique donné a une influence sur les lieux voisins.

Il y a autocorrélation spatiale positive lorsque les sites voisins présentent des valeurs semblables de Y et autocorrélation spatiale négative lorsqu'ils présentent des valeurs contradictoires. L'absence d'autocorrélation spatiale signifie que la valeur présentée par un site ne dépend pas des valeurs de ses sites voisins. L'interprétation des coefficients c de Geary et I de Moran est donnée par le tableau 4.1.

Nous avons décidé d'utiliser le coefficient I de Moran, considéré assez unanimement comme étant le meilleur choix, parce qu'il présente de bonnes propriétés locales [Ans95], ce qui nous permet de faire aussi des rapprochements entre cette nouvelle problématique et celle de la détection des *outliers* que nous avons réalisée dans le cas de la prédiction d'une variable catégorielle présentée dans le chapitre précédent. De plus, selon Cliff et Ord [CO86], le coefficient de Moran est plus puissant que l'indice c de Geary avec une variance moins sensible à la distribution des observations.

L'indice global I de Moran, dont nous donnons une expression en équation 4.2.1, a pour numérateur une covariance pondérée entre les observations contiguës et pour dénominateur une mesure de variance des observations.

$$I = \frac{n}{\sum_{i=1}^{n} \sum_{j=1,\, i \neq j}^{n} w_{i,j}} \times \frac{\sum_{i=1}^{n} \sum_{j=1,\, i \neq j}^{n} w_{i,j}(y_i - \bar{y})(y_j - \bar{y})}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \qquad (4.2.1)$$

où :

- n est égal au nombre d'unités spatiales ;
- w représente le nombre de régions contiguës (et $w_{i,j}$ est donc un poids qui vaut 1 si les sites i et j sont voisins, et 0 sinon) ;
- y_i, notation simplifiée de $Y(i)$, est la valeur (numérique) de la variable Y pour le site i ;
- et $\bar{y}$ est la valeur moyenne de la variable Y.

4.2.3 Test de structure à partir du coefficient de Moran

Pour tester la structure d'une variable à prédire numérique dans un espace de représentation donné par p variables prédictives, nous construisons à nouveau un graphe de voisinage (le graphe des voisins relatifs de Toussaint [Tou80]).

Dans ce contexte, la formule 4.2.1 peut être utilisée en prenant pour n le nombre d'exemples d'apprentissage et pour w un poids dépendant de la simple connexion, de la distance ou du rang. Dans le cas où ce poids se limite à la seule connexion, la formule se simplifie sous la forme de l'équation 4.2.2 où a indique le nombre d'arêtes et v la présence d'une connexion ($v_{i,j}$ vaut 1 si i et j sont reliés par une arête et 0 si ce n'est pas le cas).

$$I = \frac{n}{2a} \times \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} v_{i,j}(y_i - \bar{y})(y_j - \bar{y})}{\sum_i (y_i - \bar{y})^2} \qquad (4.2.2)$$

Quel que soit l'indice d'autocorrélation spatiale retenu pour tester la structure de nos données dans l'espace de représentation, nous devons faire le choix entre deux schémas probabilistes.

Schéma gaussien En nous plaçant dans un schéma gaussien, nous supposons que les observations sont le résultat de n tirages indépendants dans une population normale.

Schéma randomisé Ce cadre est moins restrictif que le schéma gaussien. Nous supposons ici que la localisation des n observations est le résultat d'un

tirage au hasard pur parmi les $n!$ permutations de ces valeurs sur l'ensemble des localisations possibles.

Cliff et Ord [CO86] indiquent comment calculer les moments relatifs de la loi suivie par I sous l'hypothèse nulle pour les schémas gaussien et randomisé. L'espérance de I, dans les deux cas, est donnée par l'équation 4.2.3.

$$E(I) = \frac{-1}{n-1} \tag{4.2.3}$$

Dans les deux cas, la variance σ_I^2 est calculée selon la formule de Huyghens (*cf.* équation 4.2.4) à travers $E(I^2)$, le moment d'ordre 2.

$$\sigma_I^2 = E(I^2) - (E(I))^2 \tag{4.2.4}$$

Dans le schéma gaussien, le moment d'ordre 2 est donné par la formule 4.2.5.

$$E(I^2) = \frac{n^2 S_1 - n S_2 + 3 S_0^2}{S_0^2 (n^2 - 1)} \tag{4.2.5}$$

où nous notons :

- $\sum_2 w_{i,j}$ pour la double somme de poids $\sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} w_{i,j}$;
- S_0 pour $\sum_2 w_{i,j}$;
- S_1 pour $\frac{1}{2} \sum_2 (w_{i,j} + w_{j,i})^2$;
- S_2 pour $\sum_{i=1}^{n} (w_{i+} + w_{+i})^2$, avec w_{i+} la somme des poids de la ligne i et w_{+j} la somme des poids de la colonne j.

Dans le schéma randomisé, le moment d'ordre 2 est donné par l'équation 4.2.6.

$$E(I^2) = \frac{n\left[\left(n^2 - 3n + 3\right) S_1 - n S_2 + 3 S_0^2\right] - b_2\left[\left(n^2 - n\right) S_1 - 2n S_2 + 6 S_0^2\right]}{(n-1)^3 S_0^2} \tag{4.2.6}$$

où b_2 est le coefficient d'aplatissement de Pearson.

La loi de I sous H_0 est asymptotiquement normale à condition qu'il n'y ait pas d'écart trop important entre les quantités $(\sum_{j=1}^{n}(w_{i,j} + w_{j,i}) \times y_j)$ qui, pour toutes les observations i, tiennent à la fois compte des poids et des valeurs de la variable Y. Cette condition étant peu contraignante, nous procédons par approximation normale pour calculer le risque critique de la valeur de I observée.

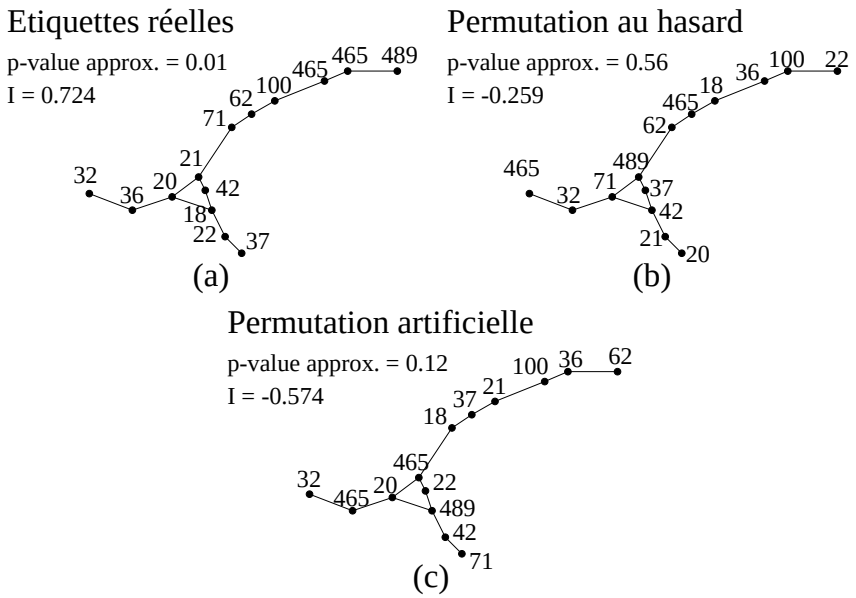

FIG. 4.1 – Illustration du test de structure

4.2.4 Expérimentations

4.2.4.1 Illustration du test de structure sur un échantillon

Afin d'expliquer le comportement de notre test de structure, nous présentons d'abord un cas simple : le traitement d'un échantillon de 14 exemples extraits de *CPU*, une base du répertoire d'apprentissage automatique de l'Université de Californie à Irvine [BM98]. Sur la figure 4.1, nous indiquons dans différents cas le graphe de voisinage (graphe des voisins relatifs) adapté à une représentation plane et les étiquettes numériques correspondant aux sommets, ainsi que la valeur du coefficient I et son risque critique (*p-value*) dans un schéma randomisé calculé avec pour poids w la simple connexion.

La figure 4.1(a) concerne les étiquettes véritables de l'échantillon d'apprentissage. Dans ce cas, comme $I = 0.724$ avec un risque critique de l'ordre de 0.01, la distribution des étiquettes peut être considérée comme structurée, ce qui est illustré par l'ordonnancement des étiquettes sur le graphe.

Sur la figure 4.1(b), nous avons opéré une permutation aléatoire des étiquettes. Le coefficient I a alors une valeur faiblement négative mais non significative.

En figure 4.1(c), nous avons attribué les étiquettes à partir d'une permutation artificielle en prenant soin d'alterner les grandes et faibles valeurs de Y. Dans ce cas, $I = -0.574$ et la *p-value* $= 0.12$. Le risque critique, sans être significatif, n'est pas très éloigné de la signification. La valeur négative de I indique qu'il s'agit d'un type particulier de structuration : les valeurs de Y voisines sont contrastées.

4.2.4.2 Expérimentations sur un ensemble de bases

Nous présentons dans le tableau 4.2 les résultats obtenus après avoir appliqué notre test de structure à différentes bases d'apprentissage dont la variable à prédire est numérique. Ces bases sont extraites du site de l'Université de Californie à Irvine (*auto-MPG*, *autos*, *CPU*, *housing* et *servo*) [BM98], ainsi que du site StatLib de l'Université Carnegie Mellon[1] (*plasma*, avec deux variables à apprendre : *β-carotène* et *rétinol*) auxquelles nous avons ajouté la base *pw-linear* générée suivant le modèle de Breiman [BFOS84].

Pour chaque base, nous avons retiré les individus présentant des données manquantes. Parmi les p variables prédictives initiales, les éventuelles variables catégorielles ont été ré-encodées sous forme disjonctive complète (le nouveau nombre de variables prédictives numériques est noté $p^\star$). Les variables prédictives numériques ont été centrées et réduites. Pour chaque base, en plus de la valeur du coefficient I de Moran et de son risque critique (*p-value*), nous indiquons le nombre d'individus présents dans la base (n), le nombre initial d'attributs prédictifs (p) et le nombre de variables prédictives numériques utilisées pour la constitution du graphe ($p^\star$).

Nous observons que pour l'essentiel des bases d'apprentissage, le test de structure est très fortement significatif (*p-values* inférieures à 0.01).

La seule base pour laquelle ceci n'est pas le cas est *plasma* avec l'apprentissage de la variable *rétinol* alors que la distribution de *β-carotène*, qui comprend les mêmes individus et les mêmes valeurs pour les variables prédictives, est considérée comme structurée, aussi cette base *plasma* mérite-t-elle quelques éléments explicatifs complémentaires.

Nierenberg *et al.* [NSB+89], les auteurs de la base sur les déterminants de la concentration plasmique en rétinol et β-carotène, avaient remarqué que des facteurs tels qu'une alimentation pauvre en caroténoïdes ou des faibles concentrations plasmiques en rétinol et β-carotène étaient associés à l'augmentation du risque de développer certains types de cancer. Ils ont

[1] http ://lib.stat.cmu.edu/datasets/

Base	n	p	$p^\star$	I	*p-value*
auto-MPG	392	7	9	0.853	0
autos	159	25	64	0.574	3.8E-18
CPU (PRP)	209	6	6	0.581	0
housing	506	13	13	0.743	0
plasma (β-carotène)	315	12	16	0.206	4.5E-07
plasma (rétinol)	315	12	16	0.027	0.48
pw-linear	200	10	10	0.327	5.4E-12
servo	167	4	12	0.62	0

TAB. 4.2 – Tests de structure sur 8 bases

ainsi réalisé une étude pour retrouver des relations existant entre les caractéristiques personnelles des sujets ainsi que les facteurs alimentaires et les concentrations plasmiques de rétinol, β-carotène et d'autres caroténoïdes. En croisant les différents profils, Nierenberg *et al.* ont observé que la concentration de rétinol plasmique varie en fonction de l'âge, du sexe et d'un seul paramètre alimentaire (sur l'ensemble des 12 variables prédictives) alors que la concentration de β-carotène est assez corrélée (aussi bien positivement que négativement) avec davantage de variables prédictives, les variables correspondant aux concentrations plasmiques de β-carotène et de rétinol n'étant pas corrélées entre elles.

Il en ressort que les distributions des valeurs associées à ces deux paramètres sur le graphe de voisinage ne s'effectuent pas de la même manière. Il est ainsi fort probable que les valeurs de la variable *β-carotène* se répartissent suivant un graphe analogue à celui présenté en figure 4.1(a) alors que celles de la variable *rétinol* ressortent plutôt d'un graphe dont la structure ressemble davantage à celui présenté en figure 4.1(b).

4.2.5 Bilan

Ce test, prolongeant nos travaux sur la séparabilité des étiquettes [ZLM01], a ceci d'original qu'il associe les graphes de voisinage aux coefficients d'autocorrélation spatiale pour rendre compte de la structure d'une variable prédictive numérique dans un espace de représentation multidimensionnel.

De la sorte, nous pouvons donner à présent une information a priori sur la qualité de l'espace de représentation correspondant à l'apprentissage d'une variable, quelle que soit sa nature : le test du poids des arêtes coupées lorsque

la variable à prédire est catégorielle, le test de structure associé au coefficient de Moran lorsque la variable à prédire est numérique.

4.3 Détection des *outliers* numériques

4.3.1 Détection des *outliers* dans le cadre de la régression

Comme l'indiquent Belsley, Kuh et Welsh [BKW80], la problématique des *outliers* a été très étudiée dans le cadre de la régression linéaire. Classiquement, les *points aberrants* sont distingués des *points leviers*. Les points aberrants sont ceux pour lesquels la réponse aux variables prédictives obéit à un modèle différent de celui des autres points alors que les points leviers sont ceux qui occupent une position très excentrée dans l'espace de représentation $\mathbb{R}^p$, ce qui leur donne un grand pouvoir d'attraction sur l'hyperplan de régression [Rit90]. Certains exemples peuvent d'ailleurs relever des deux catégories.

Les points aberrants sont repérés à partir de l'examen des résidus standardisés ou studentisés, alors que les points leviers sont détectés en calculant leur distance au centre de gravité du nuage pour une métrique à définir [Rit90].

Le point atypique examiné dans le calcul du centre de gravité et dans celui de l'hyperplan de régression n'est pas pris en compte. Pour autant, le calcul reste très sensible aux autres *outliers* possibles. Un critère commun peut aussi être utilisé, comme la distance de Cook entre les résultats de la régression avec et sans le point suspect.

Ces méthodes ont l'inconvénient de traiter les *outliers* un par un et ainsi d'être peu adaptées à la situation où il y a plusieurs *outliers*. Pour cette raison, des outils non paramétriques sont employés afin de rendre plus robuste le calcul du centre de gravité et de l'hyperplan. De tels outils consistent à remplacer la moyenne par la médiane, à employer la méthode de Theil (qui considère les droites joignant les exemples 2 à 2 et prend la médiane des pentes ainsi obtenues comme pente de la droite de régression) ou encore les M-estimateurs de Huber. Citons enfin la méthode LTS (*Least Trimmed Square*, ou « moindres carrés tronqués ») proposée par Rousseeuw et Leroy [RL87] qui minimise la somme des carrés des erreurs tronquée des exemples correspondants aux erreurs extrêmes. L'examen graphique des couples formés par les résidus robustes standardisés et les distances robustes standardisées est alors très éclairant [Rit90].

4.3.2 Autocorrélation spatiale locale

Lorsque nous nous sommes intéressés aux variables à prédire catégorielles, nous avons pu adapter notre test de séparabilité des étiquettes dans l'espace de représentation sous une forme locale afin de détecter les *outliers* et filtrer la base d'apprentissage des exemples présentant du bruit sur la variable à prédire.

Dans le cas de l'apprentissage d'une variable numérique, nous pouvons procéder à une adaptation similaire de notre test de structure fondé sur le coefficient de Moran en nous inspirant des travaux réalisés dans le domaine de l'autocorrélation spatiale locale. Anselin [Ans95] a ainsi récemment développé ce terrain en décomposant l'indice global d'autocorrélation spatiale de façon à identifier la contribution locale de chaque site. La raison à cela est que les mesures d'autocorrélations globales permettent de tester les motifs spatiaux sur l'ensemble de l'aire (ou de l'espace) d'étude mais il peut y avoir une autocorrélation significative dans des petites sections seulement, et cette information peut être noyée dans le contexte de l'ensemble.

Dans ces mesures locales, pour chaque site, c'est l'association spatiale entre la valeur de la variable Y en ce lieu et l'ensemble de celles prises dans son voisinage qui est mesurée. Ces indices permettent ainsi de détecter les zones locales d'autocorrélation spatiale. Le coefficient de Moran local (pour un site i donné) prend alors pour expression l'équation 4.3.1.

$$I_i = \frac{(y_i - \bar{y}) \times \sum_{j=1,\, j\neq i}^{n} w_{i,j}(y_j - \bar{y})}{\sum_{j=1,\, j\neq i}^{n} \frac{(y_j - \bar{y})^2}{n}} \tag{4.3.1}$$

Anselin [Ans95] a défini deux propriétés pour qu'un indice soit considéré comme un indicateur local d'association spatiale (ou "*LISA*" pour "*Local Indicators of Spatial Association*") :

- pour chaque observation, l'indice doit donner une indication d'un éventuel regroupement de valeurs similaires (ou dissemblables) dans le voisinage de cette observation ;
- la somme des indices locaux sur l'ensemble des observations est proportionnelle à l'indice global correspondant.

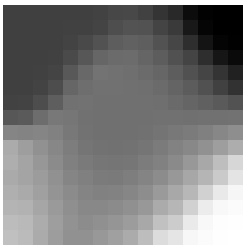

FIG. 4.2 – Image structurée

4.3.3 Application de l'analyse spatiale au traitement d'image

4.3.3.1 Introduction

Les travaux que nous avons engagés dans le domaine de l'apprentissage supervisé d'une variable numérique étant encore très récents, nous ne présenterons pas le processus de recherche des *outliers* dans une base d'apprentissage mais nous illustrerons le comportement du coefficient de Moran en tant qu'outil de détection d'*outliers* dans une image. Ce type d'application peut servir à traiter une image lorsque celle-ci a subi certaines détériorations ou dans le cas où il y eu des poussières présentes dans le dispositif d'acquisition de l'image.

Pour ce type d'application, nous nous retrouvons dans le cadre plus classique de l'autocorrélation spatiale et, de ce fait, les voisins ne sont pas obtenus à travers un graphe de voisinage construit par un ensemble de p variables prédictives mais suivant un principe de connexité : la 4-connexité (les points voisins de l'observation étudiée sont situés en haut, en bas, à droite et à gauche) et la 8-connexité (s'ajoutent aux voisins énoncés pour la 4-connexité les points en haut à gauche, en haut à droite, en bas à gauche et en bas à droite). D'autre part, les différents sites sont les n pixels de l'image et la valeur de la variable Y est le niveau de gris du pixel.

Prenons l'exemple d'une image structurée composée de 16×16 pixels (*cf.* figure 4.2). Cette image représente une forme oblique d'un gris moyen au centre, et les quatre coins de l'image sont plus (en haut) ou moins (en bas) foncé, plus (à droite) ou moins (à gauche) contrasté avec le motif gris central.

	4-connexité			8-connexité		
Bruit	I	$p\text{-}value_R$	$p\text{-}value_G$	I	$p\text{-}value_R$	$p\text{-}value_G$
0%	0,971	0	0	0,956	0	0
10%	0,678	0	0	0,659	0	0
20%	0,642	0	0	0,637	0	0
30%	0,273	$1{,}4\text{E}^{-9}$	$3{,}3\text{E}^{-9}$	0,283	$2{,}6\text{E}^{-18}$	$9{,}9\text{E}^{-17}$
40%	0,338	$7{,}0\text{E}^{-14}$	$2{,}7\text{E}^{-13}$	0,329	0	0
50%	0,161	$2{,}9\text{E}^{-4}$	$4{,}1\text{E}^{-4}$	0,174	$6{,}5\text{E}^{-8}$	$2{,}8\text{E}^{-7}$
60%	0,022	0,57	0,58	0,001	0,88	0,89
70%	0,047	0,26	0,28	0,058	0,06	0,07
80%	0,034	0,41	0,42	0,026	0,36	0,39
90%	-0,06	0,22	0,23	-0,081	0,02	0,03
100%	-0,039	0,45	0,46	0,009	0,70	0,71

TAB. 4.3 – Évolution du test de Moran global en fonction du bruit

4.3.3.2 Test de structure globale dans l'image

Dans le tableau 4.3, nous indiquons comment évolue la valeur du test de Moran global I et de son risque critique (*p-value*) en fonction du pourcentage de bruit ajoutée à l'image dans un schéma randomisé (R) ou gaussien (G). Le bruit consiste à changer la valeur d'un pixel par une valeur prise au hasard entre 0 et 255.

Nous remarquons que le risque critique, que ce soit dans les schémas gaussien ou randomisé, est significatif tant qu'il n'y a pas plus de 50% de bruit introduit dans l'image (le trait horizontal entre les lignes de 50% et 60% indique la fin de significativité du test de structure). Ce résultat statistique est à mettre en relation avec les 5 images de la figure 4.3 où celles-ci, bien que bruitées, laissent encore apparaître la forme structurée initiale (*cf.* figure 4.2). À partir de 60% de bruit, la forme est noyée par les valeurs bruitées et il est difficile de la retrouver dans l'image (*cf.* figure 4.4).

Nous notons qu'il y a une exception : avec un graphe en 8-connexité, pour 90% de bruit, le risque critique est significatif ($p\text{-}value < 0.05$). Dans ce cas, le coefficient de Moran est négatif, ce qui peut être interprété comme une structure particulière de l'image telle qu'une alternance de valeurs. Nous pouvons considérer ceci comme un artefact dû à notre manière d'introduire l'aléa dans l'image : nous avons en effet attribué aux points à modifier une nouvelle valeur prise au hasard parmi tout le domaine des valeurs possibles (de 0 à 255) alors que nous aurions pu procéder à des échanges ou permuta-

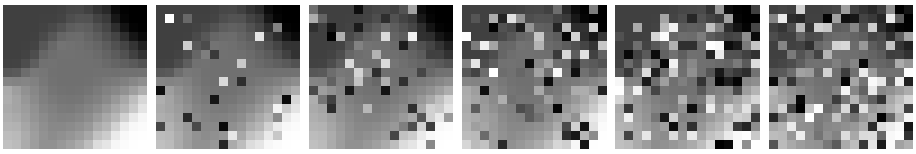

FIG. 4.3 – Images avec 0, 10, 20, 30, 40 et 50% de bruit

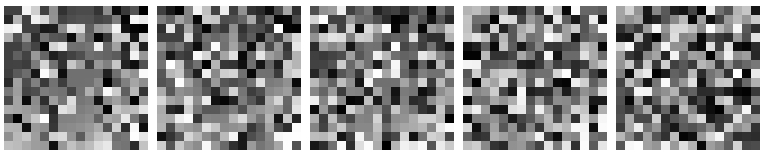

FIG. 4.4 – Images avec 60, 70, 80, 90, et 100% de bruit

tions des valeurs des pixels à modifier.

En outre, nous retrouvons un autre résultat attendu : le schéma gaussien produit des risques critiques plus élevés que dans le cadre aléatoire. Par conséquent, l'hypothèse nulle sera plus facilement rejetée dans le schéma gaussien que dans le schéma randomisé.

4.3.3.3 Détection d'*outliers* dans l'image

Anselin [Ans95] a mis en évidence une propriété intéressante du coefficient de Moran en l'associant à un diagramme de dispersion afin de visualiser les tendances locales.

Il suffit pour cela, sur chaque pixel i de l'image (ou plus généralement pour chaque exemple i de la base d'apprentissage), de représenter sur l'axe des abscisses la valeur de $Y(i)$ centrée et réduite et sur l'axe des ordonnées la valeur de la somme pondérée des valeurs de Y centrées et réduites du voisinage de i. Le coefficient global de Moran est la pente de la droite qui ajuste ce nuage de points au sens des moindres carrés.

Lorsqu'il y a une autocorrélation spatiale positive, comme c'est le cas dans la figure 4.5 pour le diagramme de dispersion des points de l'image non bruitée, la pente de la droite de régression est positive. En absence d'autocorrélation spatiale (*cf.* figure 4.6), la droite est éloignée des points et sa pente est quasi-nulle.

Le diagramme de dispersion permet ainsi de détecter les *outliers* : il s'agit des points fortement éloignés de la droite de régression donnée par le

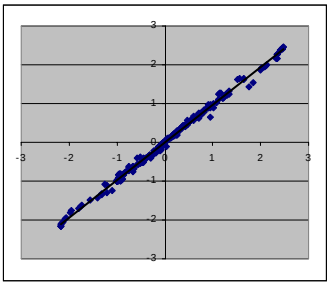

FIG. 4.5 – Diagramme de dispersion pour l'image non bruitée

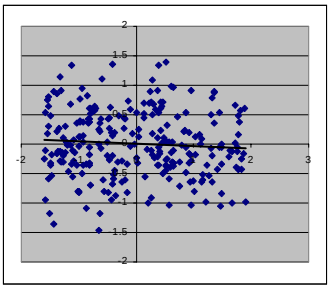

FIG. 4.6 – Diagramme de dispersion pour l'image totalement bruitée

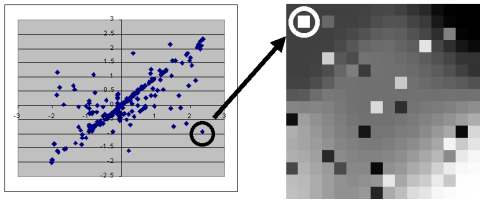

FIG. 4.7 – Détection des *outliers* à partir du diagramme de dispersion

coefficient de Moran. Sur la figure 4.7, nous indiquons comment détecter un *outlier* sur l'image de la forme dont 10% des pixels ont été bruités : l'exemple repéré comme très éloigné de la droite correspond bien à un pixel bruité, l'intensité de ce dernier contrastant d'ailleurs fortement avec les niveaux de gris de ses voisins.

4.4 Conclusion

Avec notre test fondé sur le coefficient de Moran et les graphes de voisinage, nous pouvons donc évaluer la qualité de l'espace de représentation issu de p variables prédictives dans le cas où la variable à apprendre est continue. Lorsque les valeurs de la variable à prédire se répartissent dans un espace structuré, nous pouvons supposer que des méthodes d'apprentissage fondées sur la distance pourront fournir un modèle prédictif fiable.

Un avantage de notre test est que, au lieu de procéder selon une proximité dans un plan ou un espace tridimensionnel comme cela est fait dans le cadre de l'autocorrélation spatiale, nous pouvons tenir compte d'un ensemble de variables prédictives à travers l'emploi des graphes de voisinage et ainsi appliquer notre test au domaine de l'apprentissage supervisé. L'usage des graphes de voisinage est une propriété dont nous avons déjà souligné l'intérêt car nous pouvons traiter des variables prédictives qui peuvent être aussi bien numériques que booléennes ou catégorielles, l'essentiel étant de rendre compte de l'effet relatif de chacun des facteurs dans le calcul de la matrice de distance entre les différentes observations de la base d'apprentissage.

Ajoutons cependant que le résultat de notre test de structure dépend de la qualité de la représentation globale issue de l'ensemble des variables prédictives. Par conséquent, la présence de variables prédictives non pertinentes va diminuer la significativité du test. Il est ainsi possible que certaines méthodes d'apprentissage supervisé fournissent quand même un bon modèle

prédictif d'une variable numérique malgré la présence de variables prédictives non significatives alors que le test n'indique pas la présence de structure. Ce phénomène, même s'il est absent des méthodes d'apprentissage fondées sur la distance qui tiennent compte de l'influence de toutes les variables prédictives (comme les méthodes *IBL* [AKA91]), se retrouve en particulier pour les méthodes d'apprentissage attribuant un poids différent à chacune des variables (par exemple en régression linéaire ou dans le cas des réseaux de neurones formels). Ainsi, en présence d'un mauvais espace de représentation dû à la présence de variables prédictives pertinentes dont l'effet est neutralisé par la présence d'autres variables non pertinentes, le test indiquera une absence de structure, ce qui nous renseignera sur le fait que tout apprentissage fondé sur la distance pour ces données est illusoire, mais cela ne voudra pas dire pour autant qu'une sélection des variables prédictives pertinentes [BL97] permettra d'obtenir un espace de représentation de meilleure qualité.

Enfin, nous indiquons que nous avons appliqué avec succès nos tests de structure aussi bien sous leur forme globale que locale dans le domaine du traitement d'image [LMJ03]. Nos tests ont en effet été intégrés en tant que paramètre de contrôle dans un processus de décimation, une méthode itérative qui permet de partitionner une image en régions homogènes. Lorsqu'on procède à une segmentation d'image afin d'isoler des scènes dans cette image, on peut employer une méthode de décimation itérative d'un graphe dont chaque nœud est associé à une valeur de pixel de l'image (le niveau de gris), les voisins étant définis suivant la 4-connexité (pixels situés au nord, au sud, à l'est et à l'ouest) ou la 8-connexité (N, NE, E, SE, S, SO, O, NO). La décimation fusionne des nœuds du graphe, changeant la valeur de certains pixels, ce qui permet de regrouper les régions voisines dans la nouvelle image construite. Le problème est que ce processus se poursuit jusqu'à n'avoir plus qu'un graphe à un seul nœud, ce qui revient à une image avec un seul niveau de gris. Aussi est-il important d'avoir un critère d'arrêt capable d'indiquer, à chaque étape, si les fusions locales se font entre nœuds similaires du point de vue de l'information portée (à savoir le niveau de gris).

Nous avons ainsi appliqué le test de Moran global pour indiquer à quel moment une étape de décimation produit une image non structurée, ce qui permet d'arrêter le processus de fusion et de conserver l'image considérée comme étant la mieux segmentée. Quant au test de structure locale, nous l'avons utilisé pour affiner ce processus de décimation, autorisant ou non des fusions locales afin de maintenir les détails fortement contrastés (tels que des lettres se détachant du fond).

Discrétisation de variables et fouille de données

Sommaire

Chapitre 5

Discrétisation de variables et fouille de données

Résumé

Nous proposons dans ce chapitre une méthode de discrétisation supervisée et polythétique des variables prédictives. Cette méthode, appelée *HyperCluster Finder*, utilise des amas issus d'un graphe de voisinage pour définir les bornes des intervalles sur chaque variable à discrétiser.

Nous présentons également une méthode d'apprentissage supervisé, intitulée *Data Squeezer*, qui procède par la généralisation des profils observés sur les données. Nous indiquons les avantages de cette approche, notamment dans les problèmes d'apprentissage où la variable à apprendre dépend de certaines interactions particulières entre les variables prédictives.

Data Squeezer, comme de nombreuses méthodes d'apprentissage, nécessite que ses variables prédictives soient catégorielles. Nous indiquons en quoi notre méthode d'apprentissage tire parti d'une discrétisation préalable par la méthode *HyperCluster Finder*.

5.1 Introduction

Lorsque nous avons présenté l'apprentissage à base d'exemples, nous avons indiqué que ce mode d'apprentissage permettait de traiter de manière préférentielle des variables prédictives numériques. Toutefois de nombreuses méthodes d'apprentissage supervisé, dans le domaine de l'intelligence artificielle et de l'ECD, nécessitent la présence de variables prédictives qualitatives

[Mic83]. Nous proposons dans ce chapitre une telle méthode d'apprentissage supervisé appelée *Data Squeezer* qui génère des règles de production par compression et qui est capable de traiter des problèmes d'apprentissage présentant des interactions entre variables prédictives [MZD01].

Cependant, une telle méthode nécessite l'emploi de variables prédictives catégorielles. Lorsque les variables présentes dans la base d'apprentissage se trouvent sous forme numérique, une transformation, connue sous le nom de *discrétisation*, est alors exigée. Or, si une discrétisation des variables prédictives est effectuée en tenant compte des informations de la variable à prédire et que cette dernière est la résultante d'interaction entre les variables prédictives, il est nécessaire de procéder à une discrétisation supervisée *polythétique*. Nous présentons ici une telle méthode de discrétisation, appelée *HyperCluster Finder*, qui emploie de façon originale les amas évoqués dans le chapitre 3 afin de procéder à des coupures d'intervalles sur chaque variable prédictive de manière à la fois supervisée et polythétique [MR02b, MR02a].

5.2 Discrétisation polythétique supervisée par recherche d'amas

5.2.1 Introduction

Nombreuses sont les méthodes d'apprentissage supervisé qui nécessitent des variables prédictives catégorielles pour pouvoir prédire l'étiquette d'une autre variable à laquelle elles sont liées de manière fonctionnelle [Mic83, Mit97]. Par conséquent, lorsque des bases présentent des variables prédictives numériques, il est nécessaire, avant la phase d'apprentissage, de ré-encoder chaque variable prédictive continue en variable catégorielle constituée d'un ensemble d'intervalles disjoints. Ce traitement est connu sous le nom de *discrétisation* [DKS95].

La discrétisation, en tant qu'étape préalable à l'apprentissage automatique [CGB94], est un domaine très étudié au sein de l'ECD [ZRR98]. Ainsi de nombreux auteurs considèrent qu'il est préférable de procéder à une discrétisation des variables prédictives même si la méthode d'apprentissage est capable de traiter directement des données numériques [FW99]. En effet, l'utilisation directe de variables numériques se heurte à des problèmes de coûts de calcul lorsque le processus d'apprentissage doit trier les données plus d'une fois, voire présuppose des hypothèses fort peu réalistes sur les données telles qu'une distribution normale dans le cas du bayésien naïf.

Cependant, l'étape de discrétisation doit être réalisée avec précaution puisque le changement de nature des données produit une nécessaire perte d'information. Cette perte doit être contrôlée avec soin car les performances du modèle construit par l'apprentissage supervisé dépendent en grande partie de la qualité du découpage effectué par la discrétisation [RSR96].

Classiquement, les méthodes de discrétisation sont identifiées selon deux critères :

1. la prise en compte ou non de la variable à prédire Y :
   - oui : discrétisation supervisée ;
   - non : discrétisation non supervisée ;
2. le mode de découpage des variables prédictives $X_1, X_2, ..., X_p$, :
   - effectué de manière individuelle : méthode monothétique ;
   - effectué globalement en tenant compte des interactions entre les variables prédictives : méthode polythétique.

Nous proposons dans ce chapitre une méthode originale de discrétisation supervisée polythétique [MR02b]. Cette méthode repose sur la notion d'amas et procède en deux étapes. Tout d'abord, à l'aide d'un graphe de voisinage, nous isolons des groupes d'individus de même étiquette dans des amas. Ensuite, nous procédons au découpage des variables prédictives $X_1, X_2, ..., X_p$ en projetant les extrémités des groupes ainsi construits sur chaque axe de l'espace de représentation. De la sorte, nous obtenons des bornes à partir desquelles les intervalles de discrétisation vont pouvoir être identifiés.

Par rapport aux méthodes non supervisées, cette approche a l'avantage de répondre directement à la problématique de l'apprentissage supervisé puisque, par construction, le découpage tient compte des valeurs prises par la variable à prédire Y. De plus, par rapport aux méthodes monothétiques, notre méthode tient compte des éventuelles interactions entre les variables prédictives et permet ainsi le traitement de problèmes complexes à apprendre tels que le « ou exclusif » (XOR) comme nous l'illustrerons en section 5.4.

5.2.2 Travaux réalisés dans le domaine de la discrétisation

La plupart des études réalisées dans le domaine de la discrétisation en apprentissage supervisé [DKS95] ont conclu qu'une discrétisation supervisée est préférable à une discrétisation non supervisée (telle qu'un découpage des intervalles de tailles égales ou une discrétisation en intervalles de fréquences égales).

Ces études, principalement empiriques, cherchent à évaluer les propriétés des méthodes de discrétisation en s'intéressant à la manière dont le découpage des variables prédictives continues est réalisé. La plupart des travaux récemment publiés portent sur les méthodes monothétiques, c'est-à-dire celles qui procèdent au découpage des variables prédictives indépendamment les unes des autres. Les débats critiques s'orientent ainsi sur la manière d'obtenir les meilleures frontières sur chaque variable.

Le principe des approches ascendantes [LS95] est de procéder au découpage initial des variables prédictives continues en autant d'intervalles qu'il existe de valeurs pour les différents exemples de l'échantillon d'apprentissage, puis ces méthodes cherchent à fusionner les intervalles voisins dont la distribution des étiquettes des exemples est semblable. Les approches descendantes [FI93] fonctionnent de manière opposée : elles considèrent que toutes les valeurs des variables prédictives constituent un seul intervalle et ajoutent progressivement des frontières afin de repérer des intervalles ne contenant que des exemples de la même étiquette. Les expériences indiquent toutefois que ces méthodes fonctionnent de manière équivalente en apprentissage supervisé. En outre, il arrive que des méthodes de découpage des intervalles, coûteuses en temps de calcul mais censées être optimales comparées à des méthodes de discrétisation plus basiques, ne donnent finalement pas de meilleurs résultats (estimés à travers la qualité de la prédiction obtenue par les modèles issus d'algorithmes d'apprentissage supervisé qui prennent en entrée ces variables prédictives discrétisées).

Les méthodes de discrétisation polythétiques sont très peu étudiées dans le cadre de l'apprentissage supervisé. Les travaux qui se sont intéressés aux méthodes polythétiques concernent essentiellement les règles d'association, en apprentissage non supervisé, lorsque les méthodes de discrétisation monothétiques et non supervisées ne parviennent pas à produire des résultats satisfaisants. Dans ce cas, chaque variable est découpée en relation avec les autres variables de la base de données mais, à la différence de l'apprentissage supervisé, aucune variable ne joue de rôle particulier.

Ainsi, Ludl et Widmer [LW00] proposent une approche de la discrétisation de variables dans le cadre des règles d'association qui peut être considérée comme « multi-supervisée ». Le découpage des variables X_i repose sur $(p-1)$ discrétisations monothétiques supervisées où la variable $X_{i'}$ (avec $i' \neq i$) joue le rôle de la variable prédictive. Ensuite, les intervalles jugés trop petits sont fusionnés. Cette méthode originale a cependant la faiblesse de devoir procéder, avant la discrétisation supervisée polythétique de chacune des variables X_i, à une discrétisation préalable non supervisée monothétique

de chaque variable continue $X_{i'}$ qui joue le rôle de la variable à prédire dans le découpage de la variable X_i, or une telle discrétisation comporte de nombreux problèmes tels que le choix subjectif du nombre d'intervalles à construire a priori.

À mi-chemin entre l'apprentissage supervisé et non supervisé, Bay suggère une approche qui consiste à réaliser un pavage de l'espace de représentation en découpant chaque variable continue en un ensemble d'intervalles, le nombre initial de ces intervalles étant un paramètre donné par l'utilisateur [Bay01]. La discrétisation procède ainsi par la fusion des intervalles adjacents dont la distribution des étiquettes des exemples est identique. Cependant, comme le test de fusion des intervalles utilisé est polythétique, toutes les variables jouent simultanément un rôle équivalent. Dans le cas où l'apprentissage est supervisé, Bay propose d'attribuer la variable à prédire Y en tant que variable complémentaire pour le test d'équivalence de distribution. Néanmoins, même si cette stratégie permet l'introduction d'une variable à prédire dans le processus de discrétisation, un rôle privilégié n'est pas accordé à celle-ci comparé aux variables prédictives, ce qui peut amener la discrétisation à des résultats non satisfaisants pour la prédiction.

Comparée aux approches existantes, notre méthode, en raison de son comportement à la fois supervisé et polythétique, combine leurs divers avantages. De par sa nature polythétique, *HyperCluster Finder* peut tenir compte des interactions entre les p variables prédictives puisque notre méthode repose sur la construction d'un graphe de voisinage dans $\mathbb{R}^p$. De plus, comme cette méthode de discrétisation est supervisée, elle attribue un rôle particulier à la variable à prédire Y et peut, de la sorte, déterminer le nombre d'intervalles de discrétisation le plus approprié.

5.2.3 Identification des amas dans l'espace de représentation

La méthode de discrétisation originale que nous allons exposer repose sur la notion d'amas dont nous avons donné la définition dans le chapitre 3. Nous rappelons que nous entendons par *amas* un sous-graphe connexe d'un graphe de voisinage (tel que le graphe des voisins relatifs de Toussaint [Tou80]) où tous les sommets de ce sous-graphe sont de la même étiquette.

L'identification des amas dans l'espace de représentation est réalisée en deux temps :

1. construction d'un graphe de voisinage ;
2. suppression des arêtes reliant des sommets d'étiquettes différentes.

Nous obtenons ainsi des sous-graphes connexes dont tous les éléments sont de la même étiquette.

5.2.4 Algorithme de discrétisation *HyperCluster Finder*

Notre méthode de discrétisation polythétique et supervisée procède par la recherche d'amas – ou plus exactement d'*hyperamas* car les points sont projetés dans un espace $\mathbb{R}^p$. Cette méthode, appelée *HyperCluster Finder*, décrite dans l'algorithme 12, se déroule de la manière suivante :

- projection des individus de Ω_a dans l'espace de représentation $\mathbb{R}^p$ (figure 5.1(a)) ;
- génération d'un graphe de voisinage, ici le graphe des voisins relatifs de Toussaint (figure 5.1(b)) ;
- coupure des arêtes reliant des points d'étiquettes différentes afin de constituer les amas (figure 5.1(c)) ;
- sélection des amas les plus pertinents (*cf.* section 5.2.5) ;
- recherche des valeurs minimales et maximales de chacun des amas sur les p variables continues ;
- utilisation de ces valeurs minimales et maximales pour définir des bornes sur chacune des p variables (figure 5.1(d)) ;
- pour chaque variable, le minimum et le maximum sont remplacés respectivement par moins et plus l'infini ;
- les valeurs de ces bornes délimitent un ensemble d'intervalles sur chaque variable ;
- les valeurs numériques des données sont ré-encodées par leur appartenance à un des intervalles obtenus.

Sur la figure 5.1(d), les bornes qui ont été trouvées ($d1_{X_1}$, $d2_{X_1}$ et d_{X_2}) permettent de définir les intervalles $]-\infty; d1_{X_1}[$, $[d1_{X_1}; d2_{X_1}[$ et $[d2_{X_1}; +\infty[$ pour la variable X_1 ainsi que $]-\infty; d_{X_2}[$ et $[d_{X_2}; +\infty[$ pour la variable X_2.

5.2.5 Limitation du nombre d'intervalles par sélection des amas pertinents

Afin d'éviter que des points isolés ne génèrent à eux seuls des amas, et donc des nouvelles bornes inutiles, il est possible de supprimer les amas considérés comme trop « petits » parmi les candidats à la constitution des intervalles. Une solution consiste à ne retenir que les amas dont le nombre d'individus est au moins égal à un nombre fixé a priori, ou un nombre dépendant de la taille de la base d'apprentissage ou, afin de tenir compte de

Algorithme 12 Méthode de discrétisation *HyperCluster Finder*

$G(\Sigma, A) \leftarrow GVR(\Omega_a)$
  {construction d'un graphe des voisins relatifs sur les exemples de Ω_a}
pour $\alpha \leftarrow 1$ **à** $(n_a - 1)$ **faire**
  pour $\beta \leftarrow \alpha$ **à** n_a **faire**
    si $Y(\alpha) \neq Y(\beta)$ **alors**
      $A \leftarrow A - (\alpha, \beta)$ {l'arête (α, β) est retirée du graphe}
    fin si
  fin pour
fin pour
$\aleph \leftarrow \{\omega_1\}$ {ω_1 est le premier exemple de Ω_a}
$\Upsilon_\aleph \leftarrow \{\aleph\}$ {l'ensemble des amas $\Upsilon_\aleph$ ne contient que l'amas $\aleph$}
$N_\aleph \leftarrow 1$ {$N_\aleph$ est le nombre d'amas}
pour $\alpha \leftarrow 2$ **à** n_a **faire**
  si $\alpha \notin \aleph, \forall \aleph \in \Upsilon_\aleph$ **alors**
    $N_\aleph \leftarrow N_\aleph + 1$ {création d'un nouvel amas}
    $\aleph \leftarrow \{\alpha\}$ {cet amas $\aleph$ comprend l'exemple α...}
    $\aleph \leftarrow \aleph \cup_{\beta=1}^{n_a} \{\beta\} \mid \text{graphe\_connexe}(\alpha, \beta)$ {...ainsi que tous les points β qui sont connexes au point α}
    $\Upsilon_\aleph \leftarrow \Upsilon_\aleph \cup \{\aleph\}$
  fin si
fin pour
$(\Upsilon'_\aleph, N'_\aleph) \leftarrow filtrage(\Upsilon_\aleph, N_\aleph)$
pour $i \leftarrow 1$ **à** p **faire**
  $Bornes(i) \leftarrow \varnothing$
  pour $\aleph \leftarrow 1$ **à** $N'_\aleph$ **faire**
    $Bornes(i) \leftarrow Bornes(i) \cup \{\min(X_i(\omega), \forall \omega \in \aleph)\}$
    $Bornes(i) \leftarrow Bornes(i) \cup \{\max(X_i(\omega), \forall \omega \in \aleph)\}$
  fin pour
  $Bornes(i) \leftarrow \text{Tri}(Bornes(i))$
  $Bornes_{min}(i) \leftarrow -\infty$
  $Bornes_{max}(i) \leftarrow +\infty$
fin pour
pour tout $\omega \in \Omega_a$ **faire**
  pour $i \leftarrow 1$ **à** p **faire**
    choisir $Borne_{inf}$ et $Borne_{sup} \in Bornes(i)$
    $\mid \{\{Borne_{sup} = Borne_{inf+1}\} \wedge \{Borne_{inf} \leqslant X_i(\omega) < Borne_{sup}\}\}$
    $X_i(\omega) \leftarrow [Borne_{inf}\,;\, Borne_{sup}[$ {ré-encodage}
  fin pour
fin pour

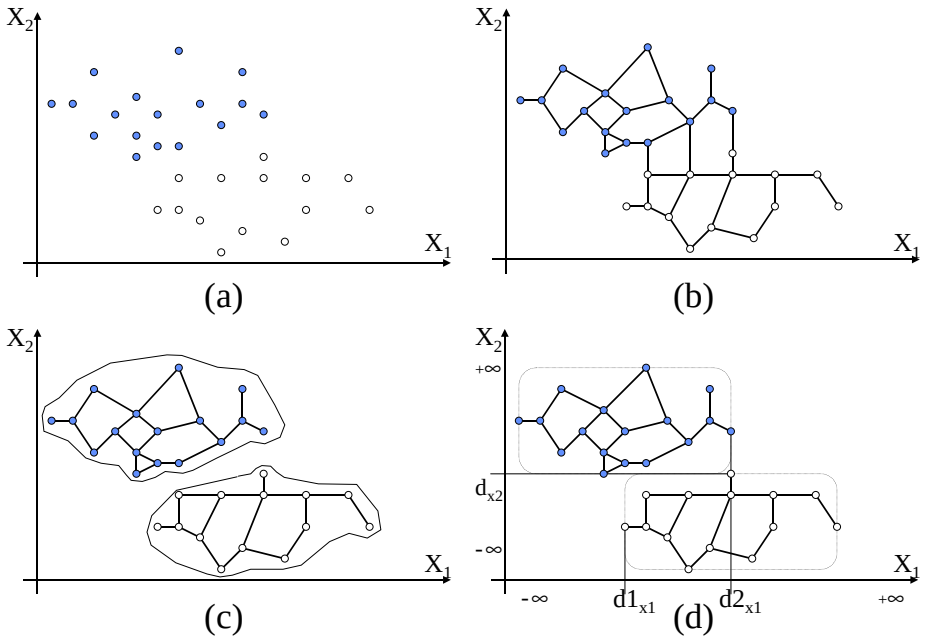

FIG. 5.1 – Méthode *HyperCluster Finder* : projection sur les axes X_1 et X_2 des bornes issues des amas

$X_1 \backslash X_2$	**0**	**1**
0	0	1
1	1	0

TAB. 5.1 – Table de vérité de la fonction logique XOR

manière plus fine de la structure de la base d'apprentissage, de la population du plus grand des amas obtenus.

5.2.6 Illustration sur un exemple en XOR numérique

Nous illustrons notre méthode en figure 5.2 à travers un exemple comprenant 100 individus décrits par 2 variables X_1 et X_2 continues. Les données, prenant des valeurs comprises entre -5 et $+5$ pour les deux variables, obéissent à une règle en XOR (*cf.* tableau 5.1), le *ou* exclusif : les individus ont l'étiquette *Vrai* (en noir) si $X_1 \geq 0$ et $X_2 < 0$ ou $X_1 < 0$ et $X_2 \geq 0$, et *Faux* (en blanc) sinon (voir figure 5.2(a)).

Sur les points projetés dans l'espace, nous construisons un graphe de voisinage (voir figure 5.2(b)). Les arêtes reliant des points d'étiquettes différentes sont coupées et les amas ainsi obtenus sont recherchés. Sur cet exemple, nous ne conservons que 4 amas (deux amas d'un seul individu ne sont pas retenus sur les 6 générés). Les valeurs minimales et maximales de chaque amas sont reportées sur les axes en figure 5.2(c). De cette manière, les intervalles formés pour la variable X_1 sont : $]-\infty;-0.3[$, $[-0.3;-0.1[$, $[-0.1;0.1[$, $[0.1;0.2[$, $[0.2;4.7[$ et $[4.7;+\infty[$.

Pour la variable X_2 : $]-\infty;-4.9[$, $[-4.9;-0.2[$, $[-0.2;0.0[$, $[0.0;0.2[$, $[0.2;4.7[$ et $[4.7;+\infty[$.

Ces intervalles, comme le montre la figure 5.2(d), permettent de réaliser un pavage de l'espace. Dans notre exemple, chaque pavé obtenu ne contient que des points d'une seule étiquette.

Après avoir remplacé les valeurs numériques des données par l'intervalle correspondant, la base peut ainsi être utilisée par des méthodes d'apprentissage supervisé nécessitant des variables prédictives catégorielles.

5.2.7 Bilan de la méthode *HyperCluster Finder*

La méthode de discrétisation *HyperCluster Finder* que nous avons présentée procède de manière polythétique et supervisée, ce qui permet en par-

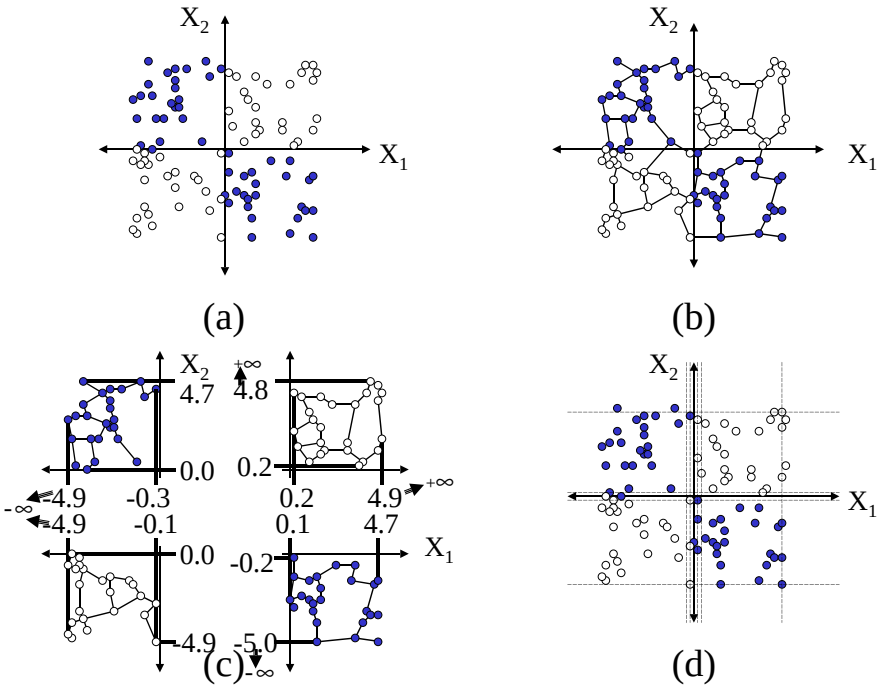

FIG. 5.2 – Méthode *HyperCluster Finder* appliquée sur une base d'apprentissage dont les valeurs quantitatives se répartissent selon la fonction logique XOR

ticulier de retrouver des intervalles adaptés au traitement des problèmes d'apprentissage comprenant des interactions entre les variables prédictives.

Sur notre exemple de données continues en XOR, la méthode *HyperCluster Finder* parvient à retrouver les bornes adéquates. Ne tenant pas compte de la valeur de l'étiquette Y, cette prouesse ne pourra être réalisée par des discrétisations non supervisées, à moins de tomber par hasard sur les bornes adéquates. Quant aux discrétisations supervisées monothétiques, ne pouvant discriminer les étiquettes sur chaque axe, elles conclurons que ni X_1 ni X_2 ne sont discrétisables sur l'exemple du XOR continu.

Nous faisons toutefois remarquer que le concept du XOR ne sera retrouvé que par des méthodes d'apprentissage qui procèdent de manière ascendante (partant des données particulières pour remonter aux concepts généraux) telles que l'algorithme AQ [Mic83].

Ainsi, les capacités de discrétisation de la méthode *HyperCluster Finder* ne peuvent être mises en évidence que si, au cours de la phase de fouille de données, la méthode d'apprentissage supervisé est en mesure d'en tirer profit. Une telle méthode d'apprentissage se devra donc d'être polythétique dans sa manière de traiter les différentes variables prédictives. Dans la section suivante, nous proposons une méthode d'apprentissage supervisé possédant ces caractéristiques.

5.3 Génération de règles par compression

5.3.1 Introduction

Dans le domaine de l'extraction des connaissances à partir de données, l'accent est mis sur la formalisation explicite des connaissances, notamment sous une forme de règles logico-déductives. Nous avons souligné dans le chapitre premier que ces connaissances ont en effet l'immense avantage d'être compréhensibles par l'esprit humain, ce qui n'est ni vraiment le cas pour des productions issues d'apprentissages automatiques par des réseaux de neurones [RM86], ni par des « séparateurs à vaste marge » (ou *SVM*, "*support vector machine*") [Vap98, SS02] ni même par les méthodes d'apprentissage à base d'exemples [AKA91].

Les arbres de décisions, au contraire, sont des méthodes d'apprentissage supervisé aboutissant à des connaissances exprimées sous forme de règles. Ils procèdent par l'application successive de critères sur une population d'apprentissage pour obtenir des ensembles de sous-populations et ceci est réalisé

afin d'isoler au sein de ces sous-groupes des individus présentant une seule étiquette. Cette manière de procéder est réalisée à travers la construction d'un arbre (ou, plus généralement, d'un graphe) et, à partir de cet arbre, les règles de production sont retrouvées en partant du sommet de l'arbre (la population de base) et en suivant le chemin pour arriver jusqu'aux feuilles (les divers groupes de populations où une étiquette ressort majoritairement), chacun des différents nœuds séparant les branches de l'arbre indiquant quels sont les critères appliqués pour aboutir à chacune des sous-populations.

Cependant il arrive que le gain obtenu en compréhension par la lisibilité d'un modèle sous forme de règles soit perdu par la génération d'un trop grand nombre de règles pour le problème étudié. Il existe ainsi des techniques permettant d'élaguer des branches de l'arbre de décision [BFOS84, Qui93b, ZR00] mais les résultats ne sont pas toujours satisfaisants.

Nous proposons ainsi une nouvelle méthode de génération de règles qui se différencie des arbres de décision en sa manière d'aborder le problème d'apprentissage [MZD01]. La création des règles dans un arbre de décision est en effet réalisée selon une approche *descendante* : elle part du sommet unique de l'arbre (la racine étant située par une curieuse convention de mathématiciens et informaticiens « en haut ») pour se développer vers le bas en appliquant de manière itérative différents critères constitués du choix d'une valeur particulière d'une variable prédictive X_i donnée. Ceci donne naissance à différentes branches, et l'application des critères est répétée jusqu'à aboutir à une feuille, c'est-à-dire une partie de la population sur laquelle il n'est plus intéressant (ou possible) d'appliquer un nouveau critère.

Dans la méthode que nous proposons, la démarche est *ascendante*, à la manière de la méthode de l'Étoile [Mic83]. Nous partons de groupes d'individus sur lesquels tous les critères disponibles sont appliqués : nous utilisons toutes les différentes modalités de chacune des p variables prédictives pour isoler des petits ensembles d'exemples de la base d'apprentissage. La production des règles est réalisée en forçant le regroupement de ces différents groupes d'individus et cette compression est effectuée en retirant certains critères. Les règles de production sont obtenues à partir des groupes d'individus issus de cette compression. Ainsi, contrairement aux arbres de décision qui procèdent par spécialisation, notre méthode procède par généralisation.

La méthode *Data Squeezer* que nous présentons ici est une méthode d'apprentissage supervisé originale qui doit son nom à une analogie faite avec un presse-citron ("*lemon squeezer*") car, à la manière dont cet instrument permet d'obtenir du jus en pressant la pulpe du fruit, notre méthode essaie d'extraire des connaissances exprimées sous forme de règles par compression

des données.

La compression opérée dans *Data Squeezer* est réalisée à travers une mesure d'incertitude : les ensembles d'individus constitués au départ sont regroupés dans le cas où ces regroupements amènent une diminution de l'incertitude globale.

5.3.2 Notations et définitions

Nous considérons une base d'apprentissage composée d'une population Ω de n exemples décrits par p variables prédictives $X_i, i = 1, .., p$ catégorielles. La variable à prédire Y est également catégorielle et un individu $\omega \in \Omega$ a comme étiquette $e = Y(\omega)$, une des valeurs de la variable à prédire parmi les r étiquettes différentes (ainsi $r = 2$ dans le cas où la variable à prédire Y est booléenne).

Les p variables prédictives étant qualitatives, nous avons pour chaque variable X_i un nombre fini de valeurs possibles.

Nous notons $\eta_i = Card(X_i(\Omega))$ le nombre de modalités possibles que peut avoir la variable prédictive X_i.

À chaque individu $\omega \in \Omega$ est associé un profil π correspondant à un ensemble de modalités particulières pour chacune des p variables prédictives. Suivant Zighed et Rakotomalala [ZR00], nous définissons les notions de *mintermes* pour rendre compte de manière plus précise de ces profils.

Définition 5.3.1 Minterme élémentaire. *Soit ω un individu de la population d'apprentissage Ω et π son profil, nous appelons minterme élémentaire, noté $X^{\diamond}(\pi)$, le profil observé de ω.*

Définition 5.3.2 Minterme généralisé. *Nous appelons minterme généralisé, ou plus simplement « minterme », noté $\boldsymbol{M}$, une réunion de mintermes élémentaires.*

$$\boldsymbol{M}(\pi, \pi') = X^{\diamond}(\pi) \cup X^{\diamond}(\pi') \tag{5.3.1}$$

Deux paramètres interviennent dans notre méthode d'apprentissage.

Le premier de ces paramètres, que nous appelons la « force de compression » et notons λ, est utilisé dans le calcul de l'incertitude (*cf.* équation 5.3.2, en page 146). Son rôle sera illustré en sous-section 5.3.4.3.

$X_1 \backslash X_2$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 0 0	0 0 0 0 0	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1	1	1 1 1
2	0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0	1 1 1	1 1 1 1	1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1 1
3	0 0	0 0 0 0	0 0 0	0 0 0 0 0	0 0 0 0 0 0	1 1 1 1 1 1 1	1	1 1 1 1 1 1 1 1 1	1 1 1	1 1
4	1 1	1 1 1 1		1 1 1	1 1 1 1 1 1 1	0 0	0 0 0 0		0 0 0	0
5	1 1 1 1	1 1	1	1 1 1	1	0 0 0 0	0 0 0	0 0	0 0	0 0 0 0
6	1 1 1 1 1	1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1	0 0 0 0 0 0 0 0		0 0 0 0 0 0 0	0 0 0	0 0 0 0

TAB. 5.2 – Tableau de contingence de la base de données en XOR catégoriel

$X_1 \backslash X_2$	**1 ; 2 ; 3 4 ; 5**	**6 ; 7 ; 8 ; 9 ; 10**
1 ; 2 ; 3	0	1
4 ; 5 ; 6	1	0

TAB. 5.3 – Représentation simplifiée de la base de données en XOR catégoriel

Le second paramètre de la méthode *Data Squeezer* est le « taux de spécification », noté ε. Nous utilisons le taux ε afin de déterminer la valeur prise par la conclusion d'une règle. Cet élément sera décrit en sous-section 5.3.3.5.

5.3.3 Étapes de la méthode *Data Squeezer*

5.3.3.1 Description du jeu de données

Nous illustrerons la manière dont procède la méthode *Data Squeezer* à travers une base d'exemples en XOR catégoriel dont nous avons représenté la répartition des effectifs sur le tableau 5.2.

Cet ensemble de données se répartit selon deux variables prédictives X_1 et X_2 et une variable à prédire Y pour laquelle il peut y avoir deux étiquettes : $e_1 = 0$ et $e_2 = 1$. Dans cet exemple, la variable X_1 comporte 6 modalités et la variable X_2 comporte 10 modalités. Ainsi sur le tableau 1, nous observons que pour la case correspondant à $X_1 = 1$ et $X_2 = 1$ tous les exemples appartiennent à l'étiquette e_1 et que nous avons 7 individus de cette catégorie (7 exemples « 0 » présents et aucun exemple « 1 »). Un minterme n'étant qu'une case de ce tableau, nous aurons 6×10 mintermes élémentaires dont certains peuvent être nuls (comme la case vide pour $X_1 = 4$ et $X_2 = 3$).

Les exemples de notre base suivent la fonction logique du XOR déjà présentée en tableau 5.1, comme l'indique le tableau 5.3, version schématique du tableau 5.2 où ne sont plus représentés les différents effectifs mais simplement les étiquettes majoritaires de chaque situation.

X_1	X_2	n_1	n_2	n_M	h
1	1	7	0	7	0,198
1	2	8	0	8	0,180
1	3	3	0	3	0,320
⋮	⋮	⋮	⋮	⋮	⋮
1	9	0	1	1	0,444
1	10	0	3	3	0,320
2	1	6	0	6	0,219
2	2	7	0	7	0,198
⋮	⋮	⋮	⋮	⋮	⋮
6	9	3	0	3	0,320
6	10	4	0	4	0,278

TAB. 5.4 – Tableau des mintermes avec des incertitudes calculées pour $\lambda = 1$

5.3.3.2 Constitution du tableau des mintermes et calcul des incertitudes

Nous disposons d'une population de n individus $\omega \in \Omega$ décrits par p variables prédictives catégorielles $X_1, X_2, ..., X_p$, et, à chaque individu ω, est associé une étiquette e_j (avec $j \in \{1\,;\, r\}$) qui est une valeur donnée de la variable à prédire ($e_j = Y(\omega)$).

Le tableau des mintermes (voir tableau 5.4) est construit avec $(p+r+2)$ valeurs en colonnes et z lignes. Les colonnes sont composées de la suite des p modalités η_i (avec $i \in \{1\,;\, p\}$) identifiant un minterme élémentaire, de la suite des r effectifs d'une étiquette donnée du profil observé (n_j étant le nombre d'individus de l'étiquette e_j pour le minterme considéré), du nombre total d'exemples présentant le profil du minterme $\mathbf{M}$ ($n_M = \sum_{j=1}^{r} n_j$), ainsi que d'une valeur d'incertitude h calculée suivant la formule 5.3.2. Notre tableau de mintermes comprend z lignes, avec $z \leqslant \sum_{i=1}^{p} \eta_i$ car nous ne tenons pas compte des mintermes élémentaires vides, c'est-à-dire ceux dont les effectifs sont nuls.

Ainsi, à partir de notre base de données XOR (tableau 5.2), nous obtenons un tableau de 57 mintermes car 3 mintermes élémentaires sont vides parmi les 60 (6 modalités de $X_1 \times 10$ modalités de X_2).

Pour chaque ligne du tableau des mintermes (tableau 5.4), nous calculons l'incertitude h. Ce calcul met en œuvre l'emploi d'un paramètre réel positif et non nul λ dont nous décrirons par la suite la propriété de force de

compression. La mesure d'incertitude h que nous utilisons repose sur l'entropie quadratique. Ainsi, pour un minterme $\mathbf{M}_k$ donné (k étant l'indice de ce minterme dans le tableau des mintermes), l'incertitude h_k sera donnée par l'équation 5.3.2.

$$h_k = \sum_{j=1}^{r} \left(\frac{n_{j,k} + \lambda}{n_{M,k} + r\lambda} \right) \left(1 - \frac{n_{j,k} + \lambda}{n_{M,k} + r\lambda} \right) \tag{5.3.2}$$

où $n_{M,k}$ est le nombre d'effectifs n_M pour le minterme de la ligne k.

Cette valeur d'incertitude h sera d'autant plus importante que le nombre d'effectifs se répartissant dans chacun des effectifs d'étiquettes sera semblable et l'incertitude diminuera jusqu'à devenir minimale si les effectifs d'un minterme sont tous d'une seule et même étiquette.

5.3.3.3 Compression des mintermes par regroupement global

À partir des valeurs d'incertitude h_k, nous pouvons calculer l'incertitude globale H d'ensemble de données selon la formule 5.3.3.

$$H = \sum_{k=1}^{z} \frac{n_{M,k}}{n} h_k \tag{5.3.3}$$

avec $n = \sum_{k=1}^{z} n_{M,k}$, somme des effectifs des différents mintermes, et donc nombre d'individus de Ω.

Pour chaque variable X_i nous sélectionnons deux modalités différentes $\mathscr{A}$ et $\mathscr{B}$ parmi les η_i possibles (par exemple, nous sélections les modalités $X_1 = 1$ et $X_1 = 2$ parmi les $\eta_{X_1} = 6$ possibles).

À chaque minterme composé de la modalité $\mathscr{A}$ (tel $\mathbf{X}_1^{\diamond}(X_1 = 1\,;\, X_2 = 2)$) nous ajoutons les effectifs des étiquettes pour le minterme similaire pourvu de la modalité $\mathscr{B}$ (ici $\mathbf{X}_{11}^{\diamond}(X_1 = 2\,;\, X_2 = 2)$. Nous calculons ensuite les diverses incertitudes locales issues de ces regroupements (soit $h_{1'}$ de $\mathbf{M}_{1'}(\{X_1 = 1\,;\, X_1 = 2\}\,;\, X_2 = 2)$ avec $\mathbf{M}_{1'} = \mathbf{X}_1^{\diamond} \cup \mathbf{X}_{11}^{\diamond}$). Lorsque ces nouvelles incertitudes locales sont calculées, nous calculons l'incertitude globale H'. Le tableau 5.5 présente les mintermes après un tel regroupement.

Nous effectuons les combinaisons de toutes les modalités possibles pour la variable X_i et n'en conservons que les meilleurs candidats, c'est-à-dire les deux modalités pour lesquelles la diminution d'incertitude globale est la plus forte. Nous opérons ensuite un traitement similaire sur les autres variables. Ce n'est qu'après avoir déterminé la variable où la diminution d'incertitude

X_1	X_2	n_1	n_2	n_M	h
{ 1 ; 2 }	1	13	0	13	0,124
{ 1 ; 2 }	2	15	0	15	0,111
{ 1 ; 2 }	3	11	0	11	0,142
⋮	⋮	⋮	⋮	⋮	⋮
6	9	3	0	3	0,320
6	10	4	0	4	0,278

TAB. 5.5 – Tableau des mintermes après le regroupement de $X_1 = 1$ avec $X_1 = 2$

globale est la plus forte que nous conservons les regroupements de mintermes pourvus des modalités de cette variable. Nous recommençons cette opération tant que des diminutions d'incertitude globale sont possibles et qu'il reste des modalités à regrouper.

Ce mode de compression permet de diminuer de façon importante le nombre de mintermes de notre tableau. Les regroupements s'opèrent de manière globale, ce qui permet d'éliminer aisément les variables non pertinentes (cas où pour une variable X_i il ne reste plus qu'un seul et même groupe de modalités pour tous les mintermes).

Avec ce procédé, dans notre exemple en XOR catégoriel, nous aboutissons aux mêmes regroupements que ceux présentés un peu plus tôt dans le tableau 5.3.

5.3.3.4 Compression des mintermes par regroupement local

Le mode de regroupement précédent permet une compression globale des mintermes selon deux valeurs prises par une modalité pour une variable donnée. Nous opérons ensuite une compression de détail en tentant de regrouper localement deux mintermes à la fois (et non plus tous les mintermes possédant deux modalités particulières). Les candidats à ce regroupement sont deux mintermes ne différant que selon une valeur pour une variable X_i donnée. Cette différence exceptée, le principe de la compression reste identique : les regroupements sont effectués si l'incertitude issue d'un regroupement H' est inférieure ou égale à l'incertitude H.

5.3.3.5 Génération des règles de production

Lorsque les phases de compression sont achevées, nous pouvons construire notre modèle. Chaque minterme présent dans le tableau issu de la compression se traduit en une règle de production à condition que le taux de succès de cette règle soit supérieur au taux de spécification ε choisi par l'utilisateur.

Nous transformons les informations portant sur les variables prédictives des mintermes en prémisses de règles et la conclusion est donnée à travers l'étiquette majoritaire du minterme en question. Quand, pour une variable X_i donnée, plusieurs modalités sont présentes en raison des regroupements, l'élément constitutif de la prémisse est donné par l'expression composée d'une suite de disjonctions qu'il est aussi possible d'écrire sous forme d'appartenance à un ensemble de valeurs. Dans le cas où sont présentes toutes les modalités possibles η_i de la variable prédictive X_i, celle-ci n'apparaît pas dans la règle.

La conclusion de la règle prend la valeur de l'étiquette e_ν pour laquelle il y a un maximum d'effectifs $n_{M,\nu}$, c'est-à-dire un vote à la majorité. Ainsi pour la règle issue d'un minterme $\mathbf{M}_k$, nous calculons le taux de succès τ_k de la règle par la proportion d'individus $n_{\nu,k}$ associée à l'étiquette majoritaire $e_{\nu,k}$ sur l'ensemble des individus n_k (*cf.* équation 5.3.4).

$$\tau_k = \frac{n_{\nu,k}}{n_k} \tag{5.3.4}$$

Si τ_k est inférieur au taux de spécification ε, la règle n'est pas considérée comme pertinente : elle est soit ôtée de l'ensemble des règles de production du modèle soit modifiée en remplaçant la valeur e_ν de la conclusion par la valeur spéciale *IND* afin de signaler que la conclusion de la règle reste indéterminée.

5.3.4 Illustration des performances de *Data Squeezer* sur des jeux de données

Nous présentons les résultats obtenus par la méthode *Data Squeezer* sur notre jeu de données en XOR catégoriel ainsi que sur un jeu de données test standard. Les performances de notre méthode sont comparées à des méthodes d'arbres de décision et de graphes d'induction : *CART* [BFOS84], *C4.5* [Qui93b], *ID3* [Qui86] et *Sipina* [ZR00].

Pour chaque méthode, nous indiquons les pourcentages de réponses correctes, incorrectes et indéterminées (valeurs *IND*) issus d'une validation croi-

Méthode	Sipina	CART	C4.5	ID3	Squeezer
Correct	54.66%	—	—	47.88%	100.00%
Incorrect	15.25%	—	—	42.80%	0.00%
Indéterminé	30.08%	—	—	9.32%	0.00%
Nb règles	35	0	0	10	4

TAB. 5.6 – Résultats obtenus sur le problème du XOR catégoriel

sée ainsi que le nombre moyen de règles obtenues.

5.3.4.1 Étude sur le jeu de données jouet en XOR catégoriel

La première série de tests effectuée a porté sur la base de données présentées en tableau 5.2 où les exemples se distribuent selon la fonction logique XOR. Notre population est constituée de 236 individus, avec 2 variables prédictives (X_1 et X_2) et 2 étiquettes à prédire (0 ou 1, soit *Faux* ou *Vrai*).

Le premier résultat que nous remarquons dans le tableau 5.6 est que, contrairement aux méthodes arborescentes, la méthode *Data Squeezer* permet d'apprendre correctement le modèle. En effet, à chaque fois, notre méthode extrait de la base d'apprentissage les 4 règles qui régissent le comportement des données se répartissant suivant la fonction logique XOR. Ces règles sont les suivantes :

Règle 1 : SI $X_1 \in \{ 1 ; 2 ; 3 \}$ **ET** $X_2 \in \{ 1 ; 2 ; 3 ; 4 ; 5 \}$ **ALORS** $Y = 0$
Règle 2 : SI $X_1 \in \{ 4 ; 5 ; 6 \}$ **ET** $X_2 \in \{ 1 ; 2 ; 3 ; 4 ; 5 \}$ **ALORS** $Y = 1$
Règle 3 : SI $X_1 \in \{ 1 ; 2 ; 3 \}$ **ET** $X_2 \in \{ 6 ; 7 ; 8 ; 9 ; 10 \}$ **ALORS** $Y = 1$
Règle 4 : SI $X_1 \in \{ 4 ; 5 ; 6 \}$ **ET** $X_2 \in \{ 6 ; 7 ; 8 ; 9 ; 10 \}$ **ALORS** $Y = 0$

Les autres méthodes échouent sur les données en XOR catégoriel : *Sipina* ou *ID3* ne donnent une prédiction correcte que pour environ 50% des exemples (ce qui est très faible car le nombre d'étiquettes $r = 2$), quant aux méthodes *CART* et *C4.5*, elles ne parviennent pas à construire d'arbre avec ces données.

5.3.4.2 Étude sur un fichier de données tests

Nous avons testé notre méthode d'apprentissage sur un fichier des bases de données provenant du serveur de l'Université de Californie d'Irvine (UCI) [BM98]. Il s'agit de "*Balance Scale Weight & Distance Database*", un fichier de 625 individus comprenant 4 variables prédictives avec pour chacune 5 modalités et 3 étiquettes différentes pour la variable à prédire (*balanced, left*

Méthode	Sipina	CART	C4.5	ID3	Squeezer	Squeezer
Correct	58.24%	62.24%	66.88%	62.72%	57.92%	73.28%
Incorrect	21.76%	33.28%	32.64%	23.36%	16.64%	24.16%
Indéterminé	20.00%	4.48%	0.48%	13.92%	25.44%	2.56%
Nb règles	65	41	21	73	20	14
λ	—	—	—	—	1	2

TAB. 5.7 – Résultats obtenus en généralisation avec le fichier Balance-Scale

et *right*). Ce fichier a été choisi car il ne présente que des variables prédictives catégorielles, ne souhaitant pas procéder à une étape de discrétisation préalable.

Nous avons effectué nos tests en utilisant pour la méthode *Data Squeezer* un taux de spécification $\varepsilon = 70\%$ et deux valeurs pour λ (1 et 2).

Sur le tableau 5.7, nous remarquons que selon la valeur de la force de compression λ et du nombre de règles qui en découle, notre méthode présente deux comportements distincts. Avec $\lambda = 1$, la méthode *Data Squeezer* a un faible taux d'exemples bien classés (résultat correct) mais produit le moins de résultats incorrects car de nombreuses réponses restent indéterminées. Avec $\lambda = 2$, la méthode *Data Squeezer* domine les autres méthodes sur le taux de bien classés mais produit aussi un grand nombre des réponses incorrectes.

5.3.4.3 Influence de la force de compression

La compression des données repose sur un calcul d'incertitude (équation 5.3.2) où intervient un paramètre λ que nous avons appelé « la force de compression ». En effet, suivant la valeur donnée à ce paramètre réel positif, la méthode *Data Squeezer* produit plus ou moins de règles comme l'indique la figure 5.3.

Selon la valeur de λ et l'incertitude calculée qui en découle, des regroupements sont considérés plus ou moins opportuns. Ce facteur agit donc sur le nombre de règles que produit notre méthode : plus le paramètre λ est important moins le nombre de règles générées est important et donc plus forte est notre compression.

Notre méthode est donc très sensible à cette force de compression et de son choix judicieux dépend la qualité du modèle produit par *Data Squeezer* : des règles produites en trop grand nombre risquent d'être trop spécifiques et de ne pas donner de bons taux de succès en généralisation, la méthode ayant

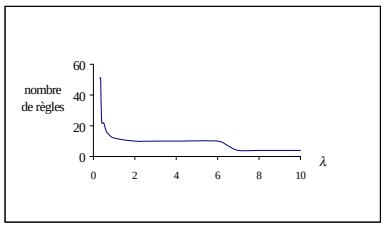

FIG. 5.3 – Évolution du nombre de règles produites en fonction de λ

procédé à un sur-apprentissage.

5.3.5 Bilan de la méthode *Data Squeezer*

En dehors du cas du XOR où *Data Squeezer* surpasse très significativement les méthodes arborescentes, les résultats que nous obtenons se situent dans le même ordre de grandeur que ces autres méthodes. Plutôt que des résultats incorrects, nous obtenons avec *Data Squeezer* des conclusions indéterminées. Il faut ainsi, suivant le cas, trouver un compromis entre la maximisation des résultats corrects qui risque de produire aussi des résultats incorrects et la minimisation des résultats incorrects qui risque de produire de nombreux résultats indéterminés. Ce compromis peut être obtenu en manipulant le paramètre λ indiquant la force de compression appliquée aux données et le paramètre ε précisant le taux de spécification des règles générées.

Notre méthode fonctionne bien pour des données redondantes puisqu'elles se retrouvent dans la même ligne du tableau de mintermes mais elle semble assez inappropriée dans le cas des bases présentant un très grand nombre de variables avec de nombreuses modalités.

Par ailleurs, la méthode que nous proposons procède suivant des regroupements ascendants contrairement aux méthodes arborescentes dont le fonctionnement est descendant. Le tableau de mintermes avant les phases de compression pourrait générer à lui seul un ensemble de règles constituant un modèle prédictif. Toutefois les règles générées selon ce modèle saturé seraient bien trop nombreuses avec, de plus, certaines variables non pertinentes. De plus, une règle pourrait être produite alors qu'elle ne concerne qu'un seul élément, or il n'est pas statistiquement plausible que d'un seul cas nous

puissions tirer une généralité.

Pour reprendre notre parallèle avec un presse-citron, sans la compression, nous aurions comme résultat un produit dilué (trop de règles) comprenant éventuellement des pépins (des règles issues de cas uniques inadéquates pour la généralisation ou des variables non pertinentes alourdissant inutilement le modèle).

Pour obtenir un produit concentré, nous cherchons à compresser ces règles potentielles au moyen du regroupement des mintermes ayant des répartitions d'étiquettes similaires, ce qui a pour effet :

- d'augmenter la taille des effectifs des étiquettes associés aux mintermes regroupés, par conséquent la règle produite sera renforcée car elle s'appliquera dans un plus grand nombre de cas ;
- de réduire le nombre de règles composant notre modèle prédictif.

De la sorte, nous essayons d'obtenir un modèle à la fois *pertinent* par ses bons scores obtenus en classement et *intéressant* d'un point de vue cognitif par une lecture aisée d'un nombre limité de règles de production.

5.4 Combinaison des méthodes *HyperCluster Finder* et *Data Squeezer*

5.4.1 Introduction

Évaluer les performances d'une méthode de discrétisation n'est pas une opération triviale. En effet, la discrétisation transforme un ensemble de variables numériques en variables catégorielles, et l'exactitude de cette transformation ne peut pas s'exprimer simplement à travers un taux d'erreur comme cela est fait lors de l'évaluation des performances d'une méthode d'apprentissage. Il est certes possible d'observer sur des cas particuliers comment les bornes de coupure des intervalles sont retrouvées ou de s'intéresser à la distribution des étiquettes des exemples au sein de chaque intervalle mais ces études sont difficiles à mener lorsqu'une comparaison entre différentes méthodes de discrétisation est souhaitée.

Les expérimentations que nous avons menées ont donc consisté à combiner la méthode de discrétisation supervisée polythétique *HyperCluster Finder* à différentes méthodes d'apprentissage supervisé à base de modèles telles que notre méthode de génération de règles par compression *Data Squeezer*, des arbres de décision ainsi qu'un graphe d'induction et à étudier les résultats, exprimés en taux d'erreur en généralisation, obtenus par ces méthodes d'ap-

prentissage [MR02a]. De plus, pour mieux comprendre le comportement de la méthode *HyperCluster Finder* et les avantages que peuvent en retirer les méthodes d'apprentissage, nous avons comparé ses performances à celles d'une autre méthode de discrétisation dans diverses situations expérimentales.

5.4.2 Protocole expérimental

5.4.2.1 Introduction

Le protocole expérimental employé consiste à manipuler trois paramètres :

- une base d'apprentissage dont les variables prédictives sont numériques et dont la variable à prédire est catégorielle ;
- une méthode de discrétisation qui rendra les variables prédictives catégorielles ;
- une méthode d'apprentissage supervisé utilisant des variables prédictives catégorielles qui fournira un modèle de la base à apprendre et dont la qualité du modèle sera estimée par un taux d'erreur en généralisation.

5.4.2.2 Méthodes de discrétisation

Nous écartons de notre étude les méthodes de discrétisation non supervisée, qu'elles procèdent par découpage en intervalles de tailles égales ou en intervalles de fréquences égales. En effet, ces méthodes dépendent d'un paramètre à fournir a priori sur la base d'apprentissage : le nombre d'intervalles de discrétisation. Du choix de ce paramètre dépend la qualité de la discrétisation :

- si trop d'intervalles sont générés, cela gênera les algorithmes d'apprentissage qui risquent de produire un modèle trop complexe,
- si trop peu d'intervalles sont produits, ces derniers ne seront pas à même de rendre compte de la richesse des données parce qu'ils contiendront en leur sein des exemples de différentes étiquettes ; par conséquent les algorithmes d'apprentissage ne parviendront pas à faire un classement des exemples selon les étiquettes à partir de ces différents intervalles et donc à fournir un modèle fiable des données.

Nous avons ainsi comparé les performances de notre méthode *HyperCluster Finder* (*HCF*) à *Fusinter*, méthode de discrétisation supervisée qui a elle-même été comparée à d'autres méthodes de discrétisation supervisée et non supervisée de la littérature [RSR96] et pour laquelle les résultats sont équivalents aux meilleurs algorithmes de discrétisation connus jusqu'alors

tels que la méthode *MDLPC* ("*Minimum Description Length Principle Criterion*") [FI93].

5.4.2.3 Méthodes d'apprentissage

Nous avons utilisé 4 méthodes d'apprentissage supervisé qui prennent en entrée des variables prédictives catégorielles :

- *ID3*, un arbre de décision [Qui86] ;
- *C4.5*, un arbre de décision avec élagage [Qui93b] ;
- *Sipina*, un graphe d'induction [ZR00] ;
- *Data Squeezer*, une méthode procédant par généralisation [MZD01].

Les trois premières méthodes sont des méthodes descendantes et procèdent par spécialisation : l'ensemble des données est progressivement découpé par l'emploi de certains critères (le choix d'une modalité particulière d'une variable prédictive) afin d'isoler les exemples d'une étiquette donnée.

Par rapport à la méthode *ID3*, *C4.5* supprime du modèle certaines branches afin de produire un arbre plus compact et donnant de meilleures capacités de généralisation.

La méthode *Sipina* est un graphe d'induction : elle présente les mêmes propriétés qu'un arbre de décision mais est plus puissante en sa capacité de représenter des formes disjonctives par la fusion de certaines branches [Oli93].

Quant à la méthode *Data Squeezer*, nous avons décrit son fonctionnement par généralisation dans la section 5.3.

Les performances de chacun des modèles obtenus par ces méthodes d'apprentissage sont évaluées à travers une validation croisée en 10 parties.

5.4.2.4 Bases de test

La première base que nous avons utilisée pour nos tests est celle qui nous a servi à illustrer la méthode de discrétisation *HyperCluster Finder*. Il s'agit de la base d'apprentissage appelée « XOR (numérique) pur » illustrée en figure 5.4. Les 100 exemples de cette base peuvent prendre deux étiquettes (*Vrai* ou *Faux*) et sont décrits par deux variables prédictives X_1 et X_2. La variable à prédire suit la fonction du *XOR* : étiquette *Vrai* (en noir) si $X_1 \geq 0$ et $X_2 < 0$ ou $X_1 < 0$ et $X_2 \geq 0$, et étiquette *Faux* (en gris clair) si ce n'est pas le cas.

La seconde base employée est appelée « XOR avec recouvrement » (*cf.*

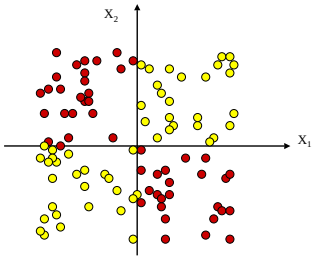

FIG. 5.4 – Base d'apprentissage *XOR pur*

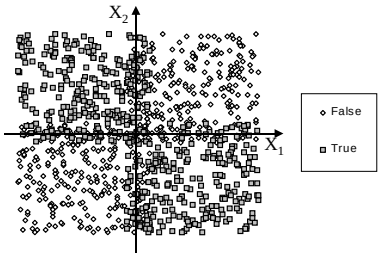

FIG. 5.5 – Base d'apprentissage *XOR avec recouvrement*

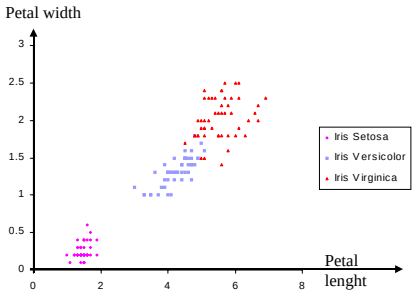

FIG. 5.6 – Base d'apprentissage *Iris*

Méthode de discrétisation	Méthode d'apprentissage	XOR pur	XOR avec recouvrement	Iris Plants
Fusinter	ID3	0,609	0,198	0,046
Fusinter	C4.5	0,616	0,543	0,054
Fusinter	Sipina	0,225	0,182	0,057
Fusinter	Data Squeezer	0,500	0,500	0,080
HCF	ID3	0,608	0,537	0,061
HCF	C4.5	0,117	0,182	0,053
HCF	Sipina	0,608	0,539	0,307
HCF	Data Squeezer	0,050	0,194	0,110

TAB. 5.8 – Taux d'erreur des méthodes d'apprentissage en validation croisée

figure 5.5). Nous avons ici 1000 exemples décrits par deux variables prédictives numériques X_1 et X_2. Comme pour la base d'apprentissage précédente, la valeur de la variable à prédire suit la fonction logique du *XOR* mais nous avons en plus un recouvrement des données sur chacun des axes.

La troisième base employée pour nos tests est la base *Iris Plants* du site de l'Université de Californie à Irvine [BM98]. Cette base d'apprentissage comprend 150 individus de trois étiquettes différentes (*setosa*, *versicolor* et *virginica*) décrits par 4 variables prédictives numériques (*petal width*, *petal length*, *sepal width* et *sepal length*). Nous présentons en figure 5.6 la répartition des exemples de la base *Iris* suivant les deux variables prédictives les plus discriminantes.

5.4.3 Résultats et discussion

Les résultats de la combinaison de chacune des méthodes de discrétisation (*Fusinter* et *HyperCluster Finder*), d'apprentissage supervisé (*ID3*, *C4.5*, *Sipina* et *Data Squeezer*) pour chaque base d'apprentissage (*XOR pur*, *XOR avec recouvrement* et *Iris*) sont présentés dans le tableau 5.8. Ces résultats concernent les taux d'erreur en généralisation des méthodes d'apprentissage pour une validation croisée en 10 parties après avoir procédé à la discrétisation des exemples sur chacun des $9/10^{\text{èmes}}$ de la base servant à l'apprentissage.

La première chose que nous observons dans le tableau 5.8 est que l'action combinée de *Data Squeezer* + *HyperCluster Finder* (*HCF*) donne les meilleurs résultats pour la base en XOR pur (avec seulement 5% d'erreur). Alors que la méthode d'apprentissage *ID3* échoue quelle que soit la méthode

de discrétisation préalable, la méthode *C4.5* tire parti de la discrétisation supervisée polythétique, au contraire de la méthode *Sipina* qui présente de meilleurs résultats avec la discrétisation *Fusinter* que *HyperCluster Finder*.

Nous précisons que la méthode *ID3* procède à un pré-élagage pour déterminer la taille de l'arbre de décision et, de la sorte, ne parvient pas à détecter les interactions entre les variables prédictives. Quant à la méthode *C4.5*, elle adopte un fonctionnement exploratoire en « franchissement d'obstacles » ("*hurdling*") et fait croître l'arbre en ajoutant une nouvelle branche même si le gain informationnel est nul. Ce n'est qu'au cours de la phase de post-élagage que la taille adéquate de l'arbre est recherchée, ce qui permet théoriquement de mieux traiter les interactions entre les variables prédictives [Qui93b]. Ces propriétés se trouvent confirmées empiriquement avec nos résultats.

Lorsque la base *XOR* comporte un recouvrement, les méthodes *C4.5* et *Data Squeezer* sont meilleures avec la discrétisation *HyperCluster Finder* que *Fusinter*. Pour la base *Iris*, les résultats sont équivalent pour les arbres de décision avec les deux méthodes de discrétisation mais les performances sont moindres pour le graphe d'induction *Sipina* avec la méthode de discrétisation *HyperCluster Finder*.

Avec les bases *XOR pur* et *XOR avec recouvrement*, la méthode *Data Squeezer* employée après *Fusinter* échoue à trouver un modèle prédictif et ne propose qu'une seule règle pour prédire l'étiquette à apprendre.

C4.5 semble avoir un comportement similaire à la méthode *Data Squeezer* et bénéficie de la discrétisation supervisée. La méthode *Sipina*, au contraire, apparaît plus adaptée à la discrétisation *Fusinter*. La méthode *ID3* semble jouer un rôle intermédiaire car, en dehors de la base *XOR pur* avec après une discrétisation avec *Fusinter*, les résultats sont proches de ceux de la méthode *Sipina*.

Nous pouvons expliquer ces résultats par la nature des méthodes d'apprentissage associées aux méthodes de discrétisation. La méthode *ID3* exploite la plus élémentaire des stratégies pour construire l'arbre d'induction. L'algorithme de *C4.5* est fondé sur celui de *ID3* mais est plus élaboré avec son post-élagage, ce qui permet d'obtenir des modèles moins complexes qu'*ID3*. *Sipina* est un graphe d'induction qui produit des graphes complexes capables d'apprendre plus ou moins bien le concept du *XOR* avec une discrétisation supervisée monothétique. Mais cette méthode produit un modèle trop complexe et a de mauvaises performances avec la discrétisation supervisée polythétique *HyperCluster Finder* en particulier lorsque celle-ci génère de nom-

breux intervalles de discrétisation et donc de nombreuses modalités pour les variables prédictives.

5.5 Conclusion

Par l'emploi d'un graphe de voisinage capable de rendre compte des effets combinés de l'ensemble des variables prédictives, notre méthode de discrétisation supervisée polythétique *HyperCluster Finder* est tout à fait adaptée aux méthodes d'apprentissage gérant les interactions entre variables prédictives comme le sont notre méthode *Data Squeezer* et, dans une moindre mesure, l'arbre de décision *C4.5*.

Ajoutons qu'une amélioration peut être apportée à la méthode de discrétisation *HyperCluster Finder* pour tenir compte de l'effet de toutes les variables prédictives. En effet, lorsque des variables catégorielles sont présentes, au lieu de les écarter ou de les transformer temporairement sous forme disjonctive complète, nous pouvons utiliser la métrique de différences de valeurs de Stanfill et Waltz [SW86] que nous avons présentée dans la sous-section 2.2.4.5 en page 60. Ce ré-encodage des variables prédictives catégorielles sous forme numérique est intéressant car il est réalisé de manière supervisée (les valeurs numériques remplaçant les modalités tenant compte de la distribution des différentes étiquettes). Cette transformation ne doit cependant durer que le temps de la création du graphe de voisinage et de la constitution des amas car, après cette phase, les variables prédictives doivent retrouver leurs différentes modalités d'origine.

Nous indiquons en outre que la méthode de discrétisation *HyperCluster Finder*, dont la complexité algorithmique est en $O(n^3)$ en raison de la création du graphe de voisinage, ne nécessite pas d'être opéré sur tous les exemples de la base d'apprentissage. La recherche des bornes de discrétisation peut être effectuée sur un échantillon limité mais représentatif de la base. Lorsque les intervalles sont retrouvés pour les diverses variables prédictives catégorielles à partir de cet échantillon, le ré-encodage de l'ensemble des valeurs numériques de la totalité des exemples de la base d'apprentissage peut être opéré.

Conclusion générale

ΑΓΕΩΜΕΤΡΗΤΟΣ ΜΗΔΕΙΣ ΕΙΣΙΤΩ

Sur le fronton de l'Académie, la fameuse école d'Athènes qui avait ouvert ses portes pendant près de mille ans à partir du IV^{e} siècle av. J.-C., étaient gravés ces trois mots que l'on peut traduire par « *que nul ne pénètre ici s'il n'est géomètre* ». Par cette phrase, Platon, son fondateur, voulait indiquer qu'avant de pouvoir faire de la philosophie, il fallait faire de la géométrie, ou plus largement des mathématiques, car ces disciplines sont le premier degré de l'intelligible et invitent à raisonner sur des réalités non sensibles.

Reprenant cette idée antique, nous nous sommes intéressés à certains problèmes rencontrés dans le domaine de l'extraction des connaissances à partir de données et nous nous sommes appuyés sur des outils géométriques afin de développer un ensemble de tests et méthodes apportant des réponses à ces problèmes.

Nous positionnant dans une filiation tenant à la fois de l'approche des sciences cognitives et de celle de l'extraction des connaissances à partir de données, nous avons souhaité, au cours de cette thèse, mettre l'accent sur la notion de connaissance.

L'objectif principal de notre travail a concerné l'évaluation de la qualité de la représentation. Pour ce faire, nous avons employé les graphes de voisinage dont nous avons exploité les propriétés pour indiquer la similarité existant entre des exemples décrits par un ensemble de variables prédictives.

Nous avons tout d'abord présenté notre enracinement scientifique, indiquant nos préoccupations au sein de la problématique de l'apprentissage supervisé et insistant sur la notion de *connaissance*. Nous avons exposé une vue synthétique des algorithmes d'apprentissage à base d'exemples, méthodes qui, au lieu de chercher à extraire une connaissance abstraite à partir des

données, conservent des éléments de connaissance à travers le stockage de certains individus de la base d'apprentissage et qui utilisent ces exemples pour prédire la valeur inconnue des exemples non appris. Nous avons ensuite décrit les principaux graphes de voisinage existant dans la littérature, ces outils géométriques mettant en avant la proximité entre des individus décrits par un ensemble de variables en les reliant par une arête.

Nous avons poursuivi ce premier volet consacré à l'état de l'art au cours du deuxième chapitre où nous avons exposé les mesures de distance et les indices de similarité les plus classiques. Fort de notre compréhension de la construction des graphes de voisinage et des manières de rendre compte de la similarité existant entre des observations, il nous a été possible de donner une estimation de la distance présente entre des exemples lorsque ceux-ci sont projetés dans un espace de représentation multidimensionnel tenant compte de l'ensemble des variables prédictives caractérisant ces exemples, que ces variables soient numériques, booléennes ou catégorielles.

Le deuxième volet sur lequel s'est articulé cette thèse concernait l'exploitation des outils présentés dans les chapitres précédents pour proposer une évaluation de la qualité de la représentation lorsque les variables à prédire sont catégorielles ou numériques. Plus précisément, dans le chapitre 3, nous nous étions situés dans le cas de l'apprentissage supervisé d'une variable catégorielle et avons exposé un test de séparabilité des étiquettes appelé le *test du poids des arêtes coupées* qui donne une information a priori sur la possibilité, pour une base d'exemples, d'être apprise par des méthodes fondées sur la distance. Nous avons également donné une version locale du test du poids des arêtes coupées destinée à identifier des exemples présentant du bruit sur la variable à prédire appelés *outliers* et avons proposé deux stratégies (le filtrage et le réétiquetage) pour traiter ces données afin d'améliorer les performances en généralisation des méthodes d'apprentissage utilisant ces bases bruitées.

Nous avons généralisé ces travaux sur l'évaluation de la qualité de la représentation dans le chapitre 4 en les appliquant au cas de l'apprentissage supervisé d'une variable numérique. Nous étant inspiré des travaux réalisés en analyse spatiale, nous avons ainsi proposé un *test de structure* qui indique si les valeurs d'une variable à prédire se distribuent dans l'espace de représentation suivant une certaine organisation. Nous avons ensuite illustré, dans le domaine du traitement d'image, comment procéder à la détection d'*outliers* numériques (des pixels bruités).

Le dernier volet de cette thèse consistait en une extension des réflexions engagées dans les chapitres précédents. Alors que nous nous étions situés jus-

qu'à présent dans le cadre des apprentissages supervisés dont les variables prédictives étaient continues ou rendues numériques d'une manière où d'une autre pour pouvoir représenter les exemples à apprendre dans un espace multidimensionnel, nous nous sommes intéressés dans le chapitre 5 aux méthodes d'apprentissage supervisé qui manipulent des variables prédictives catégorielles. Nous avons ainsi proposé une méthode de discrétisation supervisée et polythétique des variables prédictives, appelée *HyperCluster Finder*, qui repose sur la construction d'un graphe de voisinage et l'utilisation d'amas qui en découlent pour définir les bornes des intervalles sur chaque variable à discrétiser. Les performances de cette méthode de discrétisation, et en particulier ses capacités à traiter les interactions entre les variables prédictives, ont été mises en évidence en l'utilisant avec une nouvelle méthode d'apprentissage supervisé que nous avons appelée *Data Squeezer*. Nous avons décrit dans le chapitre 5 cette méthode d'apprentissage, illustrant le fait que cet algorithme procède par la généralisation des profils observés parmi les exemples d'apprentissage et fournit ainsi un modèle des données exprimé sous forme de règles.

Vu dans sa globalité, notre travail a donc consisté à partir des connaissances pour nous intéresser à la qualité d'un espace de représentation dans la problématique de l'apprentissage supervisé. Nous avons proposé des tests étudiant l'organisation des exemples dans un espace de représentation afin de savoir si les variables prédictives sont à même de fournir, avec des algorithmes d'apprentissage fondé sur la distance, un modèle prédictif d'une variable qualitative ou numérique. Les graphes de voisinage, véritable fil conducteur de nos réflexions, nous ont aussi servi à développer une méthode de discrétisation et nous avons enfin proposé une méthode d'apprentissage cherchant à tirer profit d'une telle discrétisation, cet algorithme générant un modèle sous forme de règles. De la sorte, bouclant la boucle, les diverses réponses que nous avons tenté d'apporter nous ont reconduit à la préoccupation qui avait motivé nos travaux, à savoir arriver à une méthode de fouille de données exprimant la connaissance sous une forme qui fût pertinente et qui satisfît autant notre personnalité de *cogniticien* que de "*data miner*".

Bibliographie

[AB95] Aha D.W., Bankert R.L., « A comparative evaluation of sequential feature selection algorithms », in *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pp. 1–7, Ft. Lauderdale, FL. 1995.

[ADZ00a] Auray J.P., Duru G., Zighed D.A., *Analyse des données multidimensionnelles*, volume 1 : Les méthodes de description, A. Lacassagne. 2000.

[ADZ00b] Auray J.P., Duru G., Zighed D.A., *Analyse des données multidimensionnelles*, volume 2 : Les méthodes de structuration, A. Lacassagne. 2000.

[ADZ00c] Auray J.P., Duru G., Zighed D.A., *Analyse des données multidimensionnelles*, volume 3 : Les méthodes d'explication, A. Lacassagne. 2000.

[AEM86] Aivazian S., Enukov I., Mechalkine L., *Eléments de modélisation et traitement primaire des données*, Moscou : MIR. 1986.

[Aha97] Aha D.W., « Editorial on Lazy Learning », *Artificial Intelligence Review*, 11 :7–10. 1997.

[AIS93] Agrawal R., Imielinski T., Swami A., « Mining Associations between Sets of Items in Massive Databases », in *Proceedings of the ACM-SIGMOD 1993 International Conference on Management of Data, Washington D.C.*, pp. 207–216. May 1993.

[AKA91] Aha D.W., Kibler D., Albert M.K., « Instance-based learning algorithms », *Machine Learning*, 6 :37–66. 1991.

[AMS97] Atkenson G.C., Moore A.W., Schaal S., « Locally weighted learning », *AI Review*, 11(1/5) :11–73. 1997.

[Ans95] Anselin L., « Local indicators of spatial association, LISA », *Geographical Analysis*, 27 :93–115. 1995.

[AS93] Alliot J.M., Schiex T., *Intelligence artificielle et informatique théorique*, Toulouse : Cépaduès. 1993.

[Bay01] Bay S.D., « Multivariate Discretization for Set Mining », *Knowledge and Information Systems*, 3(4) :491–512. 2001.

[BC83] Beckman R.J., Cooks R.D., « Outlier...s », *Technometrics*, 25 :119–149. 1983.

[Ben73] Benzécri J.P., *L'analyse des données*, Paris : Dunod, Tome 1 : La Taxinomie. Tome 2 : L'Analyse des Correspondances. 1973.

[BF96] Brodley C.E., Friedl M.A., « Identifying and Eliminating Mislabeled Training Instances », in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 799–805, Portland, OR : AAI Press. 1996.

[BF99] Brodley C.E., Friedl M.A., « Identifying mislabeled training data », *Journal of Artificial Intelligence Research*, 11 :131–167. 1999.

[BFOS84] Breiman L., Friedman J.H., Olshen R.A., Stone C.J., *Classification and regression trees*, Belmont, CA : Wadsworth International Group. 1984.

[BG88] Barthélemy J.P., Guénoche A., *Les arbres et les représentations des proximités*, Paris : Masson. 1988.

[BKW80] Belsley D.A., Kuh E., Welsch R.E., *Regression Diagnostics – Identifying Influential Data and Sources of Collinearity*, New-York : Wiley. 1980.

[BL84] Barnett V., Lewis T., *Outliers in statistical data*, Norwich : Wiley, 2nde édition. 1984.

[BL97] Blum A., Langley P., « Selection of Relevant Features and Examples in Machine Learning », *Artificial Intelligence*, 97(1–2) :245–271. 1997.

[BM98] Blake C.L., Merz C.J., « UCI Repository of machine learning databases », Irvine, CA : University of California, Department of Information and Computer Science [http ://www.ics.uci.edu/∼mlearn/MLRepository.html]. 1998.

[BS02] Bouroche J.M., Saporta G., *L'Analyse des Données*, Collection Que Sais-Je ?, Paris : Presses Universitaires de France, 1ère édition : 1980. 2002.

[CGB94] Chmielewski M.R., Grzymala-Busse J.W., « Global discretization of continuous attributes as preprocessing for machine

learning », in *Third International Workshop on Rough Sets and Soft Computing*, pp. 294–301. 1994.

[CH67] Cover T.M., Hart P.E., « Nearest neighbor pattern classification », *IEEE Transactions on Information Theory*, 13 :21–27. 1967.

[CO86] Cliff A.D., Ord J.K., *Spatial processes, models and applications*, London : Pion Limited. 1986.

[CP81] Chandon J.L., Pinson S., *Analyse Typologique, Théories et Applications*, Masson. 1981.

[CS93] Cost S., Salzberg S., « A weighted nearest neighbor algorithm for learning with symbolic features », *Machine Learning*, 10 :57–78. 1993.

[DHS00] Duda R.O., Hart P.E., Stork D.G., *Pattern Classification*, Wiley, 2nde édition, 1ère édition : 1973. 2000.

[DK82] Devijver P.A., Kittler J., *Pattern Recognition –A Statistical Approach*, Englewood Cliffs, NJ : Prentice-Hall International. 1982.

[DKS95] Dougherty J., Kohavi R., Sahami M., « Supervised and unsupervised discretization of continuous attributes », in *Machine Learning : Proceedings of the* 12^{th} *International Conference (ICML-95)*, pp. 194–202, Morgan Kaufmann. 1995.

[dLM00] de La Mettrie J.O., *L'homme-machine*, Éditions des Mille et une nuits, édition originale : 1747. 2000.

[Dud76] Dudani S.A., « The distance-weighted k-nearest-neighbor rule », *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(4). 1976.

[Dup99] Dupuy J.P., *Aux origines des sciences cognitives*, Paris : Éditions de la Découverte, 2nde édition, 1ère édition : 1994. 1999.

[EMTB00] Esposito F., Malerba D., Tamma V., Bock H.H., « Similarity and dissimilarity measures : classical resemblance measures », in Bock H.H., Diday E., editors, *Analysis of Symbolic data*, pp. 139–152, Springer-Verlag. 2000.

[FH51] Fix E., Hodges J.L., « Discriminatory analysis—nonparametric discrimination : Consistency properties », Technical Report 21-49-004, report no. 04, USAF School of Aviation Medicine, Randolph Field, Texas. 1951.

[FH52] Fix E., Hodges J.L., « Discriminatory analysis—nonparametric discrimination : Small sample performance », Technical Report 21-49-004, report no. 11, USAF School of Aviation Medicine, Randolph Field, Texas. 1952.

[FI93] Fayyad U.M., Irani K.B., « Multi-interval discretization of continuous-valued attributes for classification learning », in *Proceedings of the* 13^{th} *International Joint Conference on Artificial Intelligence*, pp. 1022–1027, San Mateo, CA : Morgan Kaufmann. 1993.

[FPSS96] Fayyad U.M., Piatetsky-Shapiro G., Smyth P., « From Data Mining to Knowledge Discovery : An Overview », in Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R., editors, *Advances in Knowledge Discovery and Data Mining*, pp. 1–34, The AAAI Press. 1996.

[FW99] Frank E., Witten I.H., « Making better use of global discretization », in *Proceedings of the* 16^{th} *International Conference on Machine Learning*, pp. 115–123, San Francisco, CA : Morgan Kaufmann. 1999.

[Gan90] Ganascia J.G., *L'âme-machine, les enjeux de l'intelligence artificielle*, Paris : Éditions du Seuil, Collection Science ouverte. 1990.

[Gar93] Gardner H., *Histoire de la révolution cognitive, la nouvelle science de l'esprit*, Éditions Payot, trad. fr. par J.L. Peytavin de The Mind's New Science, A History of the Cognitive Revolution : 1985. 1993.

[Gea54] Geary R.C., « The contiguity ratio and statistical mapping », *The Incorporated Statistician*, 5 :115–145. 1954.

[GS69] Gabriel K.R., Sokal R.R., « A new statistical approach to geographic variation analysis », *Systematic Zoology*, 18 :259–278. 1969.

[JD88] Jain A.K., Dubes R.C., *Algorithms for clustering data*, Prentice Hall. 1988.

[JDM00] Jain A.K., Duin R.P.W., Mao J., « Statistical Pattern Recognition : A Review », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1) :4–37. 2000.

[JKP94] John G.H., Kohavi R., Pfleger K., « Irrelevant Features and the Subset Selection Problem », in *International Conference on Machine Learning*, pp. 121–129. 1994.

[Joh95] JOHN G.H., « Robust decision trees : removing outliers from data », in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pp. 174–179, Montréal, Québec : AAI Press. 1995.

[JT92] JAROMCZYK J.W., TOUSSAINT G.T., « Relative Neighborhood Graphs And Their Relatives », *P-IEEE*, 80 :1502–1517. 1992.

[Kay92] KAYSER D., « Profondeur variable et Sciences Cognitives », in ANDLER D., editor, *Introduction aux sciences cognitives*, pp. 195–218, Paris : Gallimard, Collection Folio/Essais. 1992.

[KI85] KITTLER J., ILLINGWORTH J., « Relaxation labelling algorithms— a review », *Image and Vision Computing*, 3(4) :206–216. 1985.

[KK95] KOSSLYN S.M., KOENIG O., *Wet Mind, the new cognitive neuroscience*, New York : The Free Press, 2nde édition, 1ère édition : 1992. 1995.

[Kod99] KODRATOFF Y., « L'extraction de connaissance à partir de données : un nouveau sujet pour la recherche scientifique », in SEBBAN M., VENTURINI G., editors, *Apprentissage automatique*, pp. 9–39, Paris : Hermes Science Publication. 1999.

[Koh97] KOHONEN T., *Self-organising Maps*, Germany : Springer-Verlag, 2nde édition, 1ère édition : 1995. 1997.

[Kru56] KRUSKAL J.B., « On the shortest spanning subtree of a graph and the travelling salesman problem », *Proceedings of the American Mathematical Society*, 7 :48–50. 1956.

[Lar91] LARGERON C., *Reconnaissance des formes par relaxation : un modèle d'aide à la décision*, Thèse de doctorat, Université Lyon 1. 1991.

[Leo66] LEONTIEF W., *Input-output Economics*, New York : Oxford University Press. 1966.

[Ler70] LERMAN I.C., *Les bases de la classification automatique*, Paris : Gauthier-Villars. 1970.

[LMJ03] LALLICH S., MUHLENBACH F., JOLION J.M., « A statistical test to control a region growing process within a hierarchical graph », *Pattern Recognition*, à paraître. 2003.

[LMZ02a] LALLICH S., MUHLENBACH F., ZIGHED D.A., « Improving classification by removing or relabeling mislabeled instances », in

Foundations of Intelligent Systems, Proceedings of the 13th International Symposium on Methodologies for Intelligent Systems (ISMIS 2002), Lyon, France, June 2002, LNAI 2366, pp. 5–15, Berlin Heidelberg : Springer-Verlag. June 2002.

[LMZ02b] LALLICH S., MUHLENBACH F., ZIGHED D.A., « Test de structure pour la prédiction de variable numérique », in *Actes du IX^{ème} Congrès de la Société Francophone de Classification – SFC'02*, pp. 235–238, Toulouse. Septembre 2002.

[LS95] LIU H., SETIONO R., « Chi2 : Feature selection and discretization of numeric attributes », in *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*. 1995.

[LW00] LUDL M.C., WIDMER G., « Relative Unsupervised Discretization for Association Rule Mining », in *Proceedings of 4th European Conference for Principles of Data Mining and Knowledge Discovery (PKDD'2000)*, pp. 148–158, Springer-Verlag. 2000.

[Mic83] MICHALSKI R.S., « Theory and Methodology of Inductive Learning », in MICHALSKI R.S., CARBONNEL J.G., MITCHELL T.M., editors, *Machine Learning : An Artificial Intelligence Approach*, volume 1, pp. 83–134, Morgan Kaufmann. 1983.

[Mit97] MITCHELL T.M., *Machine learning*, New York : McGraw-Hill. 1997.

[MLZ02] MUHLENBACH F., LALLICH S., ZIGHED D.A., « Amélioration d'une classification par filtrage des exemples mal étiquetés », *ECA*, 1(4) :155–166, conférence EGC 2002. 2002.

[Mor48] MORAN P.A.P., « The interpretation of statistical maps », in *Journal of the Royal Statistical Society*, serie B, pp. 246–251. 1948.

[MR02a] MUHLENBACH F., RAKOTOMALALA R., « Multivariate supervised discretization, a neighborhood graph approach », in KUMAR V., TSUMOTO S., ZHONG N., YU P.S., WU X., editors, *Proceeding of the 2002 IEEE International Conference on Data Mining – ICDM'02*, pp. 314–321, Maebashi City, Japan. December 2002.

[MR02b] MUHLENBACH F., RAKOTOMALALA R., « Utilisation d'amas pour la discrétisation de variables », in *Actes du IX^{ème} Congrès de la Société Francophone de Classification – SFC'02*, pp. 283–286, Toulouse. Septembre 2002.

[MZD01] Muhlenbach F., Zighed D.A., D'Hondt S., « Génération de règles par compression », *ECA*, 1(1–2) :93–104, conférence EGC 2001. 2001.

[NSB+89] Nierenberg D.W., Stukel T.A., Baron J.A., Dain B.J., Greenberg E.R., « Determinants of plasma levels of beta-carotene and retinol », *American Journal of Epidemiology*, 130 :511–521. 1989.

[Oli93] Oliver J.J., « Decision Graphs – An Extension of Decision Trees », in *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics*, pp. 343–350. 1993.

[Par62] Parzen E., « On estimation of a probability density function and mode », *Annals Mathematical Statistics*, 33 :1065–1076. 1962.

[Pri57] Prim R.C., « Shortest connection networks and some generalizations », *The Bell System Technical Journal*, 36 :1389–1401. 1957.

[PS88] Preparata F.P., Shamos M.I., *Computational Geometry, an introduction*, New York : Springer-Verlag, 2nde édition, 1ère édition : 1985. 1988.

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

[Qui92] Quinlan J.R., « Learning with Continuous Classes », in *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, pp. 343–348, Singapore : World Scientific. 1992.

[Qui93a] Quinlan J.R., « Combining Instance-Based and Model-based Learning », in Utgoff P.E., editor, *Proceedings of the 10th International Conference on Machine Learning, Amherst, Massachusetts*, pp. 236–243, San Mateo, CA : Morgan Kaufmann. 1993.

[Qui93b] Quinlan J.R., *C4.5 : Program for Machine Learning*, San Mateo, Ca : Morgan Kaufmann. 1993.

[Rao65] Rao C.R., *Linear statistical inference and its applications*, New-York : John Wiley & Sons. 1965.

[RHZ76] Rosenfeld A., Hummel R.A., Zucker S.W., « Scene labeling by relaxation operations », *IEEE Transactions on Systems Man and Cybernetics*, 6(6) :420–433. 1976.

[Rit90] Ritschard G., «Détection de données atypiques», in Brissaud M., Fortet M., Zighed D.A., editors, *La modélisation : confluent des sciences* , Paris : Éditions du CNRS. 1990.

[RL87] Rousseeuw P.J., Leroy A.M., *Robust regression and outlier detection*, New York : Wiley. 1987.

[RM86] Rumelhart D.E., McClelland J.L., editors, *Parallel Distributed Processing*, Cambridge, MA : MIT Press. 1986.

[RSR96] Rabaséda S., Sebban M., Rakotomalala R., « A comparison of some contextual discretization methods », *Information Sciences : Intelligent Systems*, 92(1-4) :137–157. 1996.

[Seb96] Sebban M., *Modèles théoriques en reconnaissance de formes et architecture hybride pour machine perceptive*, Thèse de doctorat, Université Claude Bernard – Lyon I. 1996.

[SS02] Schölkopf B., Smola A., *Learning with Kernels*, MIT Press. 2002.

[SW86] Stanfill C., Waltz D., « Toward Memory-based Reasoning », *Communications of the ACM*, 29 :1213–1228. 1986.

[SZ96] Sebban M., Zighed D.A., « Test de séparabilité des classes dans $\mathbb{R}^p$ », in *XXV^{ème} colloque des Structures Economiques, Econométrie et Informatique*, Lyon. Mai 1996.

[Tom76] Tomek I., « An experiment with the edited Nearest Neighbor Rule », *IEEE Transactions on Systems, Man and Cybernetics*, 6(6) :448–452. 1976.

[Tou80] Toussaint G.T., « The relative neighbourhood graph of a finite planar set », *Pattern Recognition*, 12 :261–268. 1980.

[Tur37] Turing A.M., « On computable numbers with an application to the entscheidungs problem », *Proceedings of the London Mathematical Society*, 42 :230–265. 1937.

[Tur50] Turing A.M., « Computing machinery and intelligence », *Mind*, 59 :433–460. 1950.

[UG99] Uysal I., Guvenir H.A., « An overview of regression techniques for knowledge discovery », *Knowledge Engineering Review*, 14 :319–340. 1999.

[Vap98] Vapnik V., *Statistical Learning Theory*, NY : John Wiley. 1998.

[WAD94] Wess S., Althoff K.D., Derwand G., « Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning », in *Topics in Case-Based Reasoning*, LNAI, pp. 167–181, Springer-Verlag, selected paper from from EWCBR-93. 1994.

[Wil72] Wilson D.R., « Asymptotic properties of nearest neighbors rules using edited data », *IEEE Transactions on systems, Man and Cybernetics*, 2 :408–421. 1972.

[WM97] Wilson D.R., Martinez T.R., « Improved Heterogeneous Distance Functions », *Journal of Artificial Intelligence Research*, 6 :1–34. 1997.

[WM00] Wilson D.R., Martinez T.R., « Reduction techniques for exemplar-based learning algorithms », *Machine Learning*, 38 :257–268. 2000.

[ZLM01] Zighed D.A., Lallich S., Muhlenbach F., « Séparabilité des classes dans $\mathbb{R}^p$ », in *Actes du VIII^{ème} Congrès de la Société Francophone de Classification – SFC'01*, pp. 356–363, Pointe-à-Pitre, Guadeloupe, France. Décembre 2001.

[ZLM02] Zighed D.A., Lallich S., Muhlenbach F., « Separability Index in Supervised Learning », in Elomaa T., Mannila H., Toivonen H., editors, *Principles of Data Mining and Knowledge Discovery, Proceedings of the 6th European Conference PKDD 2002, Helsinki, Finland*, LNAI 2431, pp. 475–487, Berlin Heidelberg : Springer-Verlag. August 2002.

[ZR00] Zighed D.A., Rakotomalala R., *Graphes d'induction. Apprentissage et Data Mining*, Paris : Hermes Science Publication. 2000.

[ZRR98] Zighed D.A., Rabaséda S., Rakotomalala R., « Fusinter : a method for discretization of continuous attributes for supervised learning », *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(33) :307–326. 1998.

[ZTAL90] Zighed D.A., Tounissoux D., Auray J.P., Largeron C., « Discrimination basée sur un critère d'homogénéité locale », *Traitement du signal*, 2 :213–220. 1990.