USTL

U.F.R. d'I.E.E.A.

Numéro d'ordre : 2485

Méthodes d'induction par arbres de décision dans le cadre de l'aide au diagnostic

THÈSE

présentée et soutenue publiquement le 1er février 1999

pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille

(Spécialité Automatique et Informatique Industrielle)

par

Paul-Benoît PERCHE

Maître EEA

Composition du jury

Président :	José RAGOT	Institut National Polytechnique de Lorraine
Rapporteurs :	Thierry DENŒUX	Université de Technologie de Compiègne
	Patrick MILLOT	Université de Valenciennes et du Hainaut-Cambrésis
Co-directeurs :	Denis POMORSKI	Université de Lille I
	Marcel STAROSWIECKI	Université de Lille I
Examinateurs :	Rémi GILLERON	Université de Lille III
	Engelbert MEPHU N'GUIFO	Université d'Artois

Laboratoire d'Automatique et d'Informatique Industrielle de Lille — UPRES A 80 21

CNRS CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Remerciements

Le travail présenté dans ce mémoire a été effectué au sein du Laboratoire d'Automatique et d'Informatique Industrielle de Lille (LAIL - UPRES A 8021) dans l'équipe Analyse et Surveillance des Processus Industriels Complexes (ASPIC) sous la direction de Messieurs Marcel STAROSWIECKI *et Denis* POMORSKI*, respectivement Professeur et Maître de Conférences à l'Université des Sciences et Technologies de Lille. Je tiens à leur témoigner ma profonde gratitude pour l'accueil, le suivi et l'aide précieuse qu'ils m'ont apportés tout au long de ce travail.*

Je suis très reconnaissant à Monsieur Patrick MILLOT*, Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis et Monsieur Thierry* DENŒUX*, Maître de Conférences, habilité à diriger des recherches à l'Université de Technologie de Compiègne, pour l'honneur qu'ils me font en acceptant d'examiner ce travail et d'être les rapporteurs de cette thèse.*

J'adresse également mes remerciements à Monsieur José RAGOT*, Professeur à l'Institut National Polytechnique de Lorraine pour l'honneur qu'il me fait de présider le jury, ainsi qu'à Monsieur Rémi* GILLERON*, professeur à l'Université de Lille III et Monsieur Engelbert* MEPHU N'GUIFO*, Maître de Conférences à l'Université d'Artois, pour l'honneur qu'ils me font en examinant ce travail et en acceptant de participer au jury.*

Je tiens particulièrement à remercier tous les membres de l'équipe ASPIC, et tous les thésards pour leur sympathie et l'ambiance chaleureuse qu'ils ont su entretenir tout au long de mon séjour parmi eux.

Enfin, mes remerciements vont à tous ceux qui m'ont soutenu ou qui, d'une manière ou d'une autre ont contribué à l'élaboration de ce travail.

À ma famille,
À mes amis.

Table des matières

Introduction générale

Surveillance, Diagnostic et Apprentissage

« Effectuer le diagnostic d'un système c'est identifier son mode de fonctionnement et la cause de ce mode » (B. DUBUISSON). Nous nous proposons dans ce travail de définir les termes « mode de fonctionnement » et « diagnostic » couramment utilisés. Nous mettrons dès lors en évidence les liens unissant ces deux termes : à l'aide d'une phase d'*Observation*, il nous sera possible d'effectuer un *Apprentissage* de ces « modes de fonctionnement » dans le but de procéder à la « surveillance » et au « diagnostic » d'un « système ».

Le pari consiste donc à définir les différents termes utilisés, et à montrer les liens qui les unissent. Dans ce cadre, et afin d'employer des termes usuels, toutes les définitions proposées ici sont extraites (sauf précision) du dictionnaire *Hachette* 1994...

Nous définirons d'abord la notion de *système* (ou processus) au sens général du terme. Puis, nous définirons le terme *apprentissage*. Dans le cadre de la *surveillance*, nous déterminerons les quatre propriétés qui la caractérisent, à savoir la perception, la détection, le diagnostic, et le pronostic. Ces définitions étant posées, nous établirons un lien entre l'apprentissage et la surveillance d'un processus complexe.

Nous terminerons alors par le plan de notre travail.

Système...

Définissons d'abord la notion de système (ou processus) au sens général du terme :

« Un système se définit tout d'abord comme une entité relativement individualisable, qui se détache de son contexte ou de son milieu tout en procédant à des échanges avec son environnement » ([Wal77], p.13). À ce système est souvent associé un ensemble de capteurs plus ou moins complexes et permettant d'appréhender ces échanges ainsi qu'un ensemble d'actionneurs permettant d'agir sur le système (figure 1).

À titre d'exemple, dans le cadre médical, le système représente le patient, associé à un ensemble de capteurs (monitoring, ...) plus ou moins complexes, et à un ensemble d'actionneurs agissant sur le patient (respirateur par exemple).

Dans un cadre plus industriel, nous pouvons considérer l'exemple d'une chaîne de production, et dans ce cas, le système comprend la chaîne elle-même, les différents capteurs de position, de vitesse, de température, de débit, etc..., ainsi que les différents actionneurs qui lui sont associés, tels que vérins, vannes, moteurs, éléments chauffants, etc...

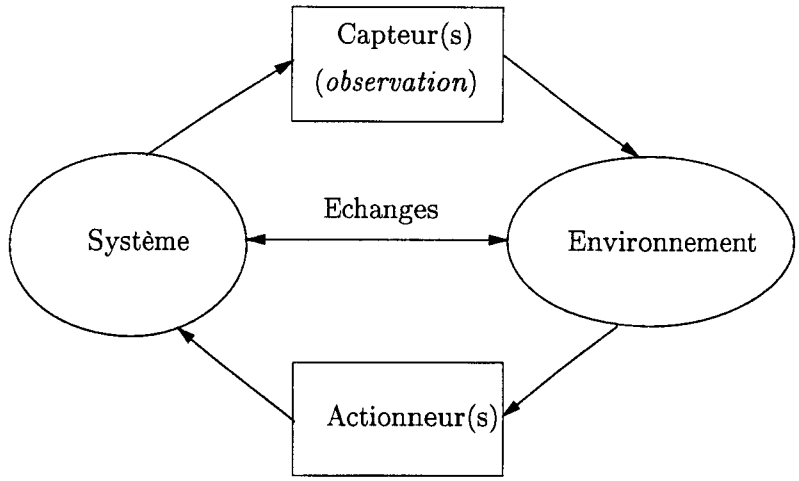

FIG. 1: *Système et environnement*

On peut également envisager de ne considérer qu'une partie de cette chaîne et dans ce cas, le système peut par exemple être constitué uniquement d'un moteur et de ses propres capteurs de vitesse, température, courant, tension, etc... Cet exemple sera par ailleurs traité dans le chapitre 5.

Dans le but de contrôler ou de réguler de tels systèmes, les notions de correction et de consigne sont introduites.

Toujours sur le même exemple médical, suivant les signaux délivrés par les capteurs, on se mettra dans tel ou tel mode de l'insufflateur ou du respirateur dans le but de corriger l'état de souffrance du patient. Un « état de souffrance » admissible du patient sera alors la consigne, et le patient pourra alors être, entre autres, en ventilation contrôlée ou assistée.

Si l'on reprend l'exemple de la chaîne de production, la consigne pourra représenter la cadence de production, ou une certaine qualité du produit fini, etc... En ce qui concerne le moteur, il peut s'agir par exemple d'une vitesse à atteindre. Les commandes des actionneurs ou du moteur sont ainsi corrigées et adaptées de manière à atteindre cette consigne.

Nous pouvons dès lors établir le schéma classique de l'automaticien (figure 2).

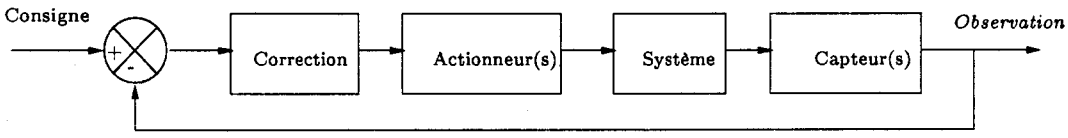

FIG. 2: *Schéma classique de la régulation*

À cette notion, il faut ajouter une propriété très importante d'évolution. En effet, le système peut d'une part être statique ou dynamique, c'est-à-dire que les relations qui lient ses caractéristiques peuvent être liées par une contrainte d'ordre sur le temps (système

dynamique) ou au contraire les observations du système sont indépendantes du temps (système statique). D'autre part, le système peut être ou ne pas être stationnaire. Un système n'est pas stationnaire si les relations liant les variables le caractérisant sont susceptibles d'évoluer dans le temps (même s'il est statique).

Dans ce cadre, il est nécessaire d'avoir un modèle de comportement du système étudié. Ce modèle peut être obtenu :

- par une phase de modélisation (connaissance des relations en général mathématiques entre les diverses grandeurs mises en jeu) ;
- par une phase d'identification paramétrique (détermination des paramètres d'une famille de modèles mathématiques) ;
- ou enfin par une phase d'apprentissage (on ne considère pas de modèle mathématique et on essaie de construire un modèle à base de règles).

Nous nous intéressons dans le cadre de ce travail à une phase d'apprentissage en considérant les systèmes pour lesquels aucun modèle mathématique de comportement *a priori* n'est connu. Nous n'excluons pas pour autant les systèmes pour lesquels un modèle mathématique de comportement existe ; dans ce cas, le modèle construit par apprentissage pourra être comparé et utilisé en complément du précédent. Dans le dernier chapitre, nous traiterons l'exemple d'un moteur asynchrone pour lequel les résidus (donc le modèle mathématique) seront pris en compte.

Apprentissage...

Le terme « *Apprentissage* » est utilisé dans les sciences dites « **cognitives** ». Ces dernières sont définies comme un « ensemble de sciences qui étudient l'intelligence (humaine, animale, artificielle) en tant qu'instrument de la cognition : psychologie, linguistique, informatique, etc... » ; la cognition étant définie comme la « Faculté de connaître - Acte intellectuel par lequel on acquiert une **connaissance** ».

Le terme « Connaissance » désigne, quant à lui, une « *Idée exacte d'une réalité, de sa situation, de son sens, de ses caractères, de son fonctionnement. Avoir une grande connaissance de la musique, des affaires* ».

Toujours sur le même exemple médical, le médecin possède la connaissance de différentes pathologies, connaissance qu'il a accumulée tout au long de son passé et de son apprentissage de la médecine. Ce médecin possède un « modèle » des différentes pathologies connues. C'est en possédant cette connaissance *a priori* que le médecin pourra effectuer un diagnostic correct.

Si l'on considère l'exemple de la chaîne de production, il est nécessaire de connaître les réactions et le comportement du système dans différentes situations afin de le commander ou de déterminer l'état dans lequel il se trouve.

La connaissance correspond dans ce cas à un résumé d'un ensemble d'informations, et nécessite un certain recul vis-à-vis de cet ensemble. Cette connaissance est appelée par les automaticiens un « **modèle** ».

Un point important concerne le temps mis par le système cognitif pour apprendre. Sauf cas particulier, la rapidité d'apprentissage n'est pas nécessairement le critère de qualité le plus prépondérant. En effet, l'apprentissage s'effectue généralement hors-ligne et donc en temps différé. L'essentiel étant de bien apprendre, cette période d'apprentissage peut devenir relativement longue dans la mesure où elle reste raisonnable.

Diagnostic et Surveillance...

« *Diagnostic* », « *Surveillance* », « *Supervision* » sont autant de termes utilisés dans la littérature et principalement par les **communautés scientifiques automaticienne et médicale**. Ces termes s'appliquent en général à un **système** évoluant dans le **temps** et susceptible de passer par de multiples **modes de fonctionnement** plus ou moins **dégradés**. Ayant une certaine **connaissance** du système **acquise par le passé**, il faut pouvoir l'**observer** et le **surveiller** dans l'optique de le **contrôler**.

Définissons tout d'abord le terme **Surveillance** comme « *l'action de surveiller* » ; et **Surveiller**, c'est « *Contrôler, suivre le déroulement de...* ».

Appliquée à un système, la surveillance implique les notions de contrôle et d'observation de celui-ci.

La surveillance d'un processus s'inscrit dans une démarche de conduite de celui-ci. Elle vise à détecter, localiser et à diagnostiquer les défaillances (ou pannes ou encore pathologies dans un cadre médical) diminuant ses performances, *via* les informations délivrées par un ensemble de capteurs (figure 3).

La surveillance d'un processus consiste alors à comparer l'état de fonctionnement de celui-ci à l'instant présent (connaissance à court terme, en opposition à la phase d'apprentissage) et la connaissance de son fonctionnement normal et de ses dysfonctionnements connus.

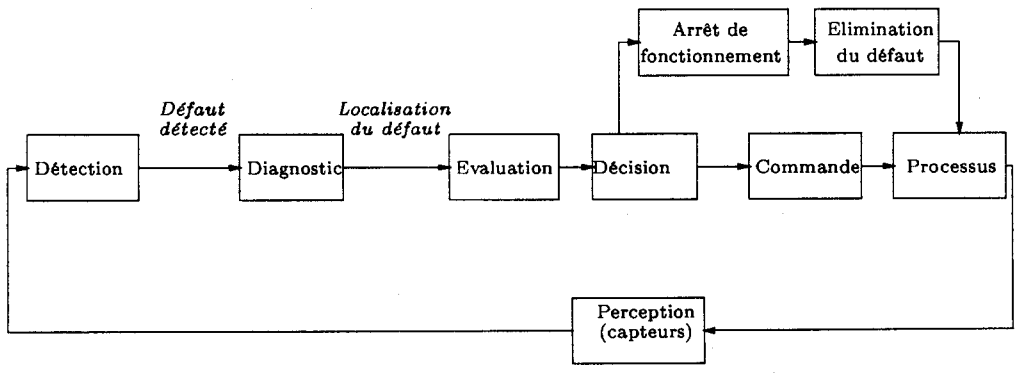

FIG. 3: *Schéma général de la surveillance d'un processus [Ise84]*

Quatre fonctions caractérisent la surveillance d'un processus [BJL+90] :

I. La perception...

Une indication (en général visuelle, olfactive ou sonore) et/ou un ensemble de capteurs fournissent un certain nombre d'informations sur le processus en question. La détermination du capteur et son placement sont des points importants qu'il ne faut surtout pas négliger.

De plus, ces informations sont en général pré-traitées (ou filtrées) afin d'en extraire un signal non (ou peu) bruité, et/ou des caractéristiques principales du signal (la moyenne, la période, le minimum, le maximum, ... seront par exemple extraits du signal initial).

C'est à partir de ces informations brutes et/ou pré-traitées que vont pouvoir se réaliser correctement les étapes suivantes, à savoir la détection, le diagnostic, et le pronostic.

II. La détection...

La détection consiste à déterminer le mode de fonctionnement dans lequel se trouve le système à partir des informations fournies par la phase de perception. À ce niveau, le système se trouve en fonctionnement normal ou anormal. Il nous faut décider entre deux hypothèses H_0 (fonctionnement normal) et H_1 (fonctionnement anormal du système). Si plusieurs modes de dysfonctionnement peuvent apparaître, le problème de détection consiste à discriminer le mode de fonctionnement normal contre tous les autres modes de dysfonctionnement.

III. Le diagnostic...

Le **Diagnostic** est défini comme un « *acte par lequel le médecin, en groupant les symptômes et les données de l'examen clinique et des divers autres examens, les rattache à une maladie bien identifiée* ». En retournant la phrase, le diagnostic est dès lors défini de la façon suivante : ayant la connaissance *a priori* d'un ensemble de maladies bien identifiées, le médecin regroupe les données qu'il a en sa possession afin de choisir une de ces maladies.

Un bon diagnostic repose donc sur une bonne phase d'apprentissage des différentes maladies ainsi que sur une bonne acquisition de données, afin de choisir parmi un panel de maladies possibles. Si l'apprentissage et/ou l'acquisition des données est faussé(s), alors le diagnostic le sera également.

Il est important de remarquer que dans cette définition, le médecin associe automatiquement l'ensemble des observations à une maladie connue. La notion de maladie inconnue n'intervient pas dans cette définition. Il ne peut donc pas y avoir une évolution de la « base de données » maladies, ni du modèle global regroupant toutes les descriptions de toutes les maladies. Afin de pallier ce problème d'évolution, des méthodes avec rejet ont été développées par [Dub90], qui consistent à ne pas choisir une conclusion (dans notre exemple, une maladie) plutôt que d'en choisir une fausse. La méthode consiste alors à attendre quelques observations supplémentaires afin de décider ou afin de faire évoluer la base de données des pathologies en question.

Dans ce sens, et d'un point de vue plus industriel, l'expert du système devra de la même façon diagnostiquer le mode de fonctionnement de son système, qui pourra évoluer suivant plusieurs modes, plus ou moins dégradés. Nous distinguons ainsi les modes de fonctionnement suivants :

- Les modes de fonctionnement normaux du système. Aucune anomalie n'est observée et le système est dans le mode de fonctionnement nominal si les produits ou services fournis correspondent tout à fait à ce que les experts du système attendent de celui-ci.
- Les modes de dysfonctionnement, ou de comportements anormaux. Le cahier des charges du système n'est que partiellement rempli. Suivant l'importance accordée à telle ou telle partie du cahier des charges, on peut distinguer plusieurs modes de fonctionnement anormaux :
  - Les modes défaillants : une panne (ou un ensemble de pannes) est intervenue (par exemple, un capteur défectueux, une panne liée au processus...) pendant le fonctionnement du système, ce qui entraîne une dégradation des performances, voire un arrêt total de celui-ci.
  - Les modes dégradés : le système ne répond pas au cahier des charges annoncé, sans pour autant entrer dans un mode dit « critique ». C'est en particulier le cas des systèmes décomposables : « Un système est dit *quasi-décomposable* s'il peut être décomposé en sous-systèmes quasi-isolés S_i reliés entre eux et avec l'environnement. Le vocabulaire utilisé pour désigner le système et ses sous-systèmes individualisables est très riche et plus ou moins spécialisé : collection, groupe, classe, agrégat, constellation, équipe pour l'ensemble ; objets, membres, unités, individus pour les éléments » ([Wal77], p.39).

    En effet, une partie du système pourra être dans un état de fonctionnement anormal sans pour autant entraîner un dysfonctionnement d'une autre partie du processus en question, ou encore un arrêt total de celui-ci.
  - Les modes critiques : le système ne répond quasiment plus au cahier des charges annoncé à l'origine.
  - Les modes interdits : le système se trouve dans une configuration dangereuse et un arrêt immédiat de celui-ci est dans ce cas souhaitable. Ce mode doit donc être détecté en « temps réel ».

C'est à ce niveau qu'il faut distinguer la panne d'un capteur (plus ou moins amorcée dans le passé, on parle alors de dérive lente de celui-ci, en opposition aux pannes brusques) n'entrant pas dans le processus de régulation du système (le capteur n'a aucune incidence sur la conduite du processus en question), de la panne « processus » dans laquelle tout ou partie du système ne répond plus au cahier des charges. On entre dans le deuxième cas dans un mode dit « dégradé ». Dans de nombreux cas de figure, il n'est pas évident de distinguer les pannes capteur des pannes liées au

mauvais fonctionnement du processus en question. Un exemple sera traité dans le chapitre 5.

Bien entendu, dans le cadre médical, les modes de dysfonctionnement correspondent aux maladies, et aux différentes pathologies anormales.

Il est bien évident que le système pourra évoluer d'un mode de fonctionnement à l'autre, et que le mode souhaitable, autant que faire se peut, est le mode de fonctionnement normal. Malheureusement, pour diverses raisons, et suivant les systèmes étudiés, des modes de dysfonctionnement sont inévitables.

C'est à ce niveau que l'expert du système en question (ou le médecin) devra disposer d'un bon modèle de fonctionnement normal de son système ainsi que d'un excellent modèle des différentes pannes (ou maladies) possibles, ou tout au moins des différentes pannes déjà observées. On voit ici tout l'intérêt d'une bonne phase d'apprentissage, qui garantira un bon diagnostic.

De plus, ce diagnostic devra s'effectuer, dans la majorité des cas, en « temps réel », sous peine de ne plus répondre au cahier des charges fixé (c'est d'autant plus vrai dans le cadre de la surveillance d'un patient sous contrôle respiratoire, ou en réanimation post-opératoire). L'exemple typique est l'analyse d'un prélèvement en laboratoire, qui nécessite parfois plusieurs jours et donc qui entraîne une perte de temps dans la boucle de rétroaction (contrôle) du système.

Associé à ce problème de « temps réel », un problème encore plus complexe est celui de la localisation du (ou des) défaut(s) conduisant à un mode de fonctionnement anormal. Afin de pallier ce problème, la prise en compte de l'évolution de l'état du système est très importante. On parle dès lors de pronostic.

IV. Le pronostic...

Il est très intéressant de pouvoir prédire l'état de fonctionnement futur d'un système afin de pouvoir pallier, autant que possible, à divers dysfonctionnements de celui-ci.

À titre d'exemple, on pourra pallier plus facilement (nous devrions plutôt dire « moins difficilement ») à un problème de santé grave si l'individu (on ne parle pas ici de patient) surveille régulièrement son état de santé. À partir du moment où on détecte certaines pathologies, on peut alors prévoir son état de santé futur, et dans ce cas un pronostic de son état de santé peut être effectué. On sait très bien qu'en règle générale, plus vite est dépisté le problème de santé, plus les chances de guérison sont importantes.

Pour reprendre le cas de la chaîne de production, la surveillance du système permet par exemple de détecter la dérive d'un capteur, ce qui peut entraîner à plus ou moins long terme, une défaillance du système. Celle-ci est donc prévisible donc pronostiquée à l'avance et peut ainsi être évitée. De même, en ce qui concerne le moteur, on peut envisager de détecter une légère défaillance ou un échauffement anormal, qui pourrait aboutir à des conséquences plus graves voire catastrophiques, si ce problème persistait. Le pronostic de cette issue permet donc d'agir de façon à l'éviter.

Surveillance et Apprentissage font-ils bon ménage ?

Dans la littérature, les termes *surveillance* et *apprentissage* sont rarement associés, et on parle dès lors de **Reconnaissance des Formes** appliquée au Diagnostic [Dub90]. L'apprentissage est vu, dans ce cas et à juste titre, comme une étape préliminaire à la surveillance d'un processus complexe.

Pourtant, nous sommes intimement persuadés que sans une bonne phase préliminaire d'apprentissage d'un modèle (qu'il soit mathématique ou non), une bonne surveillance d'un système complexe serait une gageure.

Le terme commun aux deux domaines est le terme « **connaissance** », et nous distinguons deux types de connaissances : les connaissances acquises par le passé, et les connaissances que l'on vient d'acquérir sur le système dans le but de le surveiller.

À titre d'exemple, sans aucune connaissance *a priori* du patient et de son environnement immédiat (les capteurs qui lui sont associés), le médecin ne pourra agir sur celui-ci afin de faire évoluer son état. Par contre, s'il sait comment va réagir son patient (le médecin possède un modèle du patient), il peut, en ayant observé suffisamment longtemps celui-ci, lui prodiguer une thérapeutique dans le but de l'amener dans un état de santé dit satisfaisant.

De même, à partir d'observations effectuées sur la chaîne de production, on ne pourra la contrôler ou la surveiller que si au préalable on a acquis une bonne connaissance du comportement de celle-ci.

C'est en confrontant ces deux types de connaissances que l'on pourra surveiller un système plus ou moins complexe.

Cette introduction nous a permis de montrer qu'il n'était pas question, en règle générale, d'effectuer une bonne surveillance d'un processus sans une bonne phase préliminaire d'apprentissage des états de fonctionnement de celui-ci. Le schéma ci-dessous (figure 4) paraît dès lors évident.

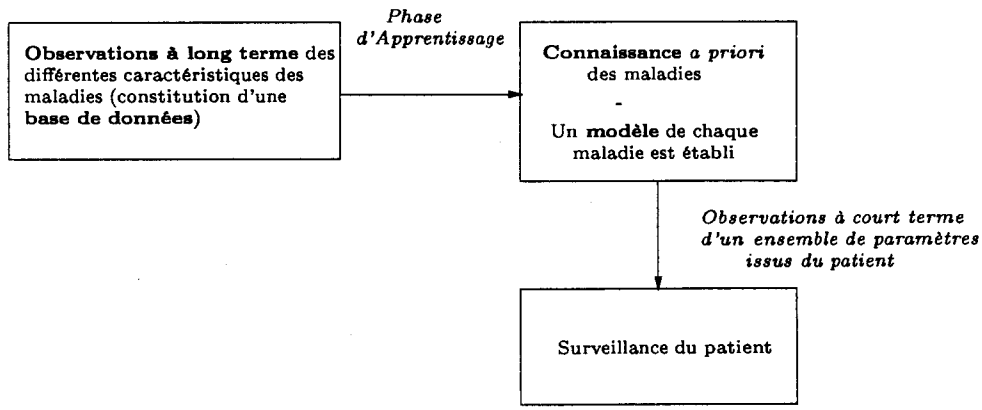

FIG. 4: *Surveillance et apprentissage*

L'un des principaux problèmes est que les modes de dysfonctionnement ne sont en général que peu rencontrés pendant la phase d'apprentissage du modèle. En effet, l'expert ne mettra pas en dysfonctionnement volontairement son système, sous peine de ne plus répondre au cahier des charges fixé (rentabilité du système, ...).

Il faut par conséquent que le système d'apprentissage puisse prendre en compte les différents modes de fonctionnement du système et ainsi faire évoluer le modèle obtenu afin que le système de surveillance puisse :

- détecter et reconnaître les dysfonctionnements déjà observés, puis localiser les différents éléments en faute (les pannes). Ces pannes peuvent être des pannes capteur(s) ou des pannes processus (sur une partie plus ou moins importante du processus);
- les prédire (on parle alors de *pronostic* ou de *diagnostic prédictif*), et éventuellement pallier ces dysfonctionnements (si possible);
- évoluer afin de trouver d'autres modes de dysfonctionnement jusqu'alors inconnus.

Il est bien évident que ces trois problèmes ne sont pas simples à résoudre et demandent des outils élaborés.

Dans le cadre de ce travail, nous présenterons dans un premier temps la démarche générale de l'étude d'un système sans modèle mathématique de comportement.

Nous définirons dans un premier chapitre quelques notions relatives aux cogniticiens et aux automaticiens et nous présenterons les données d'apprentissage sur lesquelles se basera l'analyse du système. Nous introduirons également les notions de finesse de variable ainsi que les notions de données incohérentes.

Dans un deuxième chapitre, nous présenterons les méthodes issues de l'apprentissage dans le cadre d'une analyse structurale d'un système physique plus ou moins complexe. Les problèmes de l'apprentissage et de l'analyse structurale seront présentés. Les notions de « modèle », de « connaissance » et de « prédiction » seront ainsi définies afin d'introduire les différentes démarches utilisées en apprentissage pour résoudre ces problèmes. Dans ce contexte, les principaux algorithmes utilisés dans la littérature seront présentés, et une méthodologie générale de validation de modèle permettant de mesurer la « qualité » du modèle sera présentée. Cette démarche nous permettra par exemple de ne conserver que le meilleur modèle parmi plusieurs.

Dans un deuxième temps, le cadre de travail ayant été présenté, une méthodologie d'étude d'un système physique par construction d'arbres de décision sera proposée.

Les outils de la théorie de l'information appliquée à l'analyse structurale des systèmes seront présentés dans un troisième chapitre, ce qui nous permettra d'établir un critère d'étude utilisable dans les algorithmes développés.

Dans un quatrième chapitre, nous présenterons préalablement *C4.5*, l'algorithme de construction d'arbres de décision le plus connu à l'heure actuelle, ce qui nous permettra

de développer des algorithmes permettant la construction d'un modèle de comportement par l'intermédiaire d'arbres de décision. Ces méthodes nous permettront de construire un arbre de décision, *via* une approche *ascendante* ou *descendante* associée à un algorithme *agrégatif* ou *désagrégatif* de sélection des variables pertinentes pour l'étude de notre système complexe. Les propriétés de ces algorithmes seront comparées et discutées de façon formelle (une comparaison empirique sera effectuée au chapitre 5).

Enfin, dans un dernier chapitre, les méthodes développées seront validées sur des cas concrets. Nous mettrons en évidence l'intérêt de l'utilisation des méthodes basées sur une approche ascendante désagrégative en traitant les tableaux issus de la littérature, et nous comparerons *C4.5* et l'une des meilleures approches proposées sur des données issues d'un moteur asynchrone représentant un exemple plus industriel.

Première partie

Démarche générale de l'étude d'un système sans modèle mathématique de comportement

(Position de l'apprentissage dans une démarche automaticienne)

Introduction de la première partie

L'objectif de cette partie est de présenter la démarche générale de l'étude d'un système sans modèle mathématique de comportement. En effet, nous proposons de construire le modèle d'un système afin de le surveiller dans le cadre de l'aide au diagnostic. Pour cela, nous allons présenter les différentes démarches de modélisation d'un tel système par apprentissage. Ainsi, le vocabulaire employé par les deux communautés « Apprentissage » et « Automatique » sera présenté et les différentes techniques d'apprentissage seront exposées.

Dans un souci de clarification nécessaire à la formulation d'une méthodologie générale, GEORGE J. KLIR a proposé et développé une **hiérarchie épistémologique des systèmes** [Kli75, Kli76, Kli77] dont l'idée de base est la suivante: « Plus on possède d'information sur le système considéré, et plus son niveau (hiérarchique) est élevé ». Ainsi, un système pour lequel on connaît un certain nombre de relations liant ses variables caractéristiques aura un niveau supérieur au système dont on n'a effectué qu'une simple observation.

Dans ce cadre, la figure 1 représente la démarche générale du traitement de l'information en vue de la modélisation d'un système.

Pour G.J. KLIR, un système au niveau épistémologique le plus bas, est appelé **Système-Source**. Il constitue le **Niveau 0** et correspond à une phase préparatoire. À ce niveau, sont définis l'ensemble des variables du système ainsi que leurs ensembles de modalités (ou états) respectifs. Ces variables peuvent être des variables d'entrée, de sortie du système, des variables d'état...

Le choix de ces variables dépend avant tout de l'expérience de l'expert et des connaissances qu'il possède déjà sur le système.

Le phénomène physique pourra ainsi être appréhendé *via* un ensemble de capteurs définis dans la phase d'**instrumentation** de la chaîne. Cette phase est bien entendu essentielle dans la chaîne de traitement de l'information: un mauvais choix de capteur(s) conduira alors à une impossibilité d'étudier correctement le système.

Au **Niveau 1** appelé **Système-Données**, correspond une série d'observations effectuées sur le système. Celui-ci est alors connu par une succession de modalités ou états pris par les variables, qui forment le tableau initial des données que nous nous proposons d'analyser.

Le signal issu des capteurs pendant la phase d'observation pourra être traité, afin d'obtenir un signal sans bruit et/ou afin d'obtenir des données représentant les caractéristiques principales du signal d'origine (par exemple, la période, le minimum, le maximum,

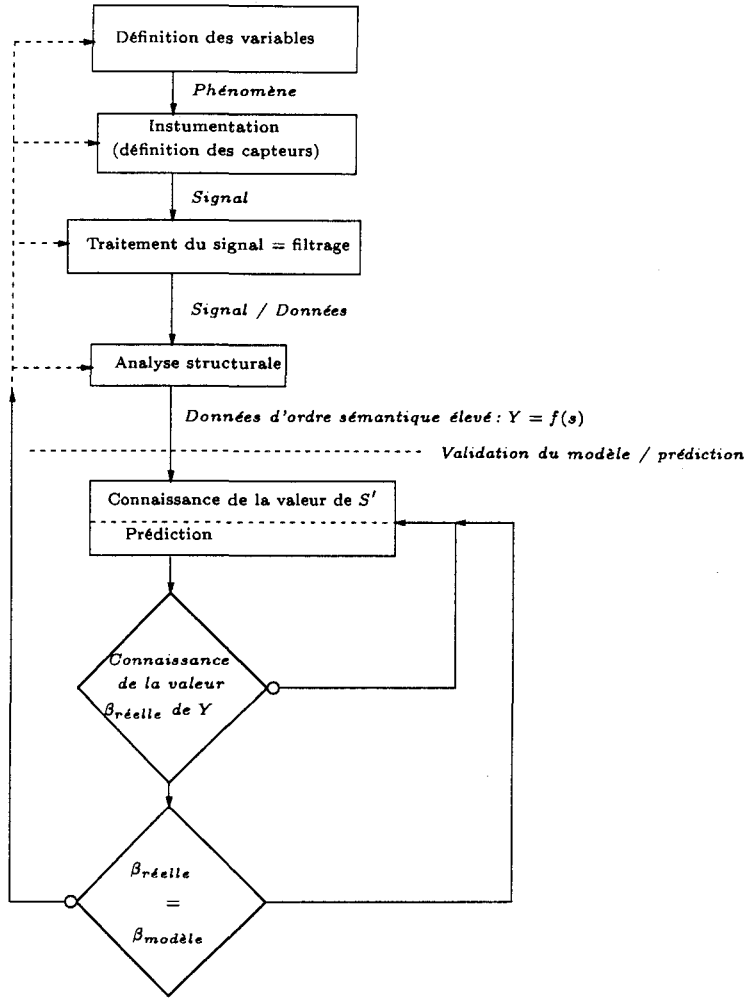

FIG. 1: *Chaîne de traitement de l'information en vue de la modélisation d'un système*

la moyenne d'un signal, ...).

D'après G.J. KLIR, le **Niveau 2** ou encore **Système-Générateur** caractérise des systèmes sur lesquels on a mis en évidence un ensemble de relations invariantes ou fonctionnelles permettant la détermination du comportement de quelques variables en fonction de celui des autres.

C'est ainsi qu'à partir des données (et/ou du signal) extraites, un modèle paramétrique ou non paramétrique $Y = f(S)$ pourra, par exemple, être construit *via* une *analyse structurale*. Ce modèle représentera le résultat de la transformation des données initiales en des données d'ordre sémantique élevé (appelé *connaissance* en apprentissage automatique), *via* une phase d'apprentissage correspondant à une généralisation des observations.

Une prédiction de la valeur de Y pourra être effectuée à partir de nouvelles observations d'un ensemble S' de variables. La connaissance de ces valeurs pourra alors éventuellement être mise en correspondance avec les valeurs réellement observées, ce qui nous permettra

de valider, d'invalider, ou de faire évoluer le modèle construit pendant la phase d'apprentissage.

L'invalidation du modèle est une conséquence directe de l'une des différentes phases précédentes :

Pour la phase de *définition des variables* et d'instrumentation, l'utilisateur peut se demander s'il a bien pris en compte l'ensemble des variables pertinentes pour son étude, et peux également remettre en question le choix des capteurs utilisés.

Il peut également remettre en question le filtrage utilisé, ou tout simplement le choix du modèle utilisé.

Enfin, G.J. KLIR considère un troisième niveau (**Niveau 3**) appelé **Système-Structure**. À ce niveau, le système est composé d'un ensemble de sous-systèmes générateurs qui interagissent entre eux et qui sont susceptibles de véhiculer une quantité importante d'information.

Les différents niveaux de la hiérarchie étant définis, toute approche d'analyse n'est autre qu'un déplacement dans cette hiérarchie. Le problème de la modélisation d'un système complexe revient alors à passer du niveau épistémologique 1 (observation du système) à un niveau supérieur.

Cette première partie est constituée de deux chapitres.

Dans le premier chapitre, nous nous proposons d'introduire certaines définitions des termes employés par les cogniticiens et les automaticiens. Nous présenterons ainsi les données d'apprentissage sur lesquelles va se baser l'analyse du système.

Dans le deuxième chapitre, nous nous proposerons de présenter les méthodes de l'apprentissage dans le cadre d'une analyse structurale d'un système physique plus ou moins complexe.

Chapitre 1

Du système aux données d'apprentissage

Dans ce premier chapitre, nous nous proposons de définir certains concepts essentiels à une bonne compréhension des chapitres suivants.

À ce titre, nous définissons dans un premier temps les notions de système physique et de système cognitif en spécifiant en particulier le vocabulaire utilisé d'une part, par la communauté de l'apprentissage, et par la communauté automaticienne d'autre part. Nous remarquerons par exemple que le terme « environnement » prend un sens complètement différent suivant la communauté scientifique à laquelle on appartient.

Dans un deuxième temps, nous nous intéresserons à la phase préliminaire d'étude d'un système, dans laquelle l'expert est confronté à un choix de variables bien souvent guidé par son « savoir-faire ». Ce choix étant fait, il pourra passer à la phase d'observation de son système sans laquelle aucune étude ne peut être réalisée. C'est à ce niveau qu'on distinguera les systèmes statiques des systèmes dynamiques. Nous définirons dès lors les notions de « signal » et de « données », notions capitales pour la suite de l'étude.

Suivant la nature des variables en question, la notion de finesse nous permettra de quantifier intuitivement l'information que peut apporter une variable sur le système. Certaines propriétés sur la relation de finesse entre variables seront dès lors démontrées.

Enfin, dans l'optique du chapitre 2, nous définirons les notions de variables à expliquer et de variables potentiellement explicatives en nous focalisant tout spécialement sur les systèmes dynamiques.

La notion d'incohérence des données sera alors introduite, et un tableau de contingence sera construit, ce qui permettra de traduire de façon claire cette incohérence.

1.1 Système physique / Système cognitif

Dans une démarche générale d'apprentissage, l'apprenant qui peut être un humain, une créature dotée d'une certaine intelligence ou enfin une machine plus ou moins intelligente,

est appelé de manière générale, **système** (ou **système cognitif**). Ce système tente d'apprendre ou de comprendre l'**environnement** (encore appelé **monde réel**) dans lequel il évolue. Cet environnement dépend fortement du contexte. Il peut s'agir d'un domaine très restreint ou au contraire très étendu incluant le système cognitif lui-même.

Pour l'automaticien, le terme « environnement » prend un tout autre sens. Dans ce cadre, le système cognitif tente d'apprendre le comportement d'un **système physique** (existence d'un flux d'informations du système physique vers le système cognitif) qui échange des informations avec son environnement immédiat.

Un problème de vocabulaire est dès lors soulevé, suivant que l'on se place dans le cadre de l'automatique, ou dans le cadre de l'apprentissage (figure 1.1).

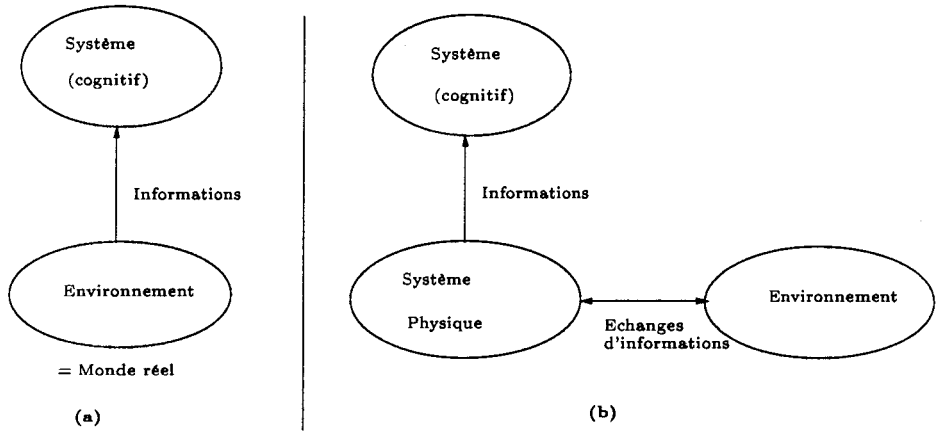

FIG. 1.1: *Quelques termes utilisés par la communauté scientifique travaillant dans le cadre de l'apprentissage (a) et travaillant dans le cadre de l'automatique (b)*

Afin de pallier d'éventuelles ambiguïtés, nous utiliserons les termes « système physique » (et non pas *environnement* ou *monde réel*) pour parler du système sur lequel on désire apprendre quelque chose, et « système cognitif » pour parler du système apprenant.

1.2 Choix des variables

Pour G.J. KLIR, un système au niveau épistémologique le plus bas, est appelé **Système-Source**. Il constitue le **Niveau 0** et correspond à une phase préparatoire. À ce niveau, sont définis l'ensemble des variables du système ainsi que leurs ensembles de modalités (ou états) respectifs. Le choix de ces variables dépend avant tout de l'expérience de l'expert et des connaissances qu'il possède déjà sur le système.

Ainsi, C.H. COOMBS [Coo64] écrit : « *si on demande à un individu s'il pense voter pour le candidat A, l'observateur, en général, enregistre sa réponse, OUI ou NON ; mais on peut se demander s'il n'y aurait pas intérêt à enregistrer le temps de réponse, ou les changements du rythme respiratoire, ou la réaction électro-dermale de l'individu, ou ce que celui-ci fait de ses mains ...* ». Parmi toutes ces variables, l'expert doit choisir, choix souvent guidé

par son « savoir-faire ». Nous verrons qu'à partir du niveau épistémologique 2, le choix de variables pertinentes (prises parmi les variables initiales) pour l'étude du système pourra être guidé.

1.3 Les données d'apprentissage

1.3.1 Base de données et ensemble d'apprentissage

Une **base de données** (database) est un stockage « *fiable* » (ce terme sera quelque peu précisé ultérieurement) d'information. C'est en général un tableau constitué de L **individus**[1] $\times$ N **variables**[2].

Dans le cadre d'une étude d'un système physique, ce tableau est construit à partir de l'observation de celui-ci : pour une observation donnée, à une variable correspond la valeur de sortie d'un capteur.

Cette base de données représente un échantillon considéré comme étant plus ou moins représentatif de l'ensemble des données potentiellement observables[3] sur le système étudié. Le cogniticien choisira ainsi soigneusement cet ensemble de données. En effet, une bonne étude d'un système physique commence nécessairement par l'établissement d'une bonne base de données ; dans le cas opposé, cette étude sera dénuée de sens.

Une distinction est parfois faite entre ce qu'on qualifie de base de données et l'**ensemble d'apprentissage** (training set). La base de données est constituée des informations brutes issues des capteurs. On parlera d'ensemble d'apprentissage[4] lorsqu'il y a recodage de la base de données dans le but de l'utiliser sur calculateur dans le cadre d'un apprentissage.

1.3.2 Système statique / Système dynamique

Dans le cadre d'une analyse structurale d'un système physique, ce niveau d'étude correspond, pour G.J. KLIR, au **Niveau 1**[5]. Ce niveau correspond à une série d'observations effectuées sur le système correspondant à une succession de modalités ou états pris par les variables, et qui constituent le tableau initial des données que nous nous proposons d'analyser. C'est à ce niveau qu'on peut distinguer les systèmes statiques des systèmes dynamiques :

- Un système statique est un système dont les observations ne dépendent pas du temps. L'évolution temporelle de ce système est inexistante (ou quasi-inexistante). C'est en général sur ce type de tableaux que travaille la communauté cognitivienne, mis à part la communauté des réseaux neuronaux. Dans ce cas, les individus du tableau

1. encore appelés **objets** ou **observations**
2. encore appelées **paramètres** ou **attributs**
3. encore appelées « **réelles** » par le cogniticien
4. ou **population d'apprentissage**
5. encore appelé **Système-Données**

initial des données ne sont pas ordonnés et les lignes d'un tel tableau peuvent être permutées ou déplacées à la convenance de l'utilisateur.

- Un système dynamique est un système possédant une évolution temporelle non négligeable. Dans ce cas, l'échantillonnage des observations est effectuée en général avec un pas constant et celles-ci sont liées par une contrainte temporelle, c'est-à-dire que les individus sont ordonnés dans le temps : on ne peut pas permuter les lignes ou modifier l'ordre d'apparition de celles-ci dans le tableau initial des données. En tant qu'automaticien, nous nous intéressons tout particulièrement à ces systèmes dynamiques.

1.3.3 Signal ou donnée?

Dans le cas d'un système physique, un capteur délivre un signal. La distinction faite entre les termes « base de données » et « ensemble d'apprentissage » prend tout son sens en prenant en compte le fait que :

- Ce signal peut être utilisé directement dans l'étude. Il entre à la fois dans la base de données et dans l'ensemble d'apprentissage sur lequel l'expert travaille.

- Ce signal peut être traité. Il subira alors une (ou plusieurs) transformation(s). À titre d'exemples, une transformée en ondelettes, une transformée de Fourier, un filtrage, etc... permettront de mettre en évidence certaines formes caractéristiques du signal non perceptibles *a priori*. Dans ce cas, le signal formera la base de données initiale, et sa transformation (associée éventuellement au signal de départ) constituera l'ensemble d'apprentissage.

  De même, certains paramètres caractéristiques du signal pourront être extraits de celui-ci. En effet, dans de nombreuses applications (notamment les applications médicales), le signal en lui-même n'a pas une grande signification pour l'expert, qui préfère se référer à certaines caractéristiques qu'il considère comme pertinentes pour son étude. À titre d'exemples, à partir du signal de la figure 1.2, on peut extraire la moyenne, la période (T), la valeur maximale (V_{max}), minimale (V_{min}), la tendance... Ces caractéristiques constitueront alors, plus que le signal, son ensemble d'apprentissage.

Il est bien entendu évident que toute transformation du signal initial n'a de sens que si une justification de cette transformation *a posteriori* (après l'étude complète) est fournie. Nous reviendrons sur ce point ultérieurement (dans le chapitre 2).

1.3.4 Les variables primaires

Soit $\Sigma^0 = \{X_1, X_2, ..., X_N\}$ l'ensemble des N *variables* importantes, pertinentes pour la description du système. Dans le cadre de l'étude d'un système dynamique, il peut s'agir

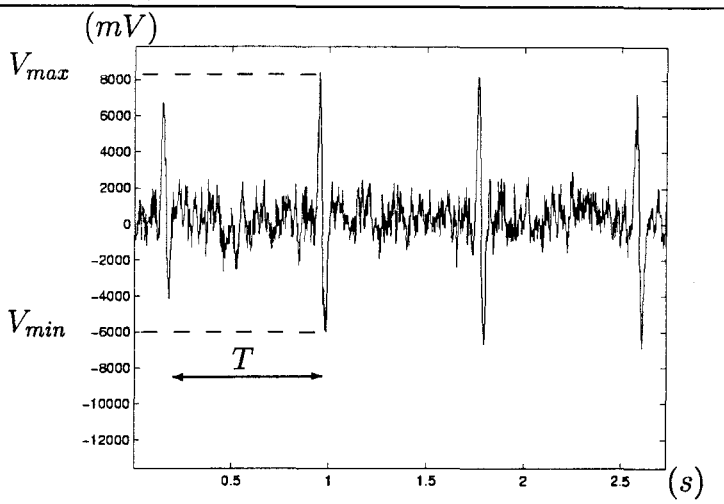

FIG. 1.2: *Exemple de signal*

des variables d'entrée, de sortie, de commande, d'état ou enfin de paramètres: constante de temps, amortissement, niveau de saturation, etc...

Ω^o représente l'ensemble de tous les individus potentiellement observables (toutes les observations possibles de Σ^0). Cet ensemble induit pour chaque variable primaire X_i un ensemble de modalités observables $\mathcal{O}_{X_i}$, qui peut être caractérisé par plusieurs structures, suivant la nature de la variable en question. Celle-ci peut être:

- ***quantitative*** (ou numérique: $\mathcal{O}_{X_i} \subset \mathbb{R}$): par exemple la température, la taille, ... , sont des variables quantitatives. Dans ce cas, $\mathcal{O}_{X_i}$ est infini;

- ***qualitative ordinale*** (ou ordonnée): $(\mathcal{O}_{X_i}, \leq)$ est un ordre total. À titre d'exemples, la variable « *taille* » composée des modalités (*petit, moyen, grand*), le « *poids* » composé des modalités (*léger, lourd*), ... , sont des variables qualitatives ordinales;

- (qualitative) ***nominale***: aucune relation n'est *a priori* définie sur $\mathcal{O}_{X_i}$; par exemple, la couleur des yeux, le mode de fonctionnement d'un système, les variables booléennes, ... , sont des variables nominales (sous-entendu « qualitatives »);

  ces deux dernières structures peuvent être imposées par l'expert du système, mais elles peuvent également être obtenue par classification non supervisée sur des données quantitatives. On parle alors de *discrétisation* des données initiales;

- ***structurée***: il existe une ou plusieurs relations(s) sur $\mathcal{O}_{X_i}$; par exemple, on peut définir une structure *taxonomique* (fournie par l'expert ou par une classification) sur les modalités de la variable « *humain* » (figure 1.3).

On peut également combiner un ordre partiel et une taxonomie (figure 1.3): (*bébé* < *enfant* < *adolescent* < *adulte*).

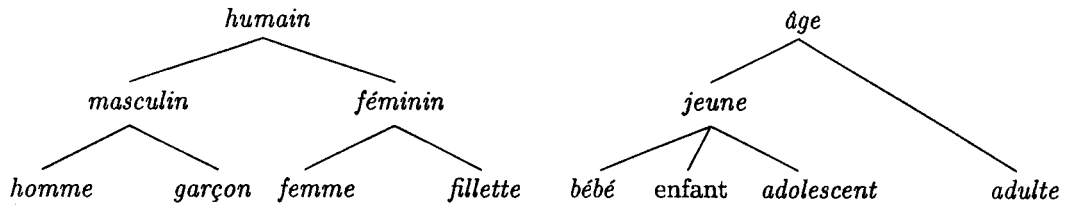

FIG. 1.3: *Exemples de variables structurées*

Nous pouvons constater que le domaine des variables structurées est extrêmement vaste.

Les grandeurs caractéristiques d'un système physique évoluant continûment dans le temps sont par définition même, des grandeurs continues.

Dans le cadre d'une étude du système, cette grandeur est appréhendée par un (ou plusieurs) capteur(s) fournissant ainsi une image de cette grandeur physique, avec une certaine précision. De plus, dans le cas de capteurs numériques, l'information est *a fortiori* discrétisée par le Convertisseur Analogique-Numérique, caractérisé, entre autres, par son *quantum* élémentaire représentant, par définition, la plus petite variation de l'entrée du CAN permettant de faire varier la valeur du bit de poids faible. À titre d'exemple, une température est une grandeur qui évolue continûment, et ne sera appréhendée qu'avec un *quantum* élémentaire de 0.01 °C. Avec l'évolution des techniques, ce *quantum* a tendance à diminuer ; mais cette discrétisation de l'information est bel et bien un fait certain.

De même, l'utilisation de grandeurs continues sur machine informatique nécessite une discrétisation de l'information continue.

Ces remarques nous conduisent donc tout naturellement à considérer les données comme des données qualitatives (dans le sens d'une discrétisation de l'information continue de base).

Afin de constituer la population d'apprentissage, un ensemble d'observations est effectué sur ces variables, et sera appelé $\Omega = \{\omega_1, \omega_2, ..., \omega_L\}$ de taille L ($\Omega \subset \Omega^o$ est l'ensemble des observations réellement effectuées), formant ainsi le tableau initial des données (tableau 1.1).

Ω \ Σ^0	X_1	...	X_i	...	X_N
ω_1					
$\vdots$			$\vdots$		
ω_j		...	$X_i(\omega_j)$	...	
$\vdots$					
ω_L					

TAB. 1.1: *Tableau initial des données*

Dans ce tableau, $X_i(\omega_j)$ désigne la modalité (la valeur) prise par la variable X_i pour l'échantillon ω_j.

Nous appellerons $M_{X_i} = \{\alpha_k^i, k = 1, ..., m_i\}$ $\forall i \in \{1...N\}$ $(m_i = \text{card}(M_{X_i}))$ l'ensemble fini des modalités observées de X_i $(\forall i \in \{1...N\})$. On a alors la relation suivante: $M_{X_i} \subset \mathcal{O}_{X_i}$. Nous définissons dès lors X_i comme une application de Ω dans M_{X_i} :

$$\begin{array}{rccc} X_i: & \Omega & \longrightarrow & M_{X_i} \\ & \omega & \longmapsto & X_i(\omega) \end{array}$$

Cet ensemble de modalités observées M_{X_i} est fini et ne peut par conséquent pas représenter l'ensemble des modalités observables de la variable X_i qui peut être infini dans le cas de variables quantitatives. Ceci peut être dû à plusieurs phénomènes:

- la précision avec laquelle on mesure X_i est faible;
- il n'y a pas assez d'observations de la variable X_i; nous sommes dans ce cas en présence d'une petite base de données; on peut donc se poser la question (*a posteriori*) de la pertinence d'un apprentissage dans ce cas;
- certaines modalités de X_i ne seront jamais observées, suite à un ensemble de contraintes entre les variables par exemple;

Remarque: Cas d'un système dynamique

Un cas important est celui d'un système dynamique pour lequel une observation est effectuée à chaque instant d'échantillonnage. L'ensemble d'apprentissage devient donc $\Omega = \{t_0, t_0+\Delta t, \ldots, t_0+(L-1)\cdot\Delta t\}$, où t_0 est le premier instant d'apprentissage, Δt la période d'échantillonnage (en général constante) et L la taille de la population d'apprentissage.

Afin de prendre en compte les variables aux différents instants, un opérateur « *retard* » est défini. Cet opérateur consiste simplement à décaler la colonne des modalités de la variable en question vers le bas (d'une position si on considère un retard d'un instant d'échantillonnage; de deux positions, si on considère la variable retardée de deux instants d'échantillonnage, ...). Ainsi, Σ^k représente l'ensemble des variables X_j $(j = 1, \ldots, N)$ retardées de k périodes d'échantillonnage. Le système est alors étudié à partir de l'ensemble des variables Σ:

$$\Sigma = \bigcup_{i=0}^{o_{max}} \Sigma^i$$

où o_{max} est l'ordre au-delà duquel l'étude du système n'a plus de signification physique. Cet ordre est déterminé par l'expert. Dans le cas très particulier d'un système statique, $o_{max} = 0$.

Deux situations sont dès lors à prendre en compte:

- N est petit et $\text{card}(M_{X_i})$ est petit $\forall i = 1, \ldots, N$, dans ce cas, l'explosion combinatoire du nombre des modalités vectorielles potentiellement observables est gérable [Pom91]

et cette notation semble raisonnable. En effet, le nombre de modalités potentiellement observables est égal à :

$$\prod_{j=0}^{o_{max}} \left(\prod_{i=1}^{N} m_i \right)$$

- Dans le cas contraire, cette explosion combinatoire du nombre des modalités vectorielles potentiellement observables en utilisant cette notation devient difficilement appréhendable. Afin de prendre en compte l'aspect dynamique d'une variable, on approxime la notion de « dérivée » si aucun modèle mathématique de la variable en question ne peut donner une expression de celle-ci. Plusieurs outils développés dans la littérature nous permettent d'approximer plus ou moins convenablement la dérivée. L'outil utilisé pour cette approximation dépend fortement de l'allure du signal initial. À titre d'exemple, la dérivée d'un signal provenant d'un ECG (électrocardiogramme) ne sera pas approximée comme la dérivée d'un signal provenant d'un capteur de température. En effet, dans le premier cas, le signal est beaucoup plus perturbé et la notion de tendance à plus ou moins long terme semble plus appropriée que la notion de dérivée.

  Dans le cadre de nos applications, nous nous proposerons dans le chapitre 5 de l'approximer par $X(i) - X(i-1)$ où $X(i)$ est la valeur de la variable X à l'instant d'échantillonnage i.

  Le nombre de dérivées successives à prendre en compte constituera l'ordre de notre système.

  D'autres travaux consistent actuellement à effectuer une analyse « temps-fréquence » à partir de laquelle un temps caractéristique de chaque variable pourra être estimé, ce qui nous permettra de distinguer l'évolution d'une variable à court terme de son évolution à long terme [CCPR97].

Remarque : Cas des données manquantes

Une autre réalité physique est l'absence de données pendant certains intervalles de temps. Ce phénomène peut être dû, entre autres, à un débranchement du capteur, à un problème de transmission des données, à des interférences venant se superposer au signal pertinent, ... Deux techniques sont utilisées dans la littérature afin de pallier ce problème :

- La valeur manquante est remplacée par une modalité supplémentaire, qui sera en général la même pendant toute la phase d'observation du système.

- Un « lissage » des données peut être effectué. À titre d'exemples, on pourra remplacer la donnée manquante par la moyenne de la variable en question, ou par une approximation polynomiale (avec un ordre déterminé à l'avance) en prenant en compte les valeurs connues précédentes et/ou suivantes.

Si un phénomène bien particulier (une panne par exemple) se produit pendant cette phase de « non observation » de la variable en question, la deuxième méthode sera complètement « aveugle » vis-à-vis de celui-ci. La première méthode mettra simplement en évidence un problème dû au fait que la donnée n'a pas pu être transmise.

Il est bien évident qu'une justification de la méthode employée afin de remplacer ces données manquantes doit être formulée *a posteriori*, et qu'elle n'est pas évidente.

1.3.5 Finesse d'une variable primaire

Étant donnée une variable primaire X_i, il est toujours possible d'en créer d'autres, en partitionnant l'ensemble des modalités M_{X_i}. Soit $M_{\tilde{X}_i}$ une partition de M_{X_i}. On notera $\tilde{X}_i$ la variable déduite de X_i, dont l'ensemble des modalités est $M_{\tilde{X}_i}$. L'ensemble $\mathbb{P}(M_{X_i})$ des partitions de M_{X_i}, ordonné par la relation de finesse $\preccurlyeq$, constitue un treillis[6] [Sza62, Bir67, BM70, KB78] dont M_{X_i} est le plus petit élément (minorant universel). On dira que la variable $\tilde{X}_i$ est plus grosse que X_i ($\tilde{X}_i \succcurlyeq Xi$), puisque chacune des classes de M_{X_i} est entièrement incluse (par définition) dans une classe de $M_{\tilde{X}_i}$. On peut noter que, pour tout élément de la population d'apprentissage, la connaissance de la modalité prise par X_i implique la connaissance de celle prise par $\tilde{X}_i$ (figure 1.4).

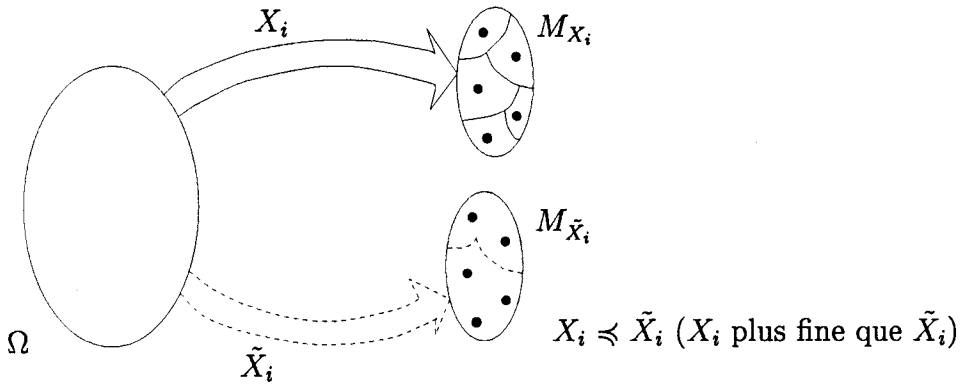

FIG. 1.4: *Finesse d'une variable primaire*

À titre d'exemple, à partir de la variable primaire *âge* (à 4 modalités) de la figure 1.3, on peut créer d'autres variables plus grosses, dont certaines sont données ci-dessous :

$\hat{a}ge_1$: {*bébé, enfant ou adulte, adolescent*}

$\hat{a}ge_2$: {*bébé ou enfant ou adolescent, adulte*}

$\hat{a}ge_3$: {*jeune ou adulte*}

6. L'ordre $(\mathbb{P}(E), \preccurlyeq)$ est muni d'une structure de treillis $\Leftrightarrow \forall p, p' \in \mathbb{P}(E) \quad \exists\, p \wedge p'$ (*infimum*) et $\exists\, p \vee p'$ (*supremum*). Le treillis peut alors être représenté par un diagramme de Hasse.

Remarques :

- Compte tenu de la structure de la variable primaire, le label « *jeune* » peut désigner la classe {*bébé ou enfant ou adolescent*}, de sorte que l'ensemble des modalités de la variable $\hat{a}ge_2$ peut s'écrire plus simplement : {*jeune, adulte*};
- Dans cet exemple, en créant des variables plus grosses que la variable {*bébé, enfant, adolescent, adulte*}, on n'a regroupé que des modalités consécutives (ex. : *enfant ou adolescent*), mais on aurait tout aussi bien pu ne pas considérer l'ordre sur ces modalités et regrouper celles-ci dans un ordre quelconque (ex. : *enfant ou adulte*). Pourtant, dans un but de simplification d'un éventuel modèle du système étudié, prendre en compte l'ordre sur les modalités nous permettra de résumer l'information (en écrivant par exemple « *jeune* » au lieu de « *bébé ou enfant ou adolescent* »). Dans le cas contraire, nous ne pouvons que réécrire l'information en extension.
- La variable $\hat{a}ge_3$ ne possède qu'une modalité. Elle constitue le plus grand élément (majorant universel) du treillis. Elle est constante, puisqu'elle prend la même modalité pour tous les objets de la population d'apprentissage. Elle n'apporte par conséquent aucune information sur le système à étudier.

À partir de cette dernière remarque, nous pouvons introduire intuitivement la notion d'information.

Notion intuitive d'information apportée par une variable sur le système étudié :

Le treillis des partitions de la variable $\hat{a}ge$ est représenté sur la figure 1.5. Il est à noter que plus on descend vers le bas du graphique, plus la variable considérée apporte de l'information sur le système. En effet, à titre d'exemple, il est clair que la variable $\hat{a}ge_2$ apporte moins d'information sur le système que la variable $\hat{a}ge_1$.

En d'autres termes, plus les modalités sont décomposées, plus on apporte d'information sur le système étudié.

1.3.6 Variables primaires et partitions de la population d'apprentissage

À chaque variable primaire $X_i \in \Sigma^0$ de notre système, nous pouvons associer une partition de la population d'apprentissage Ω, de la façon suivante (figure 1.6).

$$P_{X_i}(\Omega) = \{X_i^{-1}(u), u \in M_{X_i}\}$$

où $X_i^{-1}(u) = \{\omega | Xi(\omega) = u\}$.

L'ensemble $\mathbb{P}(\Omega)$ des partitions de Ω, muni de la relation de finesse $\preccurlyeq$, possède une structure de treillis. Ainsi, il est possible de construire une relation d'ordre et une relation d'équivalence entre les variables.

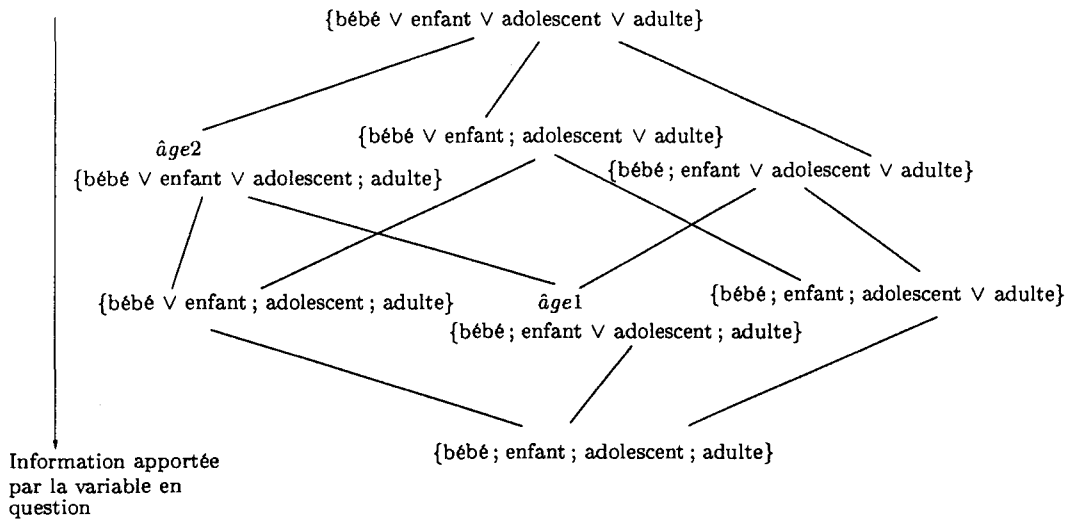

FIG. 1.5: *Treillis des partitions engendrées par la variable « âge »*

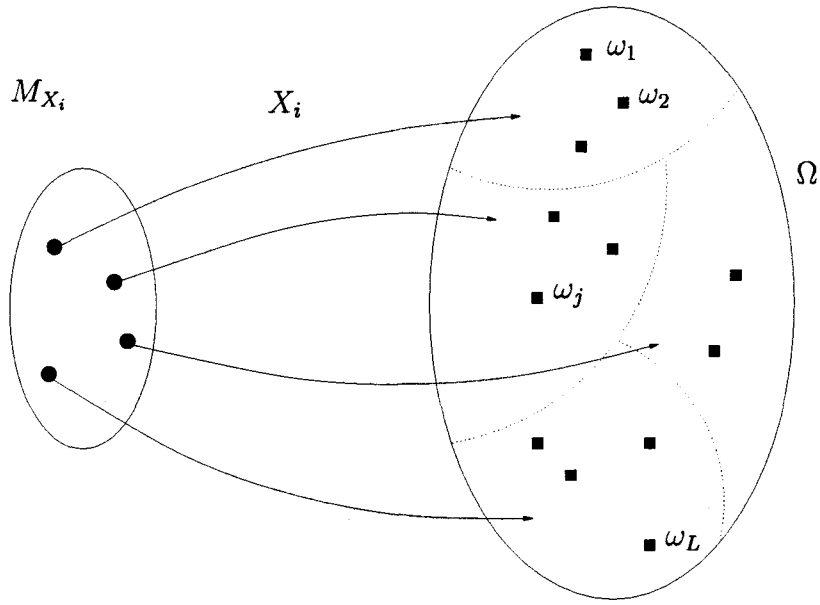

FIG. 1.6: *Partition de la population d'apprentissage engendrée par la variable X_i*

Considérons X_i et X_j $(i \neq j)$ deux variables de notre système. Nous définissons :

$$X_i \equiv X_j \Longleftrightarrow P_{X_i}(\Omega) = P_{X_j}(\Omega)$$

On dit que X_i et X_j sont des variables équivalentes pour l'étude du système.

Autrement dit, ces deux variables apportent autant d'information l'une que l'autre sur le système.

À titre d'exemple, si on considère le tableau 1.2,

Ω	*forme*	*température*	*précision*	s
ω_1	hexagonale	200 °C	1.5×10^{-3}	0
ω_2	carrée	500 °C	10^{-4}	1
ω_3	hexagonale	200 °C	1.5×10^{-3}	0
ω_4	hexagonale	200 °C	1.5×10^{-3}	0
ω_5	carrée	500 °C	10^{-4}	1

TAB. 1.2: *Un exemple de tableau de données*

toutes les variables en question apportent la même quantité d'information sur le système ; il suffit pour s'en convaincre de recoder les modalités (ex. : hexagonale $\rightarrow$ 200 °C ; carrée $\rightarrow$ 500 °C ; ...).

Propriété :

$$X_i \preccurlyeq X_j \Longleftrightarrow P_{X_i}(\Omega) \preccurlyeq P_{X_j}(\Omega)$$

où $\preccurlyeq$ représente la relation d'ordre partiel entre partitions « être plus fine que ».

Autrement dit, si X_j est une variable primaire plus grosse que X_i, la partition de Ω associée à (induite par) X_j est également plus grosse que celle induite par X_i.

Par contre, en considérant l'exemple précédent, la variable vitesse ci-dessous (tableau 1.3) apporte, elle, plus d'information que les autres sur le système étudié.

Ω	*vitesse*
ω_1	15 rad/s
ω_2	20 rad/s
ω_3	15 rad/s
ω_4	10 rad/s
ω_5	20 rad/s

TAB. 1.3: *La variable vitesse apporte plus d'information que les autres variables*

La partition induite sur Ω par la variable vitesse est $\{(\omega_1, \omega_3), (\omega_2, \omega_5), (\omega_4)\}$. Cette partition est plus fine que la partition induite sur Ω par l'une des variables précédentes :
$\{(\omega_1, \omega_3), (\omega_2, \omega_5), (\omega_4)\} \preccurlyeq \{(\omega_1, \omega_3, \omega_4), (\omega_2, \omega_5)\}$.

Remarque : Nous nous limitons volontairement à de petits tableaux afin de présenter les concepts utilisés. Dans la réalité, nous pouvons être confrontés à de gigantesques tableaux.

1.3.7 Variables vectorielles (ou multidimensionnelles)

$\Sigma = \bigcup_{i=0}^{o_{max}} \Sigma^i$ étant l'ensemble des N variables primaires retardées ou non, nous noterons $\mathcal{P}(\Sigma)$ l'ensemble de tous les vecteurs possibles dont les composantes sont des composantes élémentaires de Σ. Chaque élément S de $\mathcal{P}(\Sigma)$ représente une application de Ω dans son ensemble de modalités M_S :

$$\begin{array}{rccll} S: & \Omega & \longrightarrow & M_S & S \in \mathcal{P}(\Sigma) \\ & \omega & \longmapsto & S(\omega) & \end{array}$$

Une modalité de S sera alors la conjonction des modalités des variables primaires constituant S.

$$P_S(\Omega) = \bigcap_{X_i \in S} P_{X_i}(\Omega)$$

Nous pouvons alors en déduire la propriété de finesse suivante :

Propriété :

Soient S_1 et S_2 appartenant à $\mathcal{P}(\Sigma)$, telles que S_1 soit un sous-vecteur de S_2 (noté abusivement ici $S_1 \subset S_2$). On a alors les relations suivantes :

$$M_{S_1} \succcurlyeq M_{S_2} \text{ et } P_{S_1}(\Omega) \succcurlyeq P_{S_2}(\Omega).$$

En d'autres termes, la relation d'inclusion entre sous-ensembles de variables induit une relation de finesse entre les variables vectorielles et les partitions associées de la population d'apprentissage.

À titre d'exemple, sur le tableau suivant (extrait de [Min89], p.320, et représentant les modalités de deux attributs à partir de données sur le cancer du sein, tableau 1.4) :

Ω	*radiation*	*menopause*	*class*
ω_1	no	<60	recur
ω_2	no	$\geq$ 60	recur
ω_3	no	<60	recur
ω_4	no	not	recur
ω_5	yes	$\geq$ 60	not recur
ω_6	yes	<60	not recur
ω_7	yes	$\geq$ 60	not recur
ω_8	no	not	not recur
ω_9	no	<60	not recur
ω_{10}	no	<60	recur

TAB. 1.4: *Exemple de tableau de données*

on pourra constater que le vecteur de variables (*radiation,ménopause*) induit une partition de Ω plus fine que la variable *radiation* seule :

$\{(\omega_1, \omega_3, \omega_9, \omega_{10}), (\omega_2), (\omega_4, \omega_8), (\omega_5, \omega_7), (\omega_6)\} \preccurlyeq \{(\omega_1, \omega_2, \omega_3, \omega_4, \omega_8, \omega_9, \omega_{10}), (\omega_5, \omega_6, \omega_7)\}$

1.4 Préparation des données en vue d'un apprentissage

1.4.1 Variables à expliquer / Variables explicatives

Que ce soit dans le cadre de la surveillance d'un processus industriel complexe, ou de la surveillance d'un patient, l'expert a nécessairement besoin de confronter les données qu'il observe aux données fournies par son modèle, qu'il soit mathématique ou pas. L'établissement de ce modèle est d'abord basé sur une phase d'observation du système (paragraphe 1.3.4).

Ce modèle exprime certaines relations entre les variables considérées pertinentes ce qui nous conduit à partitionner l'ensemble des variables primaires en deux classes distinctes :

- Appelons Y ($Y \in \Sigma$) la (ou les) variable(s) *à expliquer*. Il peut s'agir par exemple d'une variable endogène, et/ou difficilement observable (ou d'acquisition coûteuse, ou souvent bruitée, ...).
- $X = (X_1, X_2, \ldots, X_p)$ (où $X_i \in \Sigma \backslash Y$) représente le vecteur des p variables permettant, tout au moins on l'espère, d'expliquer Y. Elles seront appelées variables *explicatives*. Celles-ci peuvent représenter par exemple des variables exogènes, et/ou facilement observables (ou encore d'acquisition non coûteuse, fiables, ...).

Nous pouvons alors reformuler le tableau des données (tableau 1.5)

Ω \ Σ	X	Y
	$[X_1 \ \ldots \ X_i \ \ldots \ X_p]$	
ω_1		
$\vdots$	$\vdots$	
ω_j	$\ldots \ X_i(\omega_j) \ \ldots$	$Y(\omega_j)$
$\vdots$		
ω_L		

TAB. 1.5: *Réécriture du tableau initial des données en vue d'un apprentissage*

pour lequel :

- $\Sigma = \{X_1, X_2, \ldots, X_p, Y\}$ est l'ensemble des variables *pertinentes* (retardées ou pas) pour l'étude du système ;
- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_L\}$ est la *population d'apprentissage* (de taille L) ;
- $X_i(\omega_j)$ (resp. $Y(\omega_j)$) est la *modalité* de la variable X_i (resp. Y) correspondant à l'*observation* ω_j.

Nous utiliserons tantôt la première représentation de Σ ($\Sigma = \bigcup_{i=0}^{o_{max}} \Sigma^i$), tantôt la seconde ($\Sigma = \{X_1, \ldots, X_p, Y\}$), celle-ci étant plus adaptée au problème d'explication

(nous avons tout simplement considéré toutes les variables retardées de i instants ($i = 0, \ldots, o_{max}$) du tableau de données défini à la figure 1.1 et renommé un certain nombre d'entre elles « Y »).

Cas particuliers :

- Dans le cas particulier où Y ne représente qu'une seule variable, alors on obtient $p = (o_{max} + 1) \cdot N - 1$ variables potentiellement explicatives.

- Dans le cadre particulier du diagnostic (médical ou industriel), les variables primaires composant le vecteur X sont les variables observées retardées de i instants d'échantillonnage ($i = 0, \ldots, o_{max}$), et la variable Y représente les différents modes de fonctionnement (voir l'introduction générale) qu'on cherchera à expliquer.

Notations utilisées :

- L'ensemble des modalités de la variable X_i sera noté $M_{X_i} = \{\alpha_1^i, \alpha_2^i, \ldots, \alpha_{n_i}^i\}$ ($i = 1, \ldots, p$; $n_i = \text{card}(M_{X_i})$).

- $M_X = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ représentera l'ensemble des modalités de la variable vectorielle X ($n = \text{card}(M_X)$). Ces modalités sont des conjonctions des modalités des différentes variables initiales $X_i (i = 1, \ldots, p)$.

- $M_Y = \{\beta_1, \beta_2, \ldots, \beta_m\}$ est l'ensemble des modalités de la variable à expliquer Y. Ces modalités représentent, par exemple, des modes de fonctionnement en diagnostic (*normal, faute n° 1, faute n° 2, ..., faute n° m − 1*), ou des classes d'appartenance en Reconnaissance des Formes (*classe 1, classe 2, ..., classe m*).

  Le cardinal de M_Y ($\text{card}(M_Y = m)$) est en général petit. En effet, pour diverses raisons (simplification du problème, obtention de simples résultats qualitatifs, ...), l'expert peut être amené à créer une partition de l'ensemble des modalités de Y. Chaque modalité $\beta_j (j = 1, \ldots, m)$ représentera alors une classe de la partition ainsi créée.

  À titre d'exemple, si on essaie d'expliquer la quantité de calories apportées par certaines boissons en fonction d'autres caractéristiques, on peut être amené à discrétiser l'échelle des calories : − (*peu de calories*), *o* (*pas d'excédent de calories*), + (*beaucoup de calories*).

  Dans le cadre d'un diagnostic industriel, on pourra par exemple, lors d'une étude préliminaire du système, remplacer les modalités (*normal, faute n° 1, faute n° 2, ..., faute n° m − 1*) par (*normal, faute*).

  Dans le cadre d'un diagnostic médical, le médecin pourra d'abord estimer l'état du patient comme étant *bon* ou *anormal*.

- En regard d'une modalité β_j de Y, l'ensemble $\{\omega|Y(\omega) = \beta_j\}$ représente une *classe* (ou *concept*) de l'ensemble d'apprentissage, encore appelé l'*ensemble des exemples* (en apprentissage), sous entendu de β_j. On parle de *prototypes* (en théorie de la décision). Les autres observations sont appelés *contre-exemples*. Certains auteurs utilisent les termes *exemples positifs* (notés P ou $+$) pour les exemples, et *exemples négatifs* (notés N ou $-$) pour les contre-exemples.

  L'ensemble des classes $\{\omega|Y(\omega) = \beta_j\}$ forme une partition de Ω. On dit que l'ensemble des classes induites par les modalités β_j $(j = 1, \ldots, m)$ de Y constitue une *hypothèse* [Ren86].

1.4.2 Notion d'incohérence des données d'apprentissage

La prise en compte du bruit et/ou des erreurs est un passage obligé dans l'étude d'un système complexe et ce problème est traité dans une littérature abondante. Un modèle mathématique du bruit est en général considéré dans le but de le filtrer. À titre d'exemples, on peut considérer un bruit gaussien, uniforme, ... Ces méthodes reposent sur cette hypothèse modélisatrice souvent trop forte.

Ce constat est à l'origine des études réalisées sur des bases de données incohérentes.

La cohérence signifie qu'il n'existe aucune contradiction entre les individus de la base de données (entre les observations du système physique). C'est-à-dire que le système étudié est déterministe. Dans ce cadre, le cas de données dites « bruitées » se traduira par une incohérence de la base de données. En ce qui nous concerne, nous distinguerons ultérieurement deux cas de figure entraînant une incohérence de la base de données :

- le cas où les données sont effectivement biaisées par du bruit venant se superposer au signal pertinent ;
- le cas dans lequel l'incohérence des données est due au fait qu'une partie des variables informatives du système en question n'a pas été prise en compte.

Ces deux cas peuvent bien entendu se superposer.

Afin d'illustrer cette notion de cohérence, la figure 1.7 donne un exemple de données cohérentes et incohérentes. Dans le cas (b), à un couple de modalités de (X_1, X_2) correspondent deux conclusions différentes « $\square$ » et « $\bullet$ ».

1.4.3 Le tableau de contingence

À partir du chapitre 2 nous chercherons à caractériser des relations liant Y à $S \subset \tilde{X}$, où $\tilde{X}$ est une variable plus grosse que X.

Dans ce but, et comme l'incohérence des données est un fait indéniable dans la plupart des bases de données réelles, nous réécrivons le tableau 1.5 des données sous la forme d'un *tableau de contingence* (ou tableau des *fréquences relatives*) P_{IJ} de X et Y, représenté

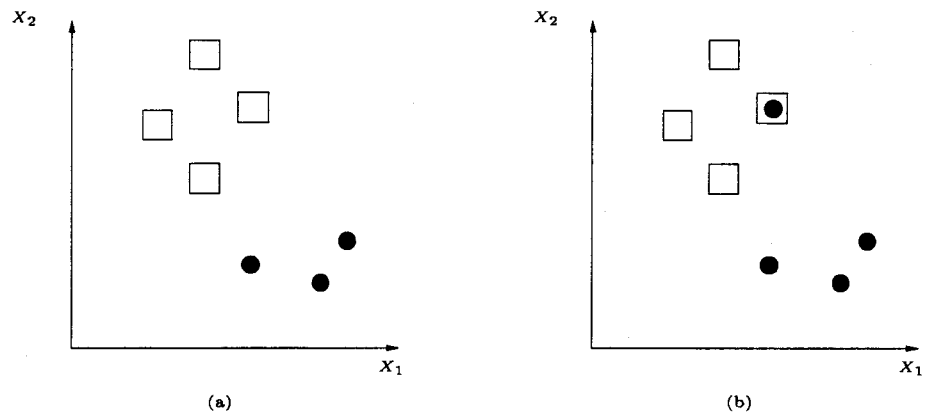

FIG. 1.7: *Exemples de données cohérentes (a) et de données incohérentes (b)*

tableau 1.6. La construction de ce tableau nécessite l'estimation des probabilités conjointes p_{ij} d'obtention des modalités α_i et β_j par les fréquences relatives ($i = 1, \ldots, n$; $j = 1, \ldots, m$).

M_X \ M_Y	β_1	$\ldots$	β_j	$\ldots$	β_m	
α_1						
$\vdots$			$\vdots$			
α_i		$\ldots$	p_{ij}	$\ldots$		$p_{i.}$
$\vdots$						
α_n						
			$p_{.j}$			

TAB. 1.6: *Tableau de contingence de X et Y*

où :

- $P_{IJ} = \{p_{ij} | i \in I, j \in J\}$;
- $I = \{1, 2, \ldots, n\}$; $n = \text{card}(M_X)$
  $J = \{1, 2, \ldots, m\}$; $m = \text{card}(M_Y)$;
- p_{ij} est la probabilité d'occurence conjointe des modalités α_i et β_j ;
- $p_{i.} = \sum_j p_{ij}$ et $p_{.j} = \sum_i p_{ij}$.

On peut également définir un tableau à partir du calcul des probabilités conditionnelles $p_{j/i}$, en divisant chaque terme de la i^e ligne du tableau de contingence par la quantité $p_{i.}$ comme le montre le tableau 1.7.

Ce dernier tableau est intéressant car il permet une meilleure représentation des cas peu observables. En effet, ce n'est pas parce que $X = \alpha_i$ n'arrive pas souvent pendant la

M_X \ M_Y	β_1	...	β_j	...	β_m
α_1					
⋮			⋮		
α_i		...	$p_{j/i}$	...	
⋮					
α_n					

$p_{j/i} = p_{ij}/p_{i.}$ est la probabilité d'obtenir $Y = \beta_j$ sachant que $X = \alpha_i$.

TAB. 1.7: *Tableau des probabilités conditionnelles*

phase d'observation du système qu'il faut lui conférer une importance moindre. C'est dans ce sens que la division par $p_{i.}$ rééquilibre les pondérations relatives des lignes.

Ce tableau nous permet de générer des relations directes ($Y = f(X)$) représentant des relations de cause à effet. Néanmoins, il est également possible de transposer le tableau des probabilités conditionnelles, c'est-à-dire remplacer X par Y (et inversement), afin de générer des relations de type rétrograde ($X = g(Y)$). Dans le cadre de la surveillance d'un processus, cela nous permettrait de prendre en compte des pannes éphémères ($p_{.j}$ faible). En effet, afin d'apprendre les relations entre les variables, l'expert du système a tout intérêt à ce que toutes les pannes possibles de son système soient représentées dans la base de données initiale. Or, l'expert hésitera à mettre en panne son système sous peine de ne plus répondre au cahier des charges fixé (rentabilité, ...) ; ou tout au moins, il laissera son système en panne le moins longtemps possible, et le système cognitif devra se contenter de ces quelques instants de pannes pour apprendre. C'est dans ce cadre de travail que ce tableau peut s'avérer très intéressant.

1.5 Conclusion

Ce chapitre nous permet de fixer le cadre de notre travail, et *a fortiori* le vocabulaire utilisé. C'est ainsi que le sens du mot « système » ne sera pas le même selon que l'on se place dans le cadre de l'apprentissage ou que l'on se place dans le cadre de l'automatique. De même, la notion de système dynamique est prépondérante en automatique.

Le choix des variables utilisées dans notre étude a également été discuté.

Ce vocabulaire étant fixé, nous avons présenté la notion d'incohérence des données d'apprentissage, ce qui nous permettra d'introduire, dans le deuxième chapitre, les problèmes de l'apprentissage et de l'analyse structurale. Nous pourrons dès lors représenter la connaissance que l'on cherchera à extraire de l'observation du système.

Chapitre 2

Utilisation des méthodes de l'apprentissage dans le cadre de l'analyse structurale d'un système physique

L'objectif de ce chapitre est de montrer que les méthodes issues de l'apprentissage peuvent être utilisées à des fins :

- de visualisation
- de structuration
- d'explication
- et de prédiction

du comportement d'un système physique plus ou moins complexe.

Dans ce sens, nous définirons dans un premier temps les notions de « modèle », de « connaissance », ainsi que la notion de « prédiction ».

Dans un deuxième temps, nous présenterons les différents problèmes que peuvent traiter les méthodes d'apprentissage automatique. Nous définirons dès lors l'apprentissage supervisé, qui nous intéresse tout particulièrement dans ce travail.

Les problèmes de l'analyse structurale seront ensuite exposés, à savoir le problème de visualisation (où se trouve l'information pertinente pour l'étude de mon système?), de structuration (décomposition d'un système complexe en un ensemble de sous-systèmes moins complexes), et d'explication (existe-t-il des relations liant certaines variables entre elles?). Dans ce cadre, un parallélisme avec les méthodes issues de l'apprentissage sera établi. Ce parallélisme nous conduira tout naturellement à représenter la connaissance obtenue sur le système par des *représentations propositionnelles.*

Afin de résoudre les problèmes de l'apprentissage ainsi que les problèmes de l'analyse structurale, nous proposerons d'appréhender les différentes démarches utilisées à l'heure actuelle, à savoir l'approche basée sur les données (ou *approche ascendante*), et l'approche basée plutôt sur la construction d'un modèle préétabli à base de règles (ou *approche descendante*).

Cette démarche générale étant établie, nous présenterons les algorithmes utilisés dans le contexte de l'apprentissage pouvant être utilisés dans le contexte de l'analyse structurale des systèmes complexes, à savoir :

- les algorithmes d'apprentissage symbolique (les arbres de décision, les systèmes à base de règles) ;
- les algorithmes à base d'instances (*IBL*, *K**) ;
- les réseaux de neurones.

Cette présentation des méthodes ne prétend pas à l'exhaustivité.

Les modèles obtenus dans le cadre de l'analyse structurale d'un système complexe, comme dans le cadre de l'apprentissage, peuvent être nombreux. L'utilisateur devra alors mesurer la qualité de chaque modèle obtenu, afin de ne garder que le meilleur. Cette qualité est donc une mesure déterminée *a posteriori* qui permettra également de mesurer la validité de prédiction de l'état des variables Y sachant la relation $Y = f(X)$ et ayant mesuré (ou observé) X.

2.1 De la connaissance (modèle) à la prédiction

La base de données étant établie, on cherchera à apprendre comment le système réagit, c'est-à-dire quelles sont ses caractéristiques principales.

Cet apprentissage du système physique passe nécessairement par une phase de « simplification », donnant lieu à une **connaissance** pour la communauté traitant de l'apprentissage, encore appelée **modèle** [HS94] dans le cadre de l'automatique.

Cette connaissance constitue en fait un niveau hiérarchique supérieur de l'information (figure 2.1) que l'on peut voir comme un résumé de l'information contenue dans la base de données initiales.

L'objectif du modèle associé à un système physique est de prédire l'état de ce dernier (à des fins de surveillance et/ou de contrôle). Cette **prédiction** pourra être établie sur la base du modèle et à partir d'un ensemble minimal d'observations que l'expert a en sa possession (figure 2.2). Cet ensemble minimal d'observations représente un échantillon de l'ensemble des « observations possibles » (encore appelées « réelles » en apprentissage). Pour un automaticien, la notion de prédiction de l'état d'un système physique est une notion temporelle (due à la dynamique du système). À partir de l'observation à plus ou moins court terme du système, il cherchera donc à obtenir l'état du système (qui évolue

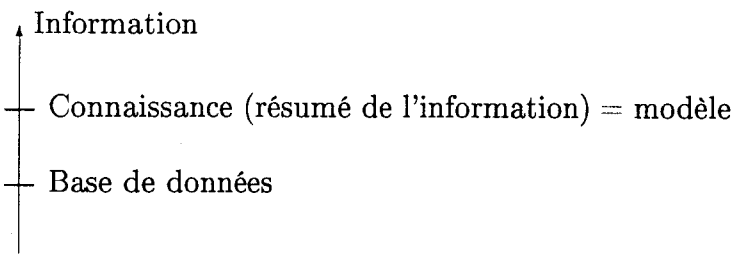

FIG. 2.1: *Niveaux de connaissance des systèmes*

dans le temps) dans un futur plus ou moins proche. La notion de prédiction en temps réel est bien entendu liée à la notion de *temps de réponse* du système. Il est important de souligner que cette notion temporelle n'est pas forcément explicite dans les problèmes que rencontre le cogniticien.

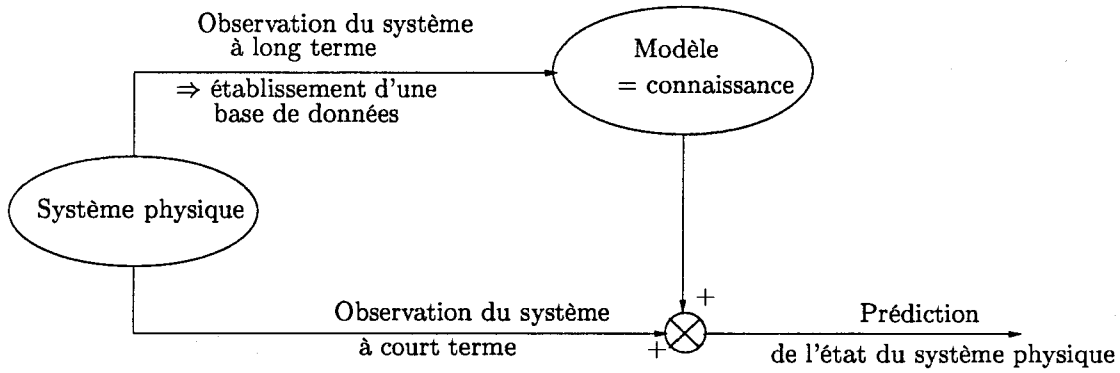

FIG. 2.2: *Système physique et modèle*

De cette notion de temps réel découle également la notion de simplicité du modèle. En effet, plus un modèle est simple, et plus il pourra être utilisé en temps réel, car moins les calculs pour prédire l'état du système sont élaborés. *A contrario*, un modèle complexe ne sera utilisable que dans une moindre mesure en temps réel. Dans cette optique, il est clair qu'un compromis sur la complexité doit être trouvé, en prenant en compte le fait que « *Tout ce qui est simple est faux, tout ce qui est compliqué est inutilisable* » (PAUL VALÉRY).

Dans ce sens, un modèle complexe est en général trop lourd à manipuler (d'un point de vue interprétabilité et d'un point de vue calculatoire dans le cadre du diagnostic), et un modèle simple repose en général sur des hypothèses simplificatrices trop fortes dans la réalité.

Un compromis entre la simplicité d'un modèle et son efficacité doit alors être trouvé. Ce compromis dépend bien entendu du type de système physique étudié ainsi que des objectifs à atteindre.

Ce problème sera discuté dans le paragraphe traitant de la qualité du modèle (paragraphe 2.6).

2.2 Les problèmes de l'apprentissage

Avant de nous placer dans le cadre de l'analyse structurale d'un système physique (paragraphe 2.3), nous présenterons le vocabulaire utilisé dans le domaine de l'apprentissage, ainsi que les principales familles de méthodes développées, ce qui nous permettra d'établir un lien entre les deux domaines.

Inférence

L'extraction d'information n'est pas une simple recopie de l'information contenue dans la base de données. Elle peut plutôt être inférée à partir de celle-ci et on distingue deux techniques d'inférence :

- La **déduction** est une technique d'inférence qui est une conséquence logique de l'information contenue dans la base. La plupart des systèmes de gestion de base de données offrent des opérateurs de déduction d'information. Par exemple, si l'on dispose de deux tables relationnelles, l'une représentant la relation entre les employés d'une entreprise et les départements, l'autre, la relation entre les départements et les directeurs, on peut inférer par un opérateur de déduction une relation entre les employés et les directeurs. Dans le cadre de l'étude d'un système physique, s'il existe des relations entre les variables d'entrée et les variables d'état d'une part, et des relations entre les variables d'état et de sortie d'autre part, alors il est possible d'inférer des relations entre les variables de sortie et d'entrée.

  Cette technique nécessite une certaine cohérence de la base de données. Dans le cas contraire, les résultats obtenus pourront être complètement contradictoires.

- L'**induction** est une technique d'inférence qui est une généralisation de l'information contenue dans la base. Par exemple, à partir des deux tables précédentes, on pourrait inférer que chaque employé a un directeur (ou que chaque entrée est en relation avec une sortie). On obtient donc un niveau d'information ou de connaissance plus élevé (figure 2.1). On recherche dans la base de données des régularités, c'est-à-dire qu'on recherche certaines combinaisons de valeurs pour certains attributs. D'une certaine façon, cette régularité est un résumé de haut niveau de l'information contenue dans la base de données. On peut également représenter une telle régularité comme une règle de prédiction des valeurs d'un attribut en fonction d'autres attributs. On cherchera par exemple à trouver la (les) relation(s) liant l'état d'une sortie en fonction de l'état des entrées.

La plus importante différence entre la déduction et l'induction est que la première conduit à des affirmations évidemment *correctes*[7] sur le domaine constitué des observations possibles (« *réelles* » en apprentissage), pourvu que la base de données soit *cohérente*; tandis que la seconde approche conduit à des affirmations vérifiées (à quelques exceptions près) par l'ensemble d'apprentissage, mais pas forcément par les données réelles.

Le processus d'induction nécessitera par conséquent une phase de sélection des règles et régularités les plus plausibles, supportées par la base d'apprentissage.

Apprentissage inductif (inductive learning)

Dans le cadre de la recherche d'un modèle d'un système physique, l'objectif est d'obtenir un modèle de haut niveau capable de prédire correctement la classe d'un individu qui n'a pas forcément été observé. L'approche inductive sera dans ce cas conseillée.

Fouille de données (Data Mining)

Cette induction est appelée fouille de données (data mining) lorsque l'ensemble d'apprentissage est une base de données, de taille généralement importante. De telles données ont tendance à être bruitées (d'où la notion d'incohérence) et certaines valeurs peuvent être manquantes.

Apprentissage automatique (machine learning)

Dans la littérature, une distinction est faite entre le *data mining* et l'*apprentissage automatique* (figure 2.3). En effet, un système cognitif n'apprend pas un modèle directement à partir de la base de données, mais à partir d'un recodage de cette base formant ce qu'on avait appelé un ensemble d'apprentissage (training set). Cette automatisation du processus d'apprentissage inductif représente l'un des domaines de recherche les plus dynamiques de l'*intelligence artificielle.*

On peut dès lors distinguer deux techniques d'apprentissage:

- l'apprentissage supervisé;
- et l'apprentissage non supervisé.

Apprentissage supervisé

En apprentissage supervisé, un « oracle » externe définit les classes et fournit au système cognitif des exemples de chaque classe; un exemple (relatif à une classe) étant défini

7. Une description correcte signifie qu'aucun contre-exemple n'est reconnu par la description. Cette notion implique la notion de cohérence de la population d'apprentissage (voir chapitre 1). En revanche, si la population d'apprentissage n'est pas cohérente, alors la description ne pourra pas être correcte.

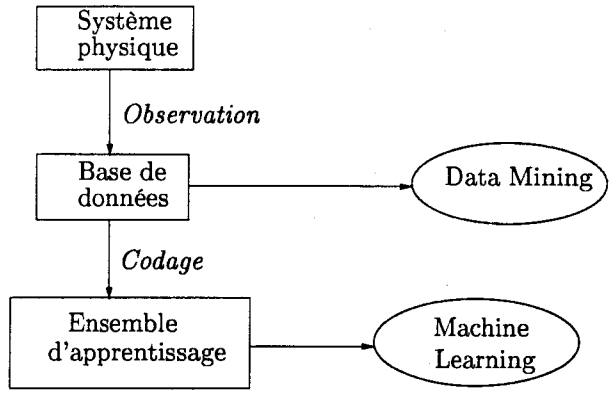

FIG. 2.3: *Quelques petites nuances de termes employés*

comme un individu appartenant à une classe. À titre d'exemple, certaines valeurs d'attributs impliqueraient une valeur particulière d'un attribut particulier; cette valeur constituant ainsi une classe particulière.

Dans les problèmes de modélisation, les classes pourront correspondre aux différents états des sorties du système; dans le cas d'un diagnostic, les classes représenteront les états de fonctionnement de celui-ci (voir chapitre 1).

Le système cognitif se propose alors de découvrir les propriétés communes de chaque classe, appelées « description » de la classe. Cette technique est aussi connue sous le nom **d'apprentissage à partir d'exemples**. Une classe munie de sa description forme une règle de classification « si *description* alors *classe* » qui peut être utilisée pour prédire la classe d'un individu non observé (*cf* tableau 2.1).

Apprentissage non supervisé

En apprentissage non supervisé, le système cognitif doit découvrir lui-même les classes, basées sur les propriétés communes des individus. Cette technique est aussi appelée **apprentissage à partir d'observations.**

Dans le cadre de l'étude d'un système physique, cette approche n'est que rarement utilisée car on connaît les classes *a priori*, c'est-à-dire l'état de la sortie ou le mode de fonctionnement se rattachant aux observations [Dub90].

Recherche itérative

Le système cognitif se base sur un ensemble d'apprentissage choisi de manière à caractériser aussi justement que possible le comportement du système physique. Ce système cognitif peut interagir avec le système physique en demandant de nouveaux exemples de manière à examiner le comportement du système physique sous certaines conditions. On parle alors de recherche itérative. Dans le cadre de l'automatique, on parlera de recherche « *en ligne* » ou de méthodes *adaptatives* dans le sens où on adapte le modèle pour que

celui-ci suive au plus juste les données observées. On peut également envisager de demander l'observation de nouvelles variables afin d'améliorer le modèle existant.

Les applications

Nous trouvons des applications de ces méthodes d'apprentissage dans les domaines suivants :

- analyses financières (attribution de prêts, prévision de marchés, ...) et marketing (construction d'un profil de clients potentiels, ...) ;
- médecine (aide au diagnostic, reconnaissance tumorale, ...) ;
- reconnaissance des formes (lecture automatique, ...) ;
- contrôle de processus (surveillance, diagnostic, ...).

2.3 Les problèmes de l'analyse structurale

Dans le cadre d'une analyse structurale d'un système, la construction d'un modèle à partir des observations fait apparaître trois types de problèmes bien distincts [BMDD+90] :

- un problème de visualisation ;
- un problème de structuration ;
- un problème d'explication sur lequel vient se greffer un problème de prédiction (ce problème sera traité dans le paragraphe 2.6).

2.3.1 Visualisation

Les tableaux de données recueillies sur le système peuvent contenir plusieurs milliers d'observations effectuées sur plusieurs dizaines de variables. Le praticien se trouve alors en présence d'une masse de données difficilement interprétable. En effet, le nuage des observations décrit par deux variables est tout à fait visualisable dans le plan. Il est déjà plus difficile, pour l'homme, de se représenter un nuage d'observations décrit par trois variables (représentation dans l'espace). Pour quatre variables, et plus, une étude visuelle des données s'avère impossible.

On cherche alors une représentation de ce nuage dans un espace de dimension plus petite.

Le problème de visualisation du nuage des observations, lorsque le nombre de variables prises en compte est important, consiste donc à le diminuer. On cherche dès lors les variables les plus importantes, les plus significatives pour notre étude. En d'autres termes, ce problème revient à trouver les variables contenant le plus d'information possible.

Deux façons d'opérer sont alors possibles :

- On utilise les méthodes factorielles [Fou85] de l'Analyse des Données (problème de recherche de facteurs principaux posé par THURSTONE dans les années 1930), qui consistent à trouver des combinaisons en général linéaires des variables initiales. Celles-ci sont cherchées de façon à ce que les nouvelles variables (encore appelées facteurs principaux) contiennent un maximum d'information.

  En général, cela n'est pas possible sans « distorsion » des mesures (« distorsion » que l'on cherche à minimiser).

  Les deux principales méthodes d'analyse factorielle sont l'Analyse en Composantes Principales [Hot33] et l'Analyse Factorielle des Correspondances [Ben80, Cib87] qui se distinguent l'une de l'autre par le type de variables qu'elles traitent, et donc, par le choix de la métrique utilisée (généralement une distance euclidienne pour la première et la distance du χ^2 pour la deuxième méthode)[8].

- On recherche directement les variables apportant la plus grande information, parmi les variables initiales prises en compte. Dans ce sens, on ne prendra en compte que les p ($p < N$; N étant le nombre de variables considérées) variables les plus informatives. Comme précédemment, la distorsion des mesures qui en résulte est due à la diminution des dimensions de l'espace de représentation.

  À titre d'exemple, une variable prenant toujours la même valeur lors de la phase d'observation du système sera considérée comme n'apportant aucune information significative. Cela ne veut pas dire qu'elle prendra toujours cette même valeur.

  De nombreuses méthodes statistiques existent (maximisation de l'inertie pour des variables quantitatives, du χ^2 pour des variables qualitatives, ...) dans une littérature très abondante. Pour notre part, nous nous intéresserons à des méthodes entropiques (présentées au chapitre 4) basées sur la Théorie de l'Information (présentée au chapitre 3).

2.3.2 Structuration

« *Un système complexe est un ensemble de sous-systèmes (plus simples) pouvant interagir entre eux* » [Del71]. Il serait dès lors intéressant de décomposer ce système en sous-systèmes faiblement couplés. C'est donc un problème de structuration (de décomposition, ou encore de **classification automatique non supervisée**), qui consiste à munir la population d'apprentissage, ou l'ensemble des variables, d'une structure particulière. Diverses techniques de décomposition sont mises en place, suivant la structure désirée.

Le but de ces *méthodes de classification* [Ler70, JL78, CDG+89] (encore appelée *taxonomie*, ou *taxinomie*) est de regrouper les individus en un nombre restreint de classes homogènes. Ces classes sont obtenues au moyen d'algorithmes formalisés, et non pas par

8. Une *analyse factorielle des correspondances* pourra également être vue comme une méthode d'explication (voir le paragraphe 2.3.3) dans le cas où X et Y sont des variables qualitatives.

des méthodes subjectives ou visuelles faisant appel à l'initiative du praticien, d'où le terme « automatique ».

Des méthodes exhaustives sont rarement utilisées, car une explosion combinatoire du nombre de solutions est dans ce cas souvent obtenue. Une utilisation en temps réel passe nécessairement par l'utilisation de gros calculateurs ou des systèmes multi-processeurs mettant en jeu la notion de *calcul distribué* (ou de *parallélisme*) [Fos95]. Cette approche sort du cadre de notre travail.

Les méthodes les plus usitées sont des méthodes heuristiques : ce sont « des algorithmes dont on considère qu'ils sont suffisamment raisonnables pour donner des résultats satisfaisants » [Rou85].

Nous distinguons deux types de méthodes heuristiques :

- les méthodes de classification hiérarchique :
  - ascendante (agrégations successives d'individus, puis de groupes) ;
  - descendante (dichotomies successives)[BS89] ;
  - algorithme de transfert (succession de dichotomies et d'agrégations, et inversement) [Tor82]. Une solution localement optimale est obtenue dans ce cas.
- les méthodes métriques (ou de coalescence). Ce sont des méthodes d'élaboration directe d'une partition :
  - algorithme de regroupement autour des centres mobiles [For65] ;
  - algorithme de J.B. MAC QUEEN [Mac67] ;
  - algorithme des nuées dynamiques [Did72].

Il existe également des méthodes de classification qui utilisent des approches statistiques, par exemple la méthode des Noyaux de ROSENBLATT-PARZEN (méthode de détection des zones à forte densité dans l'espace des observations) [Par60, Pos87].

À titre d'exemple explicatif, la figure 2.4 met en évidence les trois sous-systèmes (X_1, X_2, X_3), (X_4) et (X_5, X_6) à partir du système initial $(X_1, \ldots, X_6)$.

Le tableau 2.4 représente par exemple la valeur d'un indice de couplage (blanc : valeur $0 \equiv$ pas de couplage ; gris : valeur $1 \equiv$ il existe un couplage entre les variables).

Il est clair qu'une structuration aussi tranchée est plutôt rare. Dans le cas contraire, on pourra faire appel à des méthodes permettant de définir des fonctions d'appartenance (à un sous-système) floues. Cette approche sort du cadre de notre travail.

2.3.3 Explication

L'objectif des méthodes d'explication est la recherche des relations de dépendance entre variables. Nous distinguons dès lors d'une part, la (ou les) variable(s) à expliquer, et

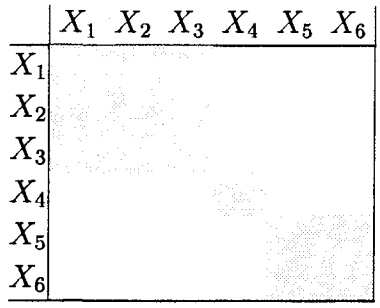

FIG. 2.4: *Exemple de structuration*

d'autre part, la (ou les) variable(s) explicative(s), ou considérée(s) comme telle(s). Nous reprendrons les notations introduites au chapitre 1.

Le problème d'explication consiste à rechercher des procédures d'affectation d'une observation quelconque : étant connues les modalités de certaines variables (variables explicatives), il s'agit d'expliquer (de prédire) les modalités des autres variables (variables à expliquer). On cherche donc à caractériser des relations de dépendance entre les premières et les secondes. Certains auteurs parlent de classement (en Reconnaissance des Formes), de **classification supervisée** (dans le sens où on connaît des représentants des diverses classes en présence), d'apprentissage à partir d'exemples, d'induction (en Intelligence Artificielle), ou de modélisation.

Cette procédure dichotomique pourra être renouvelée sur toute partition de Σ en deux classes distinctes. En effet, nous pouvons procéder de deux façons différentes :

- Y est fixée, et on cherchera les variables les plus explicatives de Y, suivant un critère donné. C'est toujours le cas en diagnostic, car Y représente les différents états de fonctionnement du système étudié.

- Y n'est pas fixée : pour chaque variable, on se posera la question de savoir si on peut l'expliquer à partir d'autres variables, et si oui, lesquelles. Dans le cas de la modélisation d'un processus physique, la condition de causalité (chère aux automaticiens, et qui peut se traduire par « *la réaction ne peut pas être en avance sur l'action* ») est alors à prendre en compte. À titre d'exemple, si une relation lie une entrée à une sortie du système, alors la sortie ne pourra jamais être en avance sur l'entrée.

Notre travail se place dans le cadre de l'*explication* (de l'*apprentissage supervisé*). Nous nous proposons alors de formaliser fonctionnellement ce problème [Sta84] :

Soient X et Y deux variables équivalentes. Par définition de la relation d'équivalence, il existe une relation bijective :

$$\begin{array}{rccc} f: & M_X & \longrightarrow & M_Y \\ & \alpha & \longmapsto & f(\alpha) \end{array} \qquad \begin{array}{rccc} f^{-1}: & M_Y & \longrightarrow & M_X \\ & \beta & \longmapsto & f^{-1}(\beta) \end{array}$$

En d'autres termes, chaque variable X et Y est un recodage univoque de l'autre, et les relations ci-dessus constituent un modèle du système $\{X, Y\}$, sous la forme d'équivalences :

$$X = \alpha \Longleftrightarrow Y = f(\alpha) \quad \forall \alpha \in M_X$$
$$X = f^{-1}(\beta) \Longleftrightarrow Y = \beta \quad \forall \beta \in M_Y$$

Remarques :

- Le tableau de contingence défini au chapitre 1 est alors un tableau d'affectation, où à une modalité de la variable X correspond une et une seule modalité de la variable Y, et réciproquement. Il y a une et une seule probabilité conjointe non nulle par ligne et par colonne.

  Considérons maintenant deux variables X et Y telles que $X \preccurlyeq Y$. Dans ce cas, nous pouvons montrer que :

  - il existe une relation $Y = f(X)$:

$$\begin{array}{rccc} f: & M_X & \longrightarrow & M_Y \\ & \alpha & \longmapsto & f(\alpha) \end{array}$$

    telle que les implications suivantes décrivent le système $\{X, Y\}$:

$$X = \alpha \Longrightarrow Y = f(\alpha) \quad \forall \alpha \in M_X$$

  - chaque ligne du tableau de contingence ne contient qu'une seule probabilité non nulle.
  - il existe une variable $\tilde{X}$ plus grosse que X et équivalente à Y (figure 2.5).

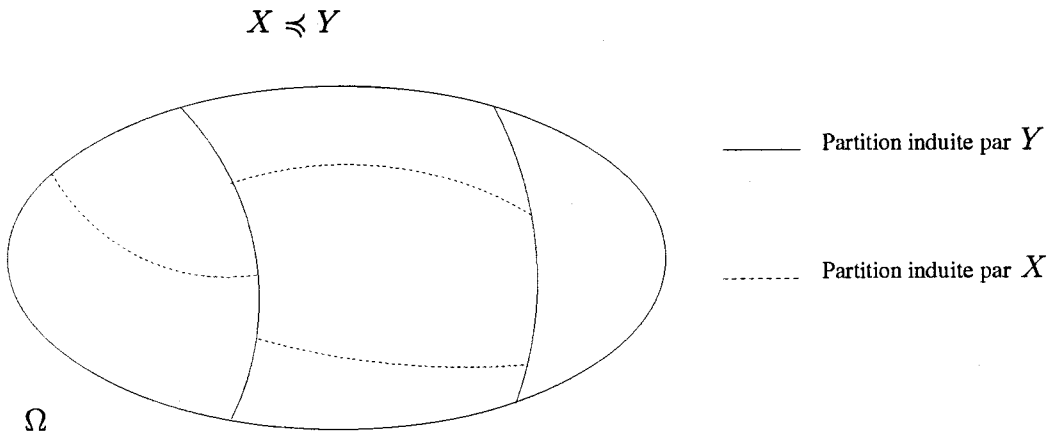

FIG. 2.5: *Partitions de la population d'apprentissage induites respectivement par X et par Y*

- Pratiquement, à la seule vue du tableau de contingence, si il existe une seule probabilité non nulle par ligne, alors :

$$[\tilde{X} \succcurlyeq X \text{ et } \tilde{X} \equiv Y] \iff X \preccurlyeq Y \iff Y = f(X).$$

Nous dirons alors que les données d'apprentissage sont cohérentes. X et Y sont totalement liées : il existe une relation déterministe entre ces deux variables.

- Dans le cas contraire (plus d'une case non nulle par ligne), les données d'apprentissage sont incohérentes. Cela signifie que plusieurs observations, caractérisées par la même modalité de X, présentent des modalités différentes de Y : on ne peut alors pas expliquer complètement Y par X. Une étude montrant l'influence du nombre de modalités de X a été entreprise dans [Sba93].

  Deux conclusions sont alors envisageables :

  - l'absence d'explication à tort : le modèle déterministe $Y = f(X)$ est représentatif du système, mais les données relevées sont biaisées (bruit, mauvais fonctionnement d'un sous-système) ;
  - l'absence d'explication à raison : certaines variables pertinentes pour l'étude de notre système ont été omises.

Le problème d'explication peut donc se poser en ces termes : connaissant une partition $P_Y(\Omega)$ de la population d'apprentissage induite par la variable à expliquer Y, il s'agit de trouver une partition $P_S(\Omega)$ de Ω induite par S (où S est un vecteur de variables appartenant à $\Sigma \backslash Y$) telle qu'elle soit la plus proche de $P_Y(\Omega)$, au sens d'un critère donné (figure 2.6).

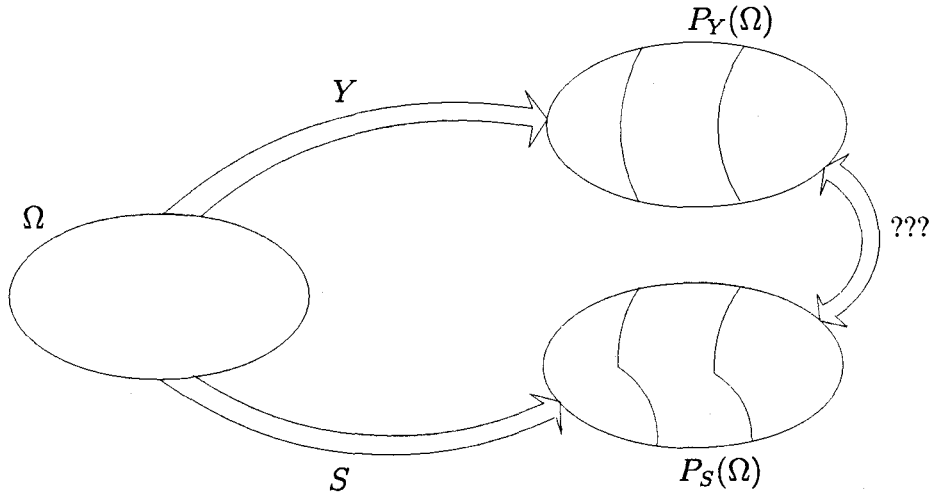

FIG. 2.6: *Problème d'explication*

Tout au long de ce travail, nous chercherons à caractériser des relations liant Y à $S \subset \tilde{X}$, où $\tilde{X}$ est une variable plus grosse que X.

Dans le cadre de la recherche de relations entre paramètres, les méthodes d'apprentissage sont très nombreuses, et on peut en général distinguer deux groupes de méthodes :

- les méthodes probabilistes, ou statistiques, qui nécessitent une hypothèse de distribution de probabilités pour chaque classe en présence (par exemple, hypothèse gaussienne, ...) ;
- les méthodes issues de l'intelligence artificielle, qui sont en général des méthodes non paramétriques. Ces méthodes peuvent, elles-mêmes, être classifiées en deux groupes :
  - les méthodes symboliques, pour lesquelles la procédure de classification peut se mettre sous la forme d'un ensemble de règles. À titre d'exemple, les méthodes permettant de construire des arbres de décision font partie de cet ensemble ;
  - les méthodes non symboliques (ou adaptatives) pour lesquelles la procédure de classification est de type « boîte noire ». À titre d'exemples, nous pouvons citer les réseaux neuronaux ainsi que les algorithmes génétiques. L'inconvénient majeur de ces méthodes est qu'elles ne donnent pas un modèle explicite d'explication.

Dans le cadre de ce travail, nous développerons des méthodes d'explication probabilistes sans modèle *a priori* et basées sur des critères issus de la théorie de l'information (voir chapitre 3), ce qui nous permettra de traiter tous les types de données sans distinction (numériques, qualitatives, ordonnées ou nominales).

2.4 Représentation de la connaissance

Une démarche courante consiste à extraire de la connaissance à partir de la base de données. Cette connaissance représente un résumé de l'information (modèle) contenue dans la base de données et constitue un niveau supérieur de représentation de l'information.

Cette modélisation pourra s'effectuer de deux façons différentes :

- en utilisant toute la base de données dont on dispose. Une fois le modèle obtenu, on pourra le valider, soit en utilisant un sous-ensemble de la base de données (tiré par exemple de façon aléatoire[9]), soit en utilisant une autre base de données issue du même système physique (voir paragraphe 2.6).
- en utilisant une partie de la base de données. La validation du modèle s'effectuera alors avec la partie non utilisée de la base de données (voir paragraphe 2.6) lors de la phase d'apprentissage.

9. Dans le cas d'une étude d'un système dynamique, le tirage aléatoire n'a pas de sens car la dynamique même du système disparaît. Il nous faut dans ce cas observer le système à des instants d'échantillonnage consécutifs.

Il existe différentes représentations possibles de la connaissance dont les plus usitées sont les *représentations propositionnelles*. Elles utilisent une formule logique, constituée de conditions sur les valeurs des attributs. Ces représentations sont décrites en général sous la forme de conjonctions de *clauses* où les clauses sont des disjonctions de conditions sur les valeurs des attributs. Il peut plus rarement s'agir de disjonctions de termes représentant des conjonctions de conditions sur les valeurs des attributs.

Afin d'illustrer ce propos, présentons sommairement les arbres de décision, les règles de production, ainsi que les listes de décision.

Les arbres de décision

Un arbre de décision est une représentation très utilisée en apprentissage automatique supervisé. Les nœuds de l'arbre correspondent aux attributs testés. Les branches sont assignées aux différentes valeurs possibles de l'attribut [10], et les feuilles correspondent aux différentes classes. Une observation est classifiée en parcourant l'arbre et en empruntant les chemins correspondant aux valeurs des attributs de cette observation.

Les règles de production

On peut également utiliser des règles de production. Il est d'ailleurs possible de passer d'un arbre de décision à un ensemble de règles de production.

Les règles de production sont largement utilisées pour la représentation de la connaissance dans les systèmes experts, et peuvent facilement être interprétées par l'être humain.

Les listes de décision

Une dernière représentation propositionnelle possible est la liste de décision. C'est une généralisation des deux dernières représentations.

Une liste de décision est en fait une liste de couples

$$(\phi_1, C_1), (\phi_2, C_2), \cdots, (\phi_r, Cr)$$

où ϕ_i est une description élémentaire, et C_i une classe. La dernière description ϕ_r est la constante « vraie ». La classe d'un objet o est C_j si j est le plus petit index d'une description ϕ_j qui couvre cet objet. Un tel index existe toujours puisque le dernier terme est toujours vrai. C'est la classe par défaut.

À ces représentations est souvent associée l'idée qu'à une observation correspondra une et une seule conclusion possible sur la base du modèle obtenu. À titre d'exemple, un système sera dans un mode de fonctionnement donné à l'instant « t », et ne pourra par conséquent pas être dans un autre état (les états sont exclusifs).

10. Si l'attribut est quantitatif, le nœud consistera à tester si la valeur réelle de l'attribut est supérieure ou inférieure à un seuil défini par l'algorithme si l'arbre est binaire.

Dans certains cas, ces classes ne seront pas forcément disjointes et la notion d'*appartenance floue* à une classe prend tout son sens. Cette notion est d'ailleurs renforcée par le fait que la population d'apprentissage peut être très incohérente, et par conséquent un apprentissage parfait n'a pas de sens dans ce contexte.

Une autre démarche est depuis peu de temps menée sur la notion d'apprentissage Pac (Presque Apprenable Correctement) [Val84, DDG96, DG97, DGS97].

Ces deux dernières démarches (*apprentissage Pac* et *règles floues*) sortent de notre cadre de travail, mais il serait très intéressant de comparer l'approche développée ici avec ces dernières.

2.5 Recherche d'explication

Afin de résoudre ce problème d'explication, différents algorithmes issus des sciences cognitives sont utilisés pour construire un modèle de comportement du système étudié, et ainsi représenter la connaissance issue des données contenues dans la base de données initiales.

Dans un premier temps, nous présenterons les deux types d'approches, à savoir les approches ascendantes et descendantes.

Puis, partant d'une description donnée, plusieurs stratégies de recherche peuvent être appliquées afin d'éviter l'aspect combinatoire de la démarche exhaustive. Nous les présenterons succinctement.

Enfin, nous présenterons les algorithmes communément utilisés dans la littérature, à savoir les algorithmes d'apprentissage symbolique, les algorithmes d'apprentissage à base d'instances, et les réseaux de neurones.

2.5.1 Les approches ascendantes et descendantes

L'idée de base est de partir d'une description initiale et de la modifier itérativement jusqu'à ce que sa « qualité » atteigne un seuil prédéfini.

Il existe deux approches qui diffèrent principalement dans le choix de la description initiale, et dans la manière de la modifier afin d'améliorer sa qualité [HS94] :

- l'approche ascendante (data-driven) ;
- l'approche descendante (model-driven).

L'approche ascendante (data-driven)

Les concepts sont appris un à la fois : pour une classe considérée, la description initiale est simplement l'ensemble de tous les exemples. Cette description est évidemment *correcte* dans le cas d'une base de données cohérente. Mais que la base de données soit cohérente ou incohérente, la description initiale sera beaucoup trop complexe. Afin de réduire cette complexité, la description est modifiée par de successives

généralisations. On obtient une règle plus générale, qui classifie les exemples correctement ou avec une certaine tolérance dans le cas d'une base de données incohérente.

À titre d'exemple, partant de la configuration (a) pour laquelle (X_1, X_2) représentent deux variables, nous pouvons chercher à caractériser la classe « □ » . Il nous faut dès lors recouvrir chaque exemple (sous-entendu de la classe « □ ») (b). Une bonne procédure de généralisation nous amènera à la configuration (c) dans laquelle tous les exemples sont couverts par la même description (figure 2.7).

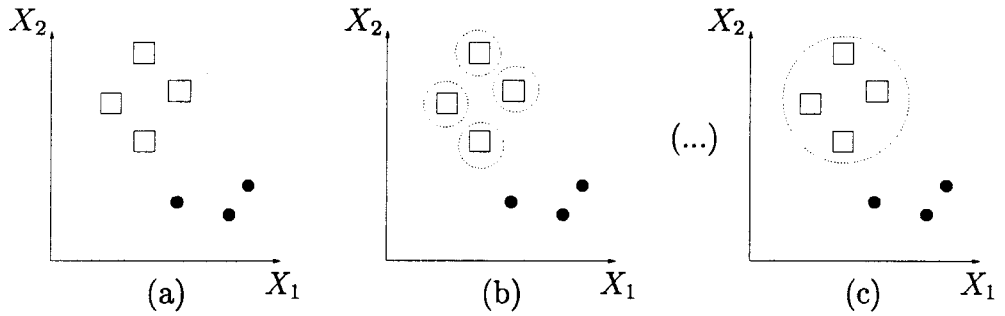

FIG. 2.7: *Exemple d'approche ascendante*

Les algorithmes AQ [MMHL86a, MMHL86b] et GENREG [Ral87, Ral88] sont des exemples d'algorithmes utilisant cette approche ascendante.

L'approche descendante (model-driven)

À l'inverse, on choisit initialement une description qui n'est pas forcément correcte mais assez générale. Cette description est ensuite transformée par une séquence de généralisations et spécialisations jusqu'à ce que sa qualité atteigne le seuil défini (figure 2.8). Les algorithmes qui construisent des arbres de décision se situent dans cette catégorie.

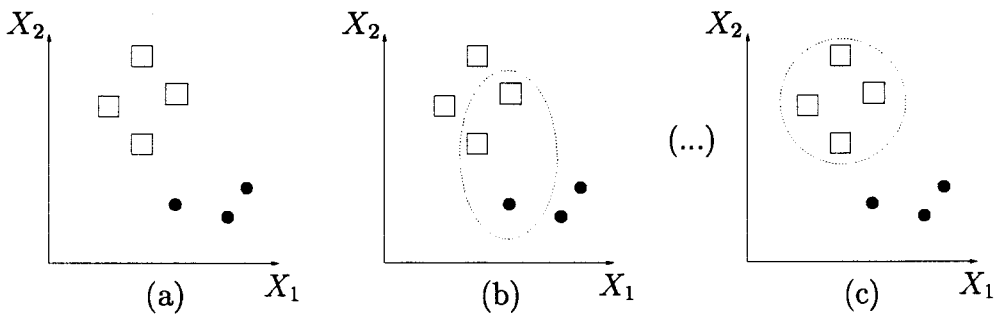

FIG. 2.8: *Exemple d'approche descendante*

2.5.2 Les stratégies de recherche

Afin de trouver la meilleure description à partir de la description initiale, la stratégie la plus simple est la *recherche exhaustive.* Mais l'espace de recherche est bien souvent trop grand et on aboutit à une explosion combinatoire du nombre de descriptions possibles, difficile à appréhender. On procède donc dans ce cas à une recherche partielle (*heuristiques*).

Cette recherche peut être représentée par un arbre où les **nœuds** sont des **descriptions** (modèles) et les branches des opérations sur ces descriptions. La description initiale est représentée par la racine de l'arbre dans lequel le système cognitif navigue afin de sélectionner une séquence d'opérations. À chaque pas, soit une seule opération est appliquée, (« *hill climber* »), soit les n meilleures sont retenues («*beam search* » c'est-à-dire plusieurs « *hill climber* » en parallèle).

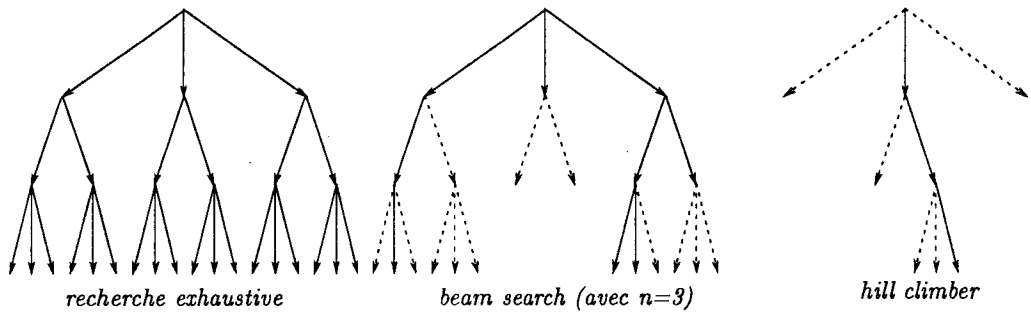

FIG. 2.9: *Stratégies de recherche*

La figure 2.9 représente les différentes stratégies de recherche des descriptions. Un nœud de l'arbre représente une description. Ainsi, la racine représente la description initiale et les branches de l'arbre représentent, quant à eux, les différentes transformations à considérer à partir de la description mère. Dans le cas de la stratégie du « *beam search* », on ne considère à chaque itération de l'algorithme de recherche, que les n (ici $n = 3$) meilleures descriptions (au sens d'un critère donné). La stratégie du « *hill climber* » consiste à ne considérer que la meilleure description à un niveau donné.

En général, l'algorithme de recherche ne choisit pas l'opération suivante au hasard, mais utilise des *heuristiques* pour sélectionner les opérations les plus susceptibles de s'approcher de la solution optimale. Une heuristique souvent utilisée est l'estimation du gain résultant de l'application des opérations considérées sur la structure actuelle. La qualité de la structure résultante peut également être calculée en utilisant la fonction de qualité ou estimée en utilisant un autre critère moins onéreux. L'algorithme peut aussi calculer la qualité de toutes les extensions possibles ou seulement celle d'un nombre restreint.

Ces algorithmes de recherche peuvent parfois présenter quelques inconvénients de convergence, en particulier si les heuristiques ne sont pas très bien adaptées, ou peuvent aboutir à un maximum local. D'autres alternatives à ces stratégies de recherche existent. Parmi celles-ci, on trouve les *algorithmes génétiques*, ou les *méthodes de recuit simulé* qui sortent de notre cadre de travail.

2.5.3 Algorithmes utilisés dans la littérature

Dans un premier temps, nous présenterons quelques algorithmes d'apprentissage symbolique. Dans un deuxième temps, nous exposerons quelques algorithmes d'apprentissage à base d'instances et enfin nous présenterons de façon succincte les techniques à base de réseaux de neurones.

2.5.3.1 Algorithmes d'apprentissage symbolique

Les algorithmes d'apprentissage symbolique sont des algorithmes qui déterminent un ensemble de règles représentant les relations entre les attributs et les classes. On trouve une multitude d'outils dans ce domaine qui se distinguent par la façon de construire ces règles. Il peut par exemple s'agir de règles du type « *prémisse* alors *conclusion* ».

Nous distinguerons pour notre part, les arbres de décision et les algorithmes à base de règles. Nous présenterons ainsi succinctement :

- le système *C4.5* qui est un système utilisant une approche descendante (model-driven) ;
- le système *AQ15* (approche ascendante - data-driven) ;
- ainsi que le système *CN2*.

Les arbres de décision *C4.5* [Qui86, Qui93] (dont le noyau algorithmique est *ID3*) représente certainement l'algorithme de construction d'arbres de décision le plus utilisé actuellement. Il est en fait largement inspiré du système CART développé par BREIMAN [BFOS84], qui constitue la référence en la matière de construction d'arbres de décision.

La construction de cet arbre se fait de manière récursive de la façon suivante : à chaque nœud, l'algorithme cherche le meilleur attribut permettant de discriminer au mieux (suivant un critère donné) la population d'apprentissage vis-à-vis des classes en présence. Ces classes pourront être prédéfinies par l'utilisateur, mais elles peuvent également représenter les modalités d'une variable qualitative. Dans le cas d'un diagnostic, les modalités représenteront les états de fonctionnement du système. Un sous-arbre est alors construit pour chaque sous-population non encore discriminée.

Afin de trouver la variable à tester à chaque nœud, cet algorithme utilise le concept de « gain d'information », qui peut être vu comme la diminution de l'incertitude de la conclusion résultant du test de l'attribut à ce nœud.

Deux conditions de sortie de cet algorithme sont considérées :

- La population d'apprentissage est complètement discriminée, dans le cas d'une population complètement cohérente. À chaque feuille de l'arbre correspond une conclusion.
- L'introduction de variable-test n'apporte aucune information supplémentaire. Le gain d'information est nul.

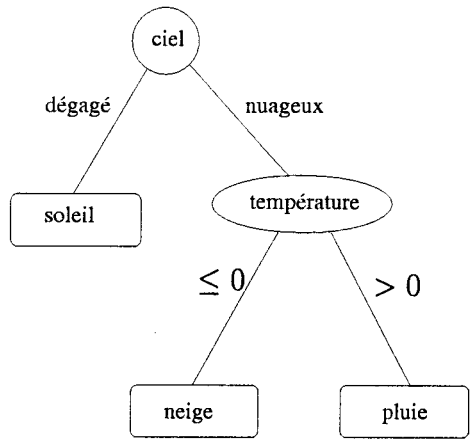

FIG. 2.10: *Un exemple simple d'arbre de décision*

La figure 2.10 représente un tel arbre.

La racine de cet arbre consiste en un test sur la variable « *ciel* » possédant deux modalités {*dégagé*; *nuageux*}. Dans le cas où le ciel est dégagé, alors la conclusion de l'arbre est « *soleil* ». Dans le cas contraire, il faut tester la variable « *température* » ; dans le cas où celle-ci est ≤ 0, alors la conclusion est « *neige* », sinon, la conclusion est « *pluie* ».

Remarques :

- Dans le cas d'un système complexe, l'arbre obtenu ne sera pas aussi simple, surtout si la population d'apprentissage est incohérente.
- On peut toujours retranscrire les arbres de décision en un ensemble de règles. À titre d'exemple, à partir de la figure 2.10, on peut construire les règles de fonctionnement suivantes :

  Si *ciel=dégagé* *alors soleil*
  Sinon *si température* ≤ 0 *alors neige*
  Sinon pluie

  L'avantage de cette représentation est qu'elle est directement compréhensible par l'humain.
- Le parcours de cet arbre consiste en un ensemble de tests sur les variables considérées menant dès lors :
  - à une conclusion (un état de fonctionnement dans le cadre d'un diagnostic) si la population d'apprentissage est cohérente ;
  - ou à un ensemble de conclusions dans le cas contraire. L'algorithme fournit la conclusion la plus probable associée à la probabilité d'erreur sur la population d'apprentissage.

La particularité des algorithmes d'apprentissage symboliques est que ceux-ci travaillent sur des partitions de l'espace de représentation. Certains de ces algorithmes, comme *C4.5*, ne testeront qu'une seule variable par nœud. Cette contrainte entraîne ainsi un raffinement de l'espace des variables en partitions de plus en plus fines et parallèlement aux axes considérés. À partir d'une frontière plus ou moins complexe quelconque séparant par exemple deux classes D_0 et D_1 (figure 2.11), on ne pourra obtenir qu'une frontière limite approximative.

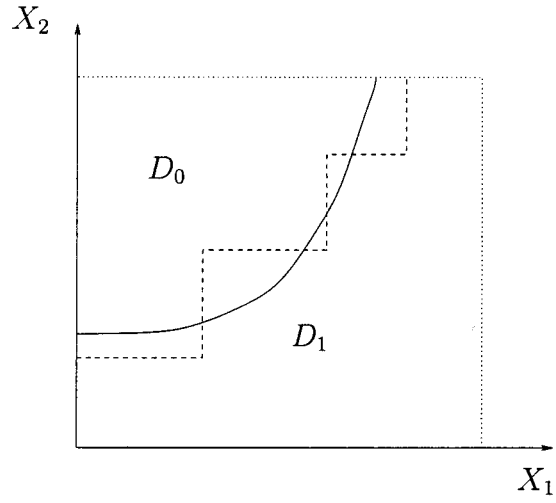

FIG. 2.11: *Exemple d'approximation d'une frontière complexe*

D'autres algorithmes peuvent effectuer une segmentation de l'espace de manière un peu moins rigide comme les réseaux de neurones ou les algorithmes d'apprentissage à base d'instances. Il existe également des systèmes qui appliquent une fonction des attributs à chaque nœud de l'arbre de décision, plutôt que d'avoir une unique valeur. C'est le cas des algorithmes proposés dans [BU95] ou de l'algorithme *OC1* proposé dans [Mur96].

Les systèmes d'apprentissage utilisant le formalisme des treillis La plupart de ces systèmes construisent complètement le treillis des concepts, à l'exception du système *LEGAL* (*Learning with Galois Lattice* [Mep94]) qui construit le sup-demi treillis en utilisant un seuil de validité et un seuil de cohérence des nœuds. Lorsque les exemples sont isolés (ce serait par exemple le cas des pannes n'arrivant que très rarement dans le cadre d'un diagnostic), cette approche n'est pas adaptée (dû au seuil de validité fixé), et il est conseillé d'utiliser *IGLUE* [NM98]. Celui-ci utilise la fonction « *entropie* » associée à une méthode descendante afin de sélectionner les concepts pertinents.

Les systèmes à base de règles

Le système *AQ* Le système *AQ15* de MICHALSKI [MMHL86a, MMHL86b] est un système d'apprentissage inductif qui génère des règles de décision, dont les prémisses sont

des formules logiques. Une caractéristique particulière à ce système est qu'il effectue une « induction constructive », c'est-à-dire qu'il utilise le domaine de connaissance pour générer de nouveaux attributs non présents dans les données initiales.

Comme de nombreux systèmes d'apprentissage, *AQ15* est initialement conçu pour construire des règles « fortes », c'est-à-dire que pour chaque classe, le système produit une règle de décision qui couvre tous les exemples positifs et ne couvre aucun exemple négatif. Le système tient compte des données incohérentes et incomplètes par pré et post traitement. Le nombre et la taille des règles produites sont diminués par une technique de post traitement appelée « rule truncating » qui n'affecte pas la précision de la classification.

Les règles sont représentées d'une manière particulière. Un *sélecteur* est une condition sur un attribut relativement à une modalité ou un ensemble de modalités, par exemple : « *couleur* = *rouge* $\vee$ *jaune* ». Une conjonction de sélecteurs est appelée *complexe.* La partie condition d'une règle de décision formée d'une disjonction de complexes, est appelée *couverture.*

Pendant la construction d'une règle de décision, *AQ* exécute une recherche heuristique dans l'espace de toutes les expressions logiques afin de déterminer celles qui couvrent tous les exemples positifs et pas les exemples négatifs. L'objectif de *AQ* est de trouver la meilleure expression au sens d'un critère que l'utilisateur définit suivant les besoins du domaine d'application.

```
Programme AQ
    Pour chaque classe Faire
        Soit COVER la couverture vide
        Tant que COVER ne couvre pas tous les exemples
        positifs Faire
            Sélectionner un noyau SEED (i.e. un exemple
            positif non couvert par COVER)
            Générer une étoile STAR (i.e. un ensemble
            de complexes qui couvrent SEED mais aucun
            exemple négatif) (voir procédure définie
            ci-dessous)
            Sélectionner le meilleur complexe BEST de
            STAR suivant le critère défini
            Ajouter le complexe BEST à la couverture
            COVER
        Fin Tant Que
    Fin Pour
Fin
```

Les principales opérations qu'effectue l'algorithme sont l'addition d'un complexe dans une couverture, et l'intersection d'un ensemble de complexes avec un ensemble de sélecteurs.

Le système génère une règle de décision pour chaque classe. À chaque pas de l'algorithme, le meilleur complexe est ajouté à la couverture. Chaque pas consiste dans un

premier temps, étant donné un exemple positif appelé *noyau*, à générer un ensemble de complexes (une *étoile*), qui couvrent le noyau et aucun exemple négatif. L'algorithme sélectionne ensuite le meilleur complexe de l'étoile suivant le critère défini par l'utilisateur. Ce complexe est ajouté à la couverture.

Le meilleur complexe est choisi de manière à maximiser le nombre d'exemples positifs couverts.

Pour réduire la taille de l'étoile jusqu'à une valeur ($MAXSTAR$) définie par l'utilisateur, des complexes sont retirés de manière à maximiser la somme des exemples positifs couverts et des exemples négatifs exclus.

```
Procédure qui génère STAR
    Soit STAR l'étoile contenant le complexe vide
    Tant que une partie de STAR couvre des exemples
    négatifs Faire
        Sélectionner un exemple négatif Eneg couvert
        par un complexe de STAR
        Spécialiser les complexes de STAR pour exclure
        Eneg :
            Soit EXTENSION l'ensemble des sélecteurs
            qui couvrent SEED mais pas Eneg
            Soit STAR l'ensemble {x ∧ y|x ∈ STAR, y ∈
            EXTENSION}
            Retirer tous les complexes déjà contenus
            dans les autres complexes de STAR
        Répéter
            Retirer les plus mauvais complexes de STAR
        Jusqu'à ce que la taille de STAR soit
        inférieure à MAXSTAR
    Fin Tant Que
Fin
```

Pour traiter les données incohérentes, AQ propose trois options. Il peut considérer ces individus incohérents comme des exemples positifs, comme des exemples négatifs, ou simplement les négliger. Si l'on dispose d'une information statistique sur la probabilité de ces exemples incohérents, ils sont préclassifiés suivant le principe du maximum de vraisemblance.

Lorsqu'il existe des exemples incorrects dans la population d'apprentissage, il peut être avantageux d'appliquer les règles construites de manière probabiliste. À côté du processus « strict », où l'on teste si une instance satisfait la description, on peut distinguer une autre manière de tester l'appartenance à une classe : un processus plus « flexible » où le degré de similarité détermine la classe.

Si on utilise un tel processus, il est possible de simplifier une description en éliminant un ou plusieurs complexes. Cette technique appelée « truncation », retire les complexes

qui couvrent le moins d'exemples. Si l'ensemble d'apprentissage est bruité, ces exemples peuvent traduire les erreurs dans les données. La règle résultante n'est plus « exacte » mais probabiliste, puisque les similarités pour les différentes classes sont comparées. Mais les règles sont plus simples et la base de connaissance est sensiblement réduite.

À chaque étape de troncature, la précision de classification est mesurée. Chaque pas produit un compromis différent entre la complexité de description et la précision de la règle.

Plusieurs expériences montrent que ces règles peuvent être tronquées sans affecter la précision de classification.

L'algorithme *CN2* Le système *CN2* de CLARK et NIBLETT [CN89] est une adaptation du système *AQ*. Comme nous l'avons vu, un inconvénient de l'algorithme *AQ* est qu'il ne s'occupe pas du bruit intrinsèquement, mais utilise un pré et post traitement. Cet algorithme dépend également beaucoup des exemples d'apprentissage spécifiques choisis pendant la recherche (les noyaux). L'objectif de *CN2* est de supprimer cette dépendance et d'incorporer une technique de traitement du bruit dans l'algorithme proprement dit. *CN2* combine en fait les meilleures caractéristiques de *ID3* et de *AQ*, car il utilise des techniques d'élagage similaires à celles utilisées dans *ID3*, et les règles conditionnelles utilisées dans *AQ*.

Le résultat obtenu avec le système *CN2* est une liste de décision, c'est-à-dire un ensemble ordonné de règles conditionnelles. Les règles sont du même type que celles obtenues dans *AQ* sauf que la partie condition est un complexe et pas une disjonction de complexes (couverture) comme dans les règles de *AQ*.

Pendant le processus de recherche, les complexes sont spécialisés en ajoutant un sélecteur conjonctif ou en retirant une valeur disjonctive dans l'un des sélecteurs. L'algorithme *CN2* génère toutes les spécialisations possibles d'un ensemble de complexes (une étoile) en effectuant l'intersection de cet ensemble avec l'ensemble de tous les sélecteurs possibles.

Le critère de qualité consiste en deux tests : un test pour déterminer l'ensemble E des exemples couverts par le complexe, ainsi que la distribution de probabilité $P = (p_1, p_2, \ldots, p_n)$ des exemples de E parmi les n classes. *CN2* utilise la mesure d'entropie (issue de la théorie de l'information que nous présenterons au chapitre 3) :

$$Entropie = -\sum_{i=1}^{n} p_i \cdot \log p_i$$

pour évaluer la qualité. En effet, plus basse est l'entropie et meilleure est cette qualité. Cette fonction favorise les complexes qui couvrent un large nombre d'exemples d'une seule classe et peu d'exemples des autres classes. Les complexes ont donc une bonne qualité sur l'ensemble d'apprentissage quand ils sont utilisés pour prédire la classe majoritairement couverte.

Le complexe doit également être significatif. Il doit en fait refléter une véritable corrélation entre les valeurs de l'attribut et les classes. *CN2* compare la distribution observée des

exemples parmi les classes avec la distribution qui serait obtenue si le complexe sélectionnait aléatoirement des exemples. Le système utilise le rapport de vraisemblance statistique donné par

$$2\sum_{i=1}^{n} f_i \cdot \log(\frac{f_i}{e_i})$$

où la distribution $F = (f_1, f_2, \ldots, f_n)$ est la distribution de fréquence observée, et $E = (e_1, e_2, \ldots, e_n)$ est la distribution générée aléatoirement.

Sous certaines hypothèses appropriées, cette distance statistique est distribuée approximativement comme la distance du χ^2 avec $n-1$ degrés de liberté. Plus cette mesure est faible, plus l'apparente régularité localisée par le complexe est due au hasard. Seuls les complexes dont cette distance est supérieure à un seuil minimum défini par l'utilisateur seront pris en compte.

L'algorithme *CN2* génère une liste de décision de manière itérative. Une règle est construite à chaque itération, en cherchant un complexe qui couvre un large nombre d'exemples d'une classe arbitraire C_i et peu d'exemples des autres classes. Après avoir trouvé un bon complexe, l'algorithme supprime de l'ensemble d'apprentissage les exemples qu'il couvre, et ajoute la règle « si *complexe* alors *classe* C_i » à la fin de la liste de décision. Avec l'ensemble restant, une nouvelle règle est construite jusqu'à ce qu'on ne trouve plus de complexe de qualité suffisante.

L'algorithme *CN2* construit des règles de production simples et compréhensibles dans des domaines où il peut y avoir du bruit. Il construit des règles probabilistes dans le sens où les prémisses des règles couvrent des exemples d'une seule classe, mais aussi éventuellement quelques exemples des autres classes. Le résultat ne dépend pas de l'ordre dans lequel les exemples sont choisis comme pour les noyaux de *AQ*. Le principal avantage de *CN2* sur *AQ* est qu'il permet d'ajuster un seuil de qualité des règles et qu'il ne se restreint pas à chercher des règles qui sont cohérentes avec l'ensemble d'exemples.

2.5.3.2 Apprentissage à base d'instances

La principale caractéristique des techniques d'apprentissage à base d'instances est qu'elles conservent des exemples typiques pour chaque classe.

Nous présenterons succintement les algorithmes *IBL*, *K** ainsi que les systèmes utilisant le formalisme des treillis.

Les algorithmes IBL (Instance-Based Learning) AHA, KIBLER et ALBERT [AKA91] définissent un ensemble d'algorithmes d'apprentissage à base d'instances qui ont trois fonctions caractéristiques :

- *Une fonction de similarité* :

  Elle mesure à quel point deux instances sont proches l'une de l'autre. Bien que cette démarche paraisse simple, le choix de cette fonction est en réalité complexe en particulier dans le cas où certains des attributs sont nominaux.

- *Une fonction de sélection d' « instance typique »* :

  Elle permet de déterminer les instances qu'il faut garder comme exemples.

- *Une fonction de classification*:

  Cette fonction permet de situer un nouvel individu par rapport à l'ensemble d'apprentissage.

Les algorithmes *IBL* sont connus sous beaucoup d'autres noms, et il existe de nombreuses variantes sur le thème de base. AHA, KIBLER et ALBERT proposent trois variantes dans leur article :

- *IBL1*:

  Cet algorithme conserve toutes les instances exemples et cherche simplement l'instance la plus proche du nouvel individu. La classe de ce nouvel individu est donc celle de la plus proche instance trouvée. Le nombre important d'instances à stocker demande néanmoins une place importante en mémoire.

- *IBL2*:

  Celui-ci procède de la même manière que le précédent, mais enlève les instances de l'ensemble d'apprentissage qui auraient déjà été correctement classées. On gagne ainsi de l'espace mémoire.

- *IBL3*:

  Cette dernière variante fonctionne comme *IBL2*, en faisant quelques hypothèses sur les données et utilise des méthodes statistiques afin d'éliminer les données incohérentes ou bruitées.

On peut en plus appliquer à ces algorithmes des approches du style des k-plus-proches voisins [CH67, Har68, Gat72]. On ne cherche plus la plus proche instance pour classifier un nouvel individu, mais un ensemble d'instances les plus proches. On effectue ensuite un vote pour choisir entre ces dernières.

Remarques :

- On retrouve le même problème de « réglage » que pour les réseaux de neurones car il faut déterminer les trois fonctions citées ci-dessus ;
- Cette manière très peu structurée « d'apprendre » ne permet que de stocker les exemples typiques sans vraiment traiter les données ;
- De plus, comme pour les réseaux de neurones, on n'obtient pas un modèle directement compréhensible par l'homme ;
- Enfin, un autre problème est que ces algorithmes demandent souvent beaucoup d'espace mémoire pour conserver les exemples, et imposent parfois une recherche de la plus proche instance assez longue ;
- Néanmoins, ils sont faciles à implanter, tester, et d'une conception simple.

L'algorithme K^* Un autre algorithme d'apprentissage à base d'instances est l'algorithme K^* [CT95]. Sa particularité est qu'il utilise comme distance de similarité une distance issue de la théorie de l'information.

La distance entre deux instances est en fait définie comme la complexité de transformation de l'une vers l'autre. Un « programme » qui transforme une instance (a) en une instance (b) est une séquence finie de transformations partant de (a) vers (b).

Issue de la théorie des complexités, la définition usuelle de la complexité de KOLMOGOROV d'un objet est la longueur de la plus petite séquence de caractères décrivant cet objet [LV97]. En utilisant cette approche, on peut définir une distance de KOLMOGOROV entre deux instances comme la longueur de la plus petite séquence de transformations qui relie ces deux instances.

Cette approche aboutit à une unique transformation : la plus courte. On obtient donc une mesure de distance qui est très sensible à de légères variations dans l'espace des instances. La distance définie dans K^* essaie de pallier ce problème en prenant en compte toutes les transformations possibles entre deux instances.

Il est en fait possible d'associer une probabilité à chaque séquence. Si la complexité (longueur) d'un programme mesurée en bits est c alors la probabilité associée est 2^{-c}. En particulier, pour chaque distance bien définie basée sur la complexité de KOLMOGOROV, la somme de ces probabilités sur toutes les transformations satisfait l'inégalité de KRAFT : $\sum 2^{-c} \leq 1$. On peut interpréter cette somme comme la probabilité qu'un programme soit généré par une séquence aléatoire de transformations.

En terme de distance entre deux instances, c'est la probabilité que l'on obtienne la deuxième instance par une séquence aléatoire de transformations depuis la première. Cette probabilité est transformée en unité de complexité en prenant le logarithme.

2.5.3.3 Les réseaux de neurones

Les réseaux de neurones essaient d'analyser les données en s'inspirant de manière grossière du fonctionnement d'un réseau de neurone biologique.

Pour comprendre ce qu'est un réseau de neurones, on doit comprendre ce que réalise un simple neurone.

Comme le montre la figure 2.12, un simple neurone contient un ensemble d'entrées (e_j). À chacune d'entre elles est associé un poids (ω_j). Les produits de ces couples entrée-poids (e_j, ω_j) sont additionnés. Une fonction monotone croissante (donnant la valeur de sortie du neurone) est alors appliquée à cette somme.

Les neurones simples sont habituellement combinés pour former un réseau de neurones. En général ces neurones sont regroupés par couches successives (figure 2.13). La première couche, appelée couche d'entrée est composée de neurones dont les entrées sont les entrées externes. La couche cachée (interne) prend en entrée les sorties de la couche d'entrée. Il peut y avoir plusieurs couches cachées avant d'atteindre la couche de sortie qui relie à nouveau le réseau au monde extérieur. Chaque couche prend en entrée les sorties de la couche précédente.

La réelle puissance des réseaux de neurones vient de la manière dont les différents poids

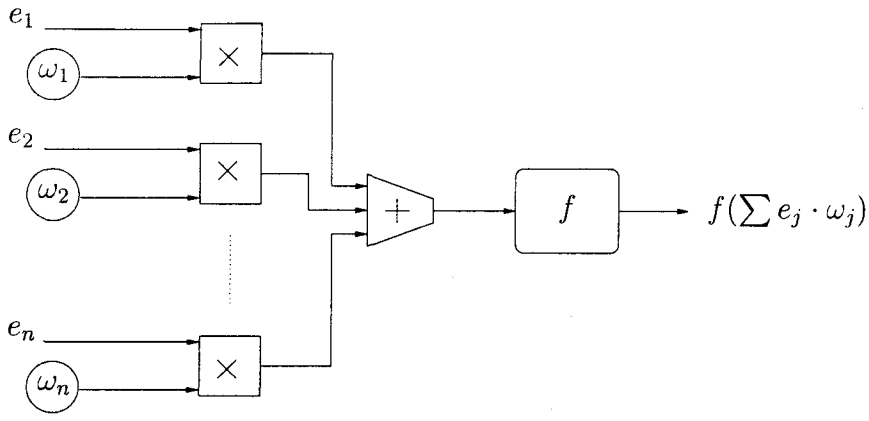

FIG. 2.12: *Structure interne d'un neurone*

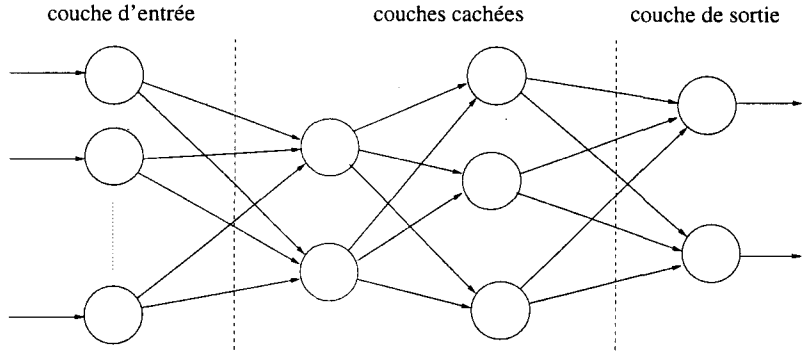

FIG. 2.13: *Un réseau de neurones*

sont ajustés. Lorsqu'il s'agit d'apprentissage supervisé, les poids sont initialement choisis arbitrairement et des exemples sont fournis à l'entrée du réseau de neurones. Les sorties sont alors comparées à ce qu'elles auraient dû être. À l'aide de techniques telles que la « rétro-propagation du gradient », les poids des neurones sont réajustés. Ainsi, le réseau de neurones « apprend » ce qu'est la sortie souhaitée, et c'est là qu'il devient utile.

Il y a différentes manières d'utiliser un réseau de neurones en tant que classifieur. L'une d'elles est d'avoir un neurone pour chaque classe possible. Ainsi, chacun de ces neurones aura une valeur proche de 1 quand la classe associée sera indiquée. Un problème survient lorsque plusieurs de ces neurones indiquent que l'exemple présenté est de leur classe. Une seconde manière, moins fréquente, est de coder les classes de façon binaire. Ceci permet d'éviter le problème précédent, mais peut conduire à affecter la précision de la fonction de sortie.

Les réseaux de neurones sont réglables et paramétrables. En effet, on peut changer le codage de la sortie, le nombre de couches cachées, le nombre de neurones par couche, l'interconnection des neurones, le choix de la fonction de sortie, le choix de la fonction d'ajustement des poids lors de l'apprentissage... Un inconvénient de ces approches est qu'il n'existe pas à notre connaissance de procédure structurée permettant de choisir la

configuration correcte pour une tâche donnée.

Les principaux algorithmes utilisé dans la littérature ayant été présentés, nous nous proposons de définir les critères permettant de caractériser la qualité d'un modèle.

2.6 Qualité (validité) du modèle

Chercher la relation mathématique $Y = f(X)$ consiste, pour une classe de fonctions f déterminée, à en rechercher la valeur optimale des paramètres (au sens d'un critère établi) sur la base d'un ensemble d'observations. La prédiction des valeurs de Y pour des valeurs de X déjà observées est directe. Par contre, la prédiction des valeurs de Y pour des valeurs de X non encore observées est directe si f est définie sur ces valeurs. Dans le cas contraire, pour certaines valeurs de X, $f(X)$ pourra ne pas être définie, et donc la valeur de Y ne pourra être évaluée.

Dans le cas de relations non analytiques (par exemple, dans le cas de relations à base de règles), la prédiction des valeurs de Y pour des valeurs de X déjà observées pose théoriquement des problèmes dans le cas où la population d'apprentissage est incohérente. L'extension à des cas non encore observés semble encore plus complexe.

Dans ce paragraphe, nous nous intéresserons au cas des relations à base de règles liant X et Y.

2.6.1 Qu'est-ce qu'un modèle valide?

Un des problèmes majeurs de l'apprentissage inductif est que le système cognitif peut construire des modèles corrects sur la base de données ou sur l'ensemble d'apprentissage. C'est-à-dire qu'il vont prédire correctement la sortie, pourvu qu'on leur fournisse en entrée un exemple d'apprentissage. Mais ils peuvent très bien donner de mauvais résultats s'ils sont en présence de situations non encore rencontrées.

Le modèle $Y = f(S)$ étant établi, l'utilisateur cherchera à prédire la valeur de Y sachant les valeurs que prennent les variables constituant S. Il nous faut dès lors passer du problème d'explication au problème de prédiction en nous posant les deux questions suivantes :

- Connaissant le modèle $Y = f(S)$ établi, est-il possible de prédire la valeur de Y sachant une valeur de S déjà observée? On se pose alors la question de la validité de la valeur de Y. Il doit être possible de quantifier « l'efficacité » de la règle en terme de prédiction (une efficacité proche de 1 conduira à une bonne prédiction, contrairement à une efficacité proche de 0).

- Connaissant le modèle $Y = f(S)$ établi, est-il possible de prédire la valeur de Y sachant une valeur de S non encore observée? On cherche alors à étendre le modèle afin de déterminer la valeur de Y. Les méthodes à base de modèle paramétrique donnent

un résultat implicite, alors que les méthodes sans modèle paramétrique de fonctionnement demandent un effort supplémentaire afin de passer du modèle (construit à partir de l'espace observé) à l'espace observable [Pom91].

Pour un grand nombre de systèmes, le nombre d'états possibles (le nombre de modalités observables de X) est infini ou très important. C'est le cas par exemple des systèmes pour lesquels les variables sont continues. On ne peut donc pas vérifier la validité du modèle pour toutes les situations possibles. On doit donc l'estimer. Si l'on peut construire plusieurs modèles, certains d'entre eux seront plus simples que d'autres. On s'attend à ce que les modèles les plus simples soient également les meilleurs. Le raisonnement sous-jacent, connu aussi sous le nom de principe du Rasoir d'Occam [Sil68], est que s'il existe plusieurs explications pour un même phénomène, il est de bon sens de choisir la plus simple, car il est plus probable qu'elle représente la vraie nature du phénomène.

Un compromis entre la simplicité d'un modèle et son efficacité doit alors être trouvé. Ce compromis dépend bien entendu du type de système physique étudié ainsi que des objectifs voulus.

Ce principe se traduit dans la communauté « informatique et apprentissage » sous le nom de principe MDL (Minimum Description Length) [Ris78, Ris83].

Pour définir ce principe, on considère un émetteur et un récepteur. L'émetteur et le récepteur connaissent tous deux une liste d'individus dans la population d'apprentissage. Mais seul l'émetteur connaît la « classe » de ces individus, et doit le transmettre au récepteur. Pour cela, il transmet une « théorie » qui permet de classifier chaque individu. Et si cette théorie est imparfaite, il doit également transmettre les exceptions et leur classe. La longueur totale de la transmission est donc la longueur du codage de la théorie à laquelle on ajoute la longueur du codage des exceptions. Le principe MDL dit que parmi toutes les théories candidates, il faut choisir celle qui minimise cette longueur totale afin par exemple, de réaliser un compromis entre simplicité et précision d'une théorie.

Certains algorithmes d'apprentissage utilisent ce principe pour généraliser les règles du modèle obtenu [Qui95].

2.6.2 Démarche générale des méthodes d'apprentissage

Il est possible d'avoir une procédure qui classifie bien tous les exemples de la population d'apprentissage mais qui ait un mauvais pouvoir de prédiction. L'objectif d'un système d'apprentissage est de construire une procédure de classification qui soit non seulement correcte sur l'échantillon mais ayant en plus un bon pouvoir de prédiction sur de nouveaux exemples. Il sera demandé à la procédure de classification induite de dépasser le pouvoir prédictif de la procédure majoritaire (qui associe à toute description la classe la plus fréquente).

Il devient donc nécessaire de procéder à une validation du système cognitif obtenu. Il est bien entendu évident qu'on ne peut pas à la fois apprendre sur une population d'apprentissage et valider sur ce même ensemble. La validation repose alors nécessairement

sur l'application du modèle (obtenu à partir de la population d'apprentissage) sur une population test. Trois approches se présentent alors :

- La première idée est de disposer d'un ensemble permettant de tester la qualité de la procédure de classification induite. On partitionne l'échantillon en un ensemble d'apprentissage et un ensemble test. La répartition entre les deux ensembles doit être faite aléatoirement. Il est alors possible d'estimer avec l'ensemble test, une des fonctions de qualité citées ci-après ou la précision de prédiction ou encore l'erreur de prédiction. La qualité de l'apprentissage augmente avec la taille de l'ensemble d'apprentissage, de même, la précision de l'estimation de la fonction de qualité augmente avec la taille de l'ensemble test. Mais, dans la pratique, la taille de l'échantillon est limitée. Cette méthode donne de bons résultats lorsque l'échantillon est « assez » grand. Il existe peu de résultats théoriques sur les tailles d'échantillon nécessaires pour utiliser cette méthode, on ne dispose que de résultats empiriques qui dépendent du problème (souvent, plusieurs centaines d'exemples). La répartition de l'échantillon entre les deux ensembles se fait en général dans des proportions 1/2, 1/2 pour chacun des deux ensembles ou 2/3 pour l'ensemble d'apprentissage et 1/3 pour l'ensemble test.

- Une deuxième méthode est la *validation croisée*. Elle consiste à découper l'échantillon en k sous-ensembles. Un ensemble d'apprentissage consiste en la réunion de $k-1$ sous-ensembles et un ensemble test au k^{ieme} sous-ensemble. On exécute alors l'apprentissage sur chacun des k ensembles d'apprentissage et on estime la fonction de qualité sur l'ensemble test correspondant. On calcule alors en général la moyenne des mesures obtenues.

- La dernière méthode est celle du *bootstrap*. Étant donné un échantillon S de taille n, on tire avec remise un ensemble d'apprentissage de taille n (un élément de S peut ne pas appartenir à l'ensemble d'apprentissage, ou y figurer plusieurs fois), l'ensemble test est S. L'estimation de la fonction de qualité est alors la moyenne des fonctions de qualité obtenues pour un certain nombre d'itérations de l'algorithme d'apprentissage.

Les deux dernières méthodes fournissent de bons estimateurs de la validité du modèle mais sont très coûteuses en temps de calcul.

2.6.3 Fonctions de qualité utilisées dans la littérature

Ces fonctions de qualité sont utilisées comme critères dans les algorithmes décrits, et doivent intégrer les notions de simplicité et de précision du modèle introduites ci-dessus.

De manière générale, une fonction de qualité assigne une valeur comprise entre 0 et 1 à chaque description, règle ou modèle considéré.

Il est possible de distinguer deux aspects de la qualité d'une description. Une description doit être *valide*, c'est-à-dire qu'elle doit être capable de classifier correctement un objet

inconnu. De plus, elle doit être *correcte (ou exacte)* vis-à-vis de la classe considérée. Ces différents critères peuvent être combinés pour former une fonction de qualité générale.

- En général, la *validité* d'une règle ne peut pas être prouvée, car son exactitude ne peut pas être vérifiée pour tous les cas possibles. On a donc besoin de l'estimer ou de trouver quelques indications sur l'éventuelle validité de la description. La plupart des systèmes cognitifs reposent sur le principe du rasoir d'Occam : plus la description est simple, plus elle est susceptible de décrire une relation réellement présente dans la base de données. Cette complexité peut être mesurée comme la taille de la description, la validité est ainsi plus grande pour les descriptions les plus simples.

- L'*exactitude* d'une description est maximale lorsqu'elle couvre tous les exemples positifs et aucun exemple négatif de la classe considérée. Nous avons cependant besoin d'étendre cette notion d'exactitude. En effet, certaines descriptions, bien que inexactes, peuvent être utiles pour la construction d'autres « meilleures » descriptions. Ainsi, il est intéressant d'associer différentes valeurs traduisant l'exactitude plutôt qu'une valeur booléenne. Parmi celles-ci, on peut introduire la *précision de classification* ainsi que la *couverture* d'une description.

  - La *précision de classification* est la probabilité que la règle classe correctement un individu, c'est-à-dire la probabilité qu'un objet couvert par la règle appartienne réellement à la classe considérée. Celle-ci est estimée par le rapport de l'effectif des individus à la fois couverts par la règle et la classe considérée, et de l'effectif des individus couverts par la règle. En reprenant les notations utilisées sur la figure 2.14 (parfois appelée diagramme de VENN), la précision de classification de la règle vaut :

    $$précision = \frac{t_p}{f_p + t_p}$$

    En généralisant pour le modèle complet à deux classes pour cet exemple, on obtient :

    $$précision = \frac{L - f_p - f_n}{L}$$

  - La *couverture* d'une description est la probabilité qu'un individu appartenant à la classe considérée soit couvert par cette description. Elle est évaluée par :

    $$couverture = \frac{t_p}{t_p + f_n}$$

  À partir de ces deux mesures, différentes règles peuvent être établies :

  - Si la couverture vaut 1, la règle est dite *complète*, c'est-à-dire que chaque individu appartenant à la classe est couvert par la règle ($f_n = 0$).
  - Si la précision de classification vaut 1, alors la règle est dite *déterministe*. Chaque objet couvert par la règle appartient à la classe considérée ($f_p = 0$).

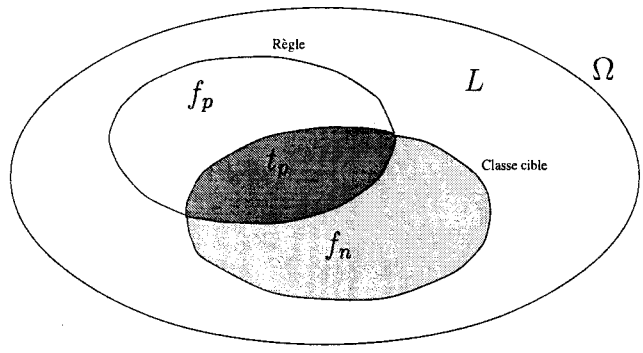

FIG. 2.14: *Diagramme de* VENN

- Si les deux mesures de précision et de couverture valent 1, alors la règle est évidemment *exacte* ($f_p = f_n = 0$).

Il existe de multiples façons de calculer le critère d'exactitude lorsque la description n'est pas strictement exacte. Dans ce cas, il est inférieur à 1. Dans [PS91], PIATETSKY-SHAPIRO propose quelques principes pour la construction d'une telle fonction.

La qualité d'une description doit donc dépendre de sa validité ainsi que de son exactitude, mais peut également prendre en compte d'autres facteurs tels que les coûts d'évaluation de la description, ou les coûts de mesure des attributs considérés, etc... Les différents critères combinés pour calculer la fonction de qualité globale peuvent aussi être pondérés de manière à en favoriser certains.

On peut également trouver quelques exemples de mesure de qualité d'un modèle dans [DF96].

2.6.4 De l'explication à la prédiction

Un modèle $Y = f(S)$ étant défini, considérons une nouvelle observation notée ω_{obs} caractérisée par :

- l'observation de la variable $S' \subset X$ dont la modalité est s' ;
- l'observation (ou non) de la variable Y dont la modalité est β' (ou « ? » dans le cas où elle n'est pas observée).

Remarque : Dans le cadre du diagnostic, cette phase de prédiction est appelée « *pronostic* » : par l'intermédiaire d'une ou de plusieurs observations des variables composant S, l'utilisateur pourra utiliser son modèle à des fins de prédictions concernant l'état de fonctionnement du système (dans un avenir plus ou moins proche).

Nous utiliserons la notation r_β pour définir une règle ayant la conclusion β (*i.e.* d'après le modèle, la variable Y devrait prendre la modalité β).

Modalité observée de Y / s' observée	$\beta' \subset M_Y$	$\beta' \not\subset M_Y$	?
$\exists r_\beta$ couvrant s' et $\beta = \beta'$	① validation (renforcement) du modèle on conclut $Y = \beta'$	②	③ prédiction on conclut $Y = \beta$
$\exists r_\beta$ couvrant s' et $\beta \neq \beta'$	④ mauvaise prédiction invalidation de tout ou partie du modèle	⑤ évolution du système avec invalidation du modèle	⑥ prédiction on conclut $Y = \beta$
$\nexists r$ qui couvre s'	⑦ extension du modèle	⑧ évolution du système ⇒ évolution du modèle par la prise en compte de la nouvelle classe	⑨ prédiction avec extension du modèle

cas identique

TAB. 2.1: *Les différents cas de figure permettant de passer du problème d'explication au problème de prédiction*

Nous pouvons alors établir le tableau 2.1, d'où plusieurs cas de figure ressortent :

- il existe une règle r_β qui couvre la nouvelle observation s' et $\beta = \beta'$ qui représente la modalité observée de Y. Dans ce cas, on conclut $Y = \beta$ et le modèle est validé (ou renforcé). Si l'on n'observe pas Y, l'utilisateur peut conclure que $Y = \beta$;

- il existe une règle r_β qui couvre la nouvelle observation s' mais $\beta \neq \beta'$. Dans ce cas, le modèle est invalidé et le système a effectué une mauvaise prédiction. Si la modalité β' n'avait pas été observée lors de l'apprentissage, alors l'évolution du modèle afin de prendre en compte β' semble incontournable. Si l'on n'observe pas Y, le système ne peut que conclure $Y = \beta$;

- il n'existe pas de règle qui couvre la nouvelle observation s'. Il est alors nécessaire d'étendre le modèle. Cette évolution peut entraîner une prise en compte d'une nouvelle valeur de Y non encore observée jusqu'alors.

Ce tableau est illustré par la figure 2.15.

Cette figure représente dans le cas d'un espace à deux dimensions, les individus observés pendant l'apprentissage (en noir), et quelques nouvelles observations de la variable S' (entourées en pointillé). Dans ce cas, la modalité observée de la variable Y est indiquée en gris lorsqu'elle existe.

M_{X_1} et M_{X_2} représentent respectivement les ensembles de modalités observées de X_1 et X_2.

Une correspondance entre les différents cas possibles et les cases du tableau 2.1 peut être faite :

- cas ⑤ et ⑧ → évolution du modèle initial ;

- cas ③, ⑥ et ⑨ → prédiction (on ne connaît pas Y et on fait confiance au modèle appris) ;

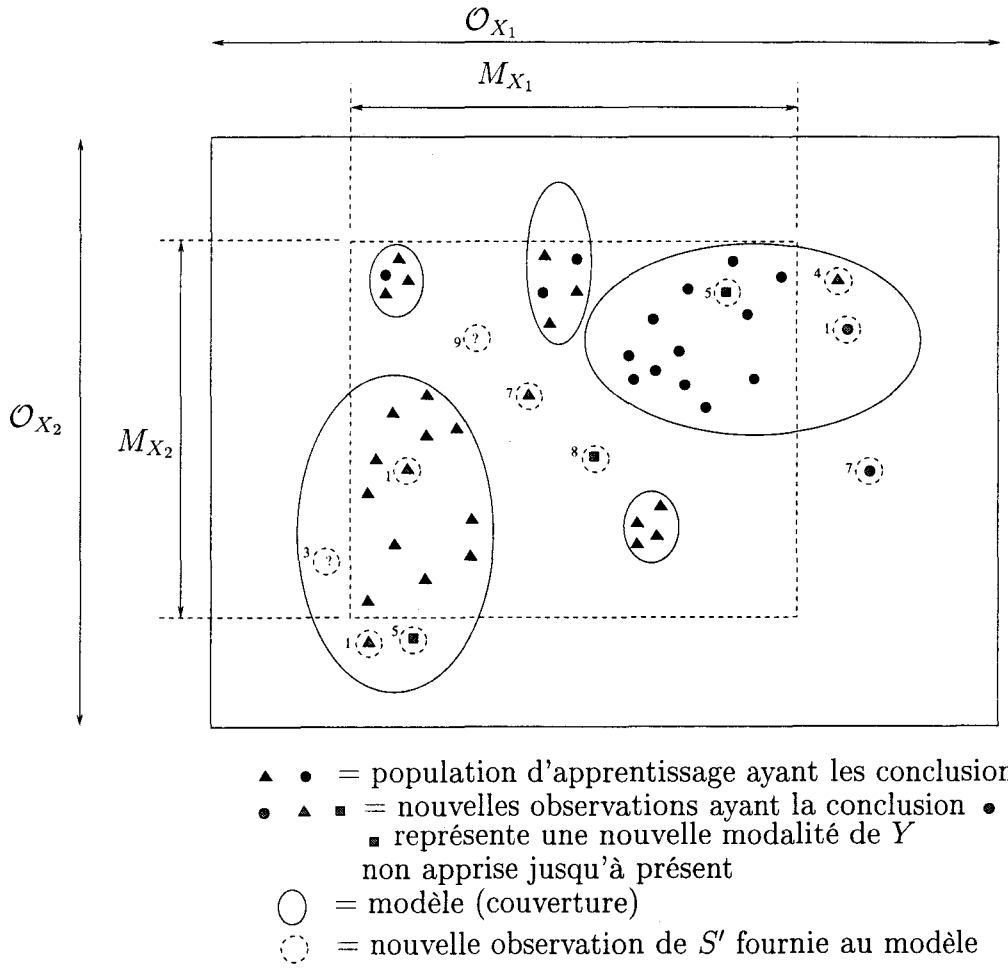

FIG. 2.15: *Du problème d'explication au problème de prédiction*

- cas ① → bonne prédiction ;
- cas ④ → mauvaise prédiction ;
- cas ⑦ → extension du modèle.

Il est à noter que l'ensemble S' des variables observées du nouvel individu est inclus dans X. Trois cas se présentent alors :

- $S' = S$ et une prédiction est envisageable (voir tableau 2.1) ;
- $S' \neq S$ et ($S' \cap S \neq \emptyset$ ou $S' \subset S$), la prédiction est dégradée. Nous devons garder en mémoire les différentes distributions de probabilité des différentes règles susceptibles de couvrir l'individu. En effet, d'après A. KAUFMANN dans « Les expertons » (p. 9) : « *La valuation d'un groupe est la plupart du temps agrégée en une seule valuation pour chaque donnée : autrement dit, on fait tomber le désordre ou entropie au départ. Pourquoi ne pas le faire disparaître le plus tard possible et travailler avec les*

statistiques elles-mêmes plutôt qu'à l'aide des représentations réduites : moyennes, espérances mathématiques, variances, moments, etc... ? Dans bien des cas, cela est possible et souhaitable. Progressivement, en étudiant de plus en plus en profondeur l'utilisation associée des mathématiques du flou et celle des probabilités, je me suis aperçu qu'une sorte de sainte règle devait être suivie dans la modélisation et dans le traitement informatique : faire tomber l'entropie probabiliste ou non probabiliste le plus tard possible. » ;

- $S' \cap S = \emptyset$, on ne peut pas prédire.

2.7 Conclusion

L'objectif de ce chapitre était de montrer que les problèmes de l'analyse structurale des systèmes complexes peuvent être appréhendés par des méthodes issues de l'apprentissage automatique.

C'est dans ce cadre que les algorithmes d'apprentissage symbolique (les arbres de décision, les systèmes à base de règles), les algorithmes à base d'instances (*IBL*, *K**) ainsi que les réseaux de neurones ont été très sommairement présentés.

Les modèles obtenus peuvent être nombreux. L'utilisateur devra alors choisir le modèle ayant la meilleur *qualité*. Cette qualité représente une mesure déterminée *a posteriori* afin de prédire l'état des variables Y sachant la relation $Y = f(X)$ et ayant mesuré (ou observé) X.

Pour pallier une mauvaise qualité de modèle(s), des méthodes ont été développées afin de déterminer une rupture de modèle dans le cas de systèmes non stationnaires. Ces méthodes consistent à choisir entre deux hypothèses, l'hypothèse H_0 pour laquelle on considère que le modèle considéré est bon, et l'hypothèse H_1 pour laquelle le modèle retenu ne correspond plus au comportement réel du système ([Nik95, BN93, PSS96]).

Deuxième partie

Méthodologie d'étude d'un système physique par construction d'arbres de décision

Introduction de la deuxième partie

La première partie de ce mémoire permettait de présenter le cadre du travail. C'est ainsi que nous avons présenté les données sur lesquelles repose le travail. Puis les méthodes issues de l'apprentissage automatique ont été replacées dans le cadre de l'analyse structurale des systèmes complexes.

Cette seconde partie propose d'établir le schéma suivant :

Dans un premier temps, nous présenterons les outils de la théorie de l'information appliquée à l'analyse structurale des systèmes, ce qui nous permettra d'établir un critère d'étude utilisable dans nos algorithmes. Quelques propriétés intéressantes pour notre étude seront alors mises en exergue. C'est ainsi que l'entropie conditionnelle pourra être vue comme la part d'incertitude sur les conclusions ayant pris connaissance des variables testées.

Le critère étant établi, nous proposerons de définir des algorithmes permettant la construction d'un modèle de comportement par l'intermédiaire d'arbres de décision. Afin de définir les principes généraux des arbres de décision, nous présenterons *C4.5* qui est certainement l'algorithme de construction d'arbres de décision le plus usité. Plusieurs points de vue de construction d'arbres de décision peuvent être considérés et combinés afin d'obtenir des algorithmes performants. C'est ainsi que les méthodes de construction par niveau seront opposées aux méthodes de construction par nœud, et des comparaisons des différentes méthodes seront établies.

Enfin, un dernier chapitre permettra de valider les méthodes développées sur des cas concrets. Nous proposerons ainsi de traiter des tableaux de données disponibles sur le WEB et permettant à la communauté de l'apprentissage automatique une comparaison des méthodes de façon purement empirique. Nous traiterons également une base de données issues d'un moteur asynchrone afin de nous replacer dans le cadre du diagnostic. L'objectif de cette étude est de discriminer l'état de fonctionnement normal du moteur, ainsi qu'un état de dysfonctionnement, à savoir un dysfonctionnement de type « capteur », une dérive de la résistance dite « rotorique » étant considérée comme un fonctionnement normal.

Chapitre 3

Les outils de la théorie de l'information appliquée à l'analyse structurale des systèmes

La théorie de l'information (T.I.) est née en 1948 avec les travaux de C.E. SHANNON [Sha48, Sha49a, Sha49b] et de N. WIENER qui ont connu d'emblée un retentissement considérable.

Conçue initialement pour l'étude de la transmission des messages émis par une source à un récepteur à travers un canal soumis au bruit, celle-ci a connu un développement rapide et fécond [Sle74]. De nombreux travaux ont été consacrés à cette théorie, qui ne sont en définitive qu'une exploitation et extension des travaux proposés par SHANNON, HARTLEY [Har28], WIENER et NYQUIST [Nyq24]. Cette théorie a subi plusieurs mutations, de grandes voies se sont dégagées principalement dans les trois domaines suivants:

- Communication: dans ce cas, on étudie les phénomènes liés à l'acheminement des signaux, aux organes de transmission, aux fonctions de modulation, de détection, etc... [Wyn81, Osw86].

- Mathématique: la T.I. est vue sous un angle plus abstrait. Dans le cadre de la théorie de la mesure, de l'information relativiste, plusieurs généralisations ont été proposées soit au niveau purement mathématique ou bien pour des applications particulières. Nous citons parmi les auteurs: KAMPÉ DE FÉRIET, BENVENUTI, B. FORTE [KF67, KFB69, KB69, KB72] sur l'information généralisée; C. LANGRAND [Lan73], J. LOSFELD, A. DUSSAUCHOY [Dus80], J.P. BARTHÉLÉMY [BG88], S.GUIASU [Gui77] pour d'autres types d'informations que celles de SHANNON et enfin G. JUMARIE [Jum78, Jum79a, Jum79b, Jum80] pour l'information relativiste.

- Analyse des systèmes: elle a pour objet l'étude des flux d'informations échangés entre les différentes composantes du système. **C'est dans cette optique que nous nous plaçons pour présenter des résultats relatifs à l'analyse des systèmes dans l'esprit de l'Automatique.**

Dans ce domaine, les premiers travaux viennent essentiellement du côté américain avec W.R. ASHBY [Ash56, Ash65b, Ash69], R.C. CONANT [Con69, CA70, Con76], G. KLIR [Kli69, Kli75, Kli76, Kli77] et avec M. RICHETIN [Ric75], J. DUFOUR [Duf79], DUSSAUCHOY [Dus80], V. TORO [Tor82], M. SBAÏ [Sba83], M. BARBOUCHA [Bar87], D. POMORSKI [Pom91] côté français. Ainsi, dans la plupart de ces travaux, on met en évidence les notions relatives aux lois de transmission et d'évolution de l'information dans un système, on étudie les relations existant entre la disponibilité de l'information et la capacité de contrôle et de régulation d'un système. On distingue également les aspects de sélection, de blocage, de transmission interne et externe des quantités d'information créées ou relayées par le système.

Toutes ces démarches reposent sur des concepts de mesure d'information et d'entropie.

3.1 Introduction à la théorie de l'information

Le caractère aléatoire inhérent aux processus de transmission de l'information implique l'utilisation des méthodes statistiques pour l'étude de ces processus. Pourtant on ne peut se limiter aux concepts classiques de la Théorie des Probabilités et d'autres notions probabilistes doivent être introduites. C'est ainsi que la Théorie de l'Information peut aider à pallier ces problèmes.

Supposons un système physique pouvant se trouver dans un état quelconque. Ce système est alors caractérisé par un certain degré d'incertitude. Les renseignements obtenus sur ce système sont d'autant plus importants que son incertitude *a priori* est grande. Il paraît donc important de disposer d'une mesure de son degré d'incertitude.

Afin de mieux comprendre cette notion, considérons deux systèmes : une pièce de monnaie bien équilibrée et un dé à six faces non pipé. La pièce de monnaie, jetée en l'air, peut présenter deux côtés différents : un côté pile et un côté face équiprobables. Le dé possède six états potentiels différents, équiprobables entre eux.

On peut alors se poser la question suivante : quel est le système possédant l'incertitude la plus grande? Il apparaît logique de dire que le système ayant l'incertitude la plus grande est le dé car son nombre d'états possibles (potentiels) est plus grand. Nous pourrions donc penser que le degré d'incertitude d'un système physique est caractérisé par le nombre d'états potentiels de ce système.

Montrons que la seule connaissance du nombre des états potentiels n'est pas suffisant : Prenons par exemple un système quelconque ayant les caractéristiques suivantes :

- il peut se trouver dans un état (1) (par exemple, un état de bon fonctionnement), qui a une probabilité *a priori* de 0.99 ;

- il peut également se trouver dans un second état, noté (2) (état de panne ou d'arrêt), qui est réalisé avec une probabilité *a priori* de 0.01.

Le degré d'incertitude sur l'état du système est très faible car il a toutes les chances de se trouver dans l'état (1) beaucoup plus souvent que dans l'état (2).

Nous voyons donc que le degré d'incertitude d'un système physique est déterminé par :

- le nombre d'états potentiels ;
- la probabilité d'obtention (d'occurrence) de ces états.

Appelons *Entropie* la mesure du degré d'incertitude d'un système physique.

Définition de l'entropie

Soient Ω un univers d'événements et A un événement donné. Sa réalisation apporte une quantité d'information et on note $I_\Phi(A)$ ce nombre [Agg74, Agg76] tel que :

$$I_\Phi(A) = \Phi(P(A)) - \Phi(P(\Omega))$$

où :

- Ω est l'ensemble des événements élémentaires ω ;
- $P(A)$ est la probabilité d'occurence de l'événement A ;
- Φ est une application de $[0,1]$ dans $\mathbb{R}^+$ telle que $\Phi(1) \geq 0$.

Par ailleurs, on sait que la réalisation d'un événement peu probable apporte beaucoup d'information alors que la réalisation d'un événement très probable (ou presque certain) apporte peu d'information. C'est pourquoi, une mesure d'information doit être une fonction décroissante de probabilité. Toujours selon AGGARWAL, $I_\Phi(A)$ vérifie les axiomes fondamentaux de KAMPÉ DE FÉRIET et FORTE. L'information $H_\Phi(X)$ fournie par la partition : $X = \{A_1, A_2, \ldots, A_n\}$ (encore appelée entropie de X) est définie par :

$$\begin{aligned} H_\Phi(X) &= \sum_{i=1}^{n} P(A_i) \cdot I_\Phi(P(A_i)) \\ &= \sum_{i=1}^{n} P(A_i) \cdot \Phi(P(A_i)) - \Phi(1) \\ &= \sum_{i=1}^{n} p_i \cdot \Phi_i - \Phi(1) \end{aligned}$$

en notant p_i la probabilité de trouver le système X dans l'état (i) $(i = 1, \ldots, n)$.

De nombreuses entropies ont été définies (cf [Bar87], pp.I.27-I.28). Nous nous intéresserons, dans le cadre de ce travail, à deux entropies couramment utilisées :

- l'entropie du MAX : $\Phi(t) = (1/t - 1) \cdot \delta(t - p_M)$

  où :

  - $p_M = \max_i p_i$

- $\delta(t - p_M)$ est l'impulsion de Dirac au point p_M.

L'entropie du MAX est donc définie par : $H_M(X) = 1 - \max_i p_i$

- l'entropie de SHANNON : $\Phi(t) = -\log(t)$

$$\begin{aligned} H_S(X) &= -\sum_{i=1}^{n} p_i \cdot \log p_i \\ &= E[-\log p(X)] \end{aligned}$$

où $E[f(x)]$ est la fonction espérance mathématique (encore appelée moyenne pondérée) de $f(x)$.

Compte tenu de ces définitions, nous pouvons remarquer que l'entropie du MAX correspond à une approche locale, contrairement à l'entropie de SHANNON, mieux adaptée à un problème du type global (dû au signe somme Σ qui permet de faire une moyenne sur un domaine donné).

D'autre part, l'entropie de SHANNON vérifie certaines propriétés intéressantes que nous examinerons par la suite, justifiant son utilisation comme mesure du degré d'incertitude d'un système.

Remarque :

La base du logarithme peut être quelconque (strictement supérieur à 1). Le changement de base est équivalent à une simple multiplication de l'entropie par un nombre constant. En pratique, on utilise des logarithmes de base 2. On mesure donc l'entropie en unités binaires (bits). Ceci s'accorde bien avec le système binaire de représentation des informations dans les calculateurs. Ultérieurement, le symbole log représentera des logarithmes binaires.

3.2 L'entropie de Shannon

Reprenons la définition de l'entropie de SHANNON :

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i$$

où :

- $\log \equiv \log_2$;
- $\sum_{i=1}^{n} p_i = 1$;
- n est le nombre d'états possibles du système.

Propriétés :

- $H(X) = \log n = \max_{p_i} H(X) \Longleftrightarrow$ système à n états équiprobables

$$\forall p_i \quad p_i = 1/n \quad (i = 1, \ldots, n)$$

et dans ce cas, l'entropie est maximale.

- $H(X) = 0 \Longleftrightarrow$ système à état unique

L'entropie d'un événement certain est nulle.

- Dans le cas général, $0 \leq H(X) \leq \log n$.

On peut remarquer qu'il est équivalent de mesurer l'entropie d'une variable ou celle de la partition de Ω induite par cette variable. $H(X) = H(P_X(\Omega))$.

Entropie d'un système composé

Considérons deux variables, notées X et Y, du système étudié. Chacune de ces variables possède son propre ensemble de modalités :

$M_X = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ et $M_Y = \{\beta_1, \beta_2, \ldots, \beta_m\}$.

Supposons également que les probabilités conjointes d'obtention des modalités α_i et β_j soient correctement estimées par les fréquences relatives.

Nous pouvons reprendre le tableau de contingence suivant (tableau 3.1) :

M_X \ M_Y	β_1	$\ldots$	β_j	$\ldots$	β_m	
α_1						
$\vdots$			$\vdots$			
α_i		$\ldots$	p_{ij}	$\ldots$		$p_{i.}$
$\vdots$						
α_n						
			$p_{.j}$			

TAB. 3.1: *Tableau de contingence*

où :

- X et Y sont deux variables du système. Elles peuvent être vectorielles ou primaires ;
- $M_X = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ et $M_Y = \{\beta_1, \beta_2, \ldots, \beta_m\}$ sont respectivement les ensembles de modalités potentielles de X et de Y ;
- $p_{i.} = \sum_j p_{ij}$ pour $i = 1, \ldots, n$;

- $p_{.j} = \sum_i p_{ij}$ pour $j = 1, \ldots, m$.

Par définition de l'entropie de SHANNON, nous avons :

$$\begin{aligned} H(X,Y) &= -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \cdot \log p_{ij} \\ &= E[-\log p(X,Y)] \end{aligned}$$

Remarques :

- Nous pouvons montrer l'inégalité suivante :

$$H(X,Y) \leq H(X) + H(Y)$$

- En vertu du théorème de multiplication des probabilités des événements indépendants : $P(X,Y) = P(X) \cdot P(Y)$

  on a $\log P(X,Y) = \log P(X) + \log P(Y)$

  et donc $H(X,Y) = H(X) + H(Y)$

- On peut également étendre la définition de l'entropie d'un ensemble de plus de deux variables.

Quelques propriétés :

- Soient A et B deux vecteurs de variables, alors

$$H(A \cup B) \leq H(A) + H(B)$$

  avec égalité si et seulement si A et B sont des vecteurs statistiquement indépendants de variables (voir remarque précédente). On peut expliquer ceci par le fait qu'un groupe de variables peut contenir un peu d'information sur les autres (par exemple, connaissant l'âge d'un jeune enfant on a déjà une idée sur sa taille). Ainsi, dans $H(A) + H(B)$ on compte deux fois cette partie commune d'information. Et c'est seulement lorsque ces groupes de variables sont tout à fait indépendants qu'ils n'apportent aucune information répétée (la connaissance de l'âge d'un jeune enfant ne nous permet pas de tirer des renseignements sur son état de santé).

- Soient Q et R deux partitions de Ω,

  si $Q \preccurlyeq R$ alors $H(Q) \geq H(R)$

- De même, si X et Y sont deux variables définies sur Ω,

  si $X \preccurlyeq Y$ alors $H(Y) \leq H(X)$

  On avait remarqué au chapitre 2 que si X était plus fine que Y, il existait une relation $Y = f(X)$. En d'autres termes, connaissant X il est possible de connaître Y. Il est donc normal que X apporte au moins autant d'information que Y.

- A et B étant deux vecteurs de variables primaires,

  si $A \subseteq B$[11] alors $H(A) \leq H(B)$.

  Il est évident que l'information et/ou l'incertitude doit augmenter lorsque l'ensemble de variables devient plus grand.

La mesure de l'information par l'entropie est particulièrement adaptée au problème de visualisation des variables primaires. En effet, il consiste à chercher un vecteur $S \subset \mathcal{P}(\Sigma)$ tel qu'il contienne un maximum d'information, c'est-à-dire tel que son entropie soit maximale.

À titre d'exemple, sur le tableau extrait de [Mar87] donné en annexe (tableau A.1), on peut calculer :

$$H(Tail.) = -(15/27 \cdot \log 15/27 + 5/27 \cdot \log 5/27 + 7/27 \cdot \log 7/27) = 1.43\text{bits}$$

(15 modalités «+», 5 modalités «*o*», et 7 modalités «−» pour une population d'apprentissage composée de 27 individus).

De même :

$H(Poids) = 1.46$ bits	(5 « + », 14 « o », 8 « - »)
$H(Veloc.) = 1.58$ bits	(9 « + », 8 « o », 10 « - »)
$H(Intell.) = 1.51$ bits	(6 « + », 13 « o », 8 « - »)
$H(Aff.) = H(Agr.) \approx 1$ bit	
$H(Fonction) = 1.58$ bits	(8 « Ut. », 13 « Ch. », 10 « Co. »)

Les variables apportant le plus d'information sont les variables *Fonction* et *Vélocité*. Par contre, les variables *Affection* et *Agressivité* n'apportent que très peu d'information.

De même, on pourrait calculer l'entropie des variables vectorielles.

3.3 L'entropie conditionnelle

Considérons maintenant deux systèmes X et Y dépendants l'un de l'autre.

$p(\beta_j/\alpha_i)$ représente la probabilité conditionnelle pour que le système Y se trouve dans l'état β_j lorsque le système X se trouve dans l'état α_i.

Or, l'entropie du système Y lorsque le système X se trouve dans l'état α_i est :

$$H(Y/\alpha_i) = -\sum_{j=1}^{m} p(\beta_j/\alpha_i) \cdot \log p(\beta_j/\alpha_i).$$

L'entropie totale du système Y est donc :

11. Le signe $\subseteq$ signifie ici que l'ensemble des variables constituant le vecteur A est inclus dans l'ensemble des variables constituant le vecteur B.

$$
\begin{aligned}
H(Y/X) &= \sum_{i=1}^{n} p_{i.} \cdot H(Y/\alpha_i) \\
&= -\sum_{i=1}^{n} p_{i.} \cdot \sum_{j=1}^{m} p(\beta_j/\alpha_i) \cdot \log p(\beta_j/\alpha_i) \\
H(Y/X) &= -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{i.} \cdot p(\beta_j/\alpha_i) \cdot \log p(\beta_j/\alpha_i).
\end{aligned}
$$

On note aussi :

$$
\begin{aligned}
H(Y/X) &= -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{i.} \cdot p_{j/i} \cdot \log p_{j/i} \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \cdot \log p_{j/i}
\end{aligned}
$$

avec $p_{j/i} = p(\beta_j/\alpha_i)$

Remarque:

En vertu de la définition de la probabilité conditionnelle ($p(X,Y) = p(X) \cdot p(Y/X)$), on démontre que : $H(Y/X) = H(X,Y) - H(X)$

L'entropie conditionnelle est particulièrement intéressante dans le cas du problème d'explication. En effet $H(Y/X)$ peut être interprétée comme la quantité d'information qu'il reste à connaître (l'incertitude) sur Y, une fois que l'on possède déjà l'information fournie par X. Cette quantité représente l'entropie de Y non expliquée par X. Trois cas distincts sont alors susceptibles d'apparaître :

- **1er cas :** $H(Y/X) = H(Y)$: X et Y sont indépendants statistiquement. Aucune relation entre ces grandeurs ne pourra être mise en évidence.
- **2e cas :** $H(Y/X) = 0$: X et Y sont totalement liées et il existe une variable $\tilde{X} \succcurlyeq X$ (au sens large) telle que $Y = f(\tilde{X})$. On peut alors expliquer toutes les modalités de Y, sur tout le domaine M_X.
- **3e cas :** $0 < H(Y/X) < H(Y)$: X n'explique que partiellement Y. Nous distinguons alors deux causes possibles :
  - l'absence d'explication à tort : le modèle déterministe $Y = f(X)$ est représentatif du système, mais les données relevées sont biaisées. Ceci peut être dû à la présence d'un bruit important sur les mesures effectuées ou/et au mauvais fonctionnement d'un sous-système.
  - l'absence d'explication à raison : certaines variables explicatives, pertinentes pour l'étude de notre système, ont été omises, ou le modèle est suffisamment compliqué pour ne pas être déterministe.

3.4 Entropie et information

Nous avons vu que l'entropie d'une variable est la mesure de son degré d'incertitude. Il semble naturel de dire que l'obtention d'informations sur cette variable diminue son incertitude. La quantité d'information obtenue sur cette variable peut donc être mesurée par la diminution de son entropie.

Prenons le cas particulier d'une variable X_i ayant une entropie *a priori* $H(X_i)$. Si après obtention d'information, cette variable est totalement connue, alors son entropie s'annule. Notons $I(X_i : X_i)$ l'information obtenue avec la détermination de X_i :

$$\begin{aligned} I(X_i : X_i) &= H(X_i) - 0 \\ &= H(X_i) \end{aligned}$$

Ce que nous venons d'énoncer pour la variable X_i peut être généralisé pour tout $S \in \mathcal{P}(\Sigma)$, où $\mathcal{P}(\Sigma)$ est l'ensemble de tous les vecteurs possibles dont les composantes sont des composantes élémentaires de Σ.

Ainsi, $H(S)$ représente une mesure de l'information nécessaire pour connaître S.

Si l'on considère maintenant deux variables X et Y, l'inégalité $H(X,Y) \leq H(X) + H(Y)$ peut être interprétée par le fait que, dans l'expression $H(X) + H(Y)$, on compte deux fois une partie de l'information qui se trouve aussi bien dans X que dans Y.

Afin de mesurer cette quantité d'information, nous définissons la *transinformation interne* [Ash65b] et la *transinformation externe.*

3.4.1 Transinformation interne

Par définition :

$$\begin{aligned} \overset{\circ}{I}: \quad \mathcal{P}(\Sigma) &\longrightarrow \mathbb{R} \\ S &\longmapsto \overset{\circ}{I} = \sum_{X \in S} H(X) - H(S) \end{aligned}$$

$\overset{\circ}{I}$ mesure la quantité d'information qui est transmise dans un groupe de variables.

3.4.2 Transinformation externe

Par définition :

$$\begin{aligned} \overset{\leftrightarrow}{I}: \quad P(\mathcal{P}(\Sigma)) &\longrightarrow \mathbb{R} \\ R = \{R_1, R_2, \ldots, R_r\} &\longmapsto \overset{\leftrightarrow}{I} = \sum_{i=1}^{r} H(R_i) - H(\bigcup_{i=1}^{r} R_i) \end{aligned}$$

$\overset{\leftrightarrow}{I}$ mesure la quantité d'information qui est transmise entre plusieurs groupes de variables, où $P(E)$ représente l'ensemble des parties de l'ensemble E.

De nombreuses propriétés intéressantes ont été démontrées [Sha49a, Sha49b, Mil63, Ash65a, Tor82] et M. Sbaï en fait une synthèse [Sba83].

3.5 Détermination de quelques indices intéressants

L'incohérence des données recueillies sur un système complexe implique un manque d'explication de Y par X (où X et Y représentent des variables vectorielles). En effet, nous ne pourrons pas trouver une application f telle que $Y = f(X)$. En pratique, nous chercherons à l'approcher ($Y \approx f(X)$).

Nous déterminerons alors quelques indices mesurant l'*explicabilité* (ou la *modélisabilité*) de Y par X :

Soit $S \in \mathcal{P}(X)$:

- $\overset{\leftrightarrow}{I}(Y : S) = H(Y) + H(S) - H(Y,S) = H(Y) - H(Y/S)$ est l'entropie de Y expliquée par S ;

- $H(Y/S) = H(Y,S) - H(S)$ est l'entropie de Y non expliquée par S ;

- $\%H(Y/S) = \frac{H(Y/S)}{H(Y)}$ est le pourcentage de l'entropie de Y non expliquée par S.

Si l'on remplace S par X dans les trois expressions ci-dessus, nous obtenons :

- $H(Y/X) = H(Y,X) - H(X)$ est la partie non explicable de Y ;

- $\overset{\leftrightarrow}{I}(Y : X) = H(Y) - H(Y/X)$ est la partie explicable de Y ;

- $\%H(Y/X) = \frac{H(Y/X)}{H(Y)}$ est le pourcentage non explicable de Y ;

- $1 - \%H(Y/X) = \frac{\overset{\leftrightarrow}{I}(X;Y)}{H(Y)}$ est le pourcentage explicable de Y.

D'autres indices peuvent également être définis :

- $\begin{array}{rcll} \%ENPE(Y/S) & = & \frac{H(Y/S)-H(Y/X)}{\overset{\leftrightarrow}{I}(Y:X)} & \text{si } \overset{\leftrightarrow}{I}(X:Y) > 0 \\ & = & 0 & \text{si } \overset{\leftrightarrow}{I}(X:Y) = 0 \end{array}$ représente le pourcentage encore non expliqué par S de la partie explicable de Y.

- $m(Y/S) = 1 - \frac{H(Y/S)}{H(Y)} = \frac{H(Y)+H(S)-H(Y,S)}{H(Y)}$

  Cet indice a été proposé par CONANT [Con72] et mesure la modélisabilité de Y par S et possède la propriété d'être compris entre 0 et 1 (0 : Y non modélisable par S et 1 : Y modélisable totalement par S).

3.6 Application au problème d'explication

3.6.1 Le modèle atomique

Au chapitre 2, nous avons montré que chercher $Y = f(X)$ était équivalent à rechercher $\tilde{X}$ ($\tilde{X}$ plus grosse que X) se rapprochant le plus possible de Y en termes de partitions induites sur Ω.

C'est sur cette idée que M. BARBOUCHA [Bar87] propose une approche « système expert », qui consiste à générer à partir du tableau de contingence, un certain nombre de règles permettant de décrire au mieux le fonctionnement du processus.

Ces règles sont de la forme :

$$X \in A \Longrightarrow Y \in B$$

où X est un vecteur de variables explicatives et Y est la variable à expliquer.

A est un sous-ensemble de M_X, B est un sous-ensemble de M_Y.

Si l'on est en présence d'une population d'apprentissage cohérente, (une seule valeur non nulle par ligne dans le tableau de contingence), on peut trouver une relation du type $Y = f(X)$ et dans ce cas, B se réduirait à un singleton. C'est généralement faux car on traite souvent de systèmes complexes, et il y a la plupart du temps absence d'explication à tort (bruits, erreurs de mesure, ...) ou absence d'explication à raison (il manque des variables explicatives).

La non cohérence de la population d'apprentissage se traduit par le tableau de contingence P_{IJ} défini au chapitre 1.

Rappelons simplement que :

- X et Y sont deux variables du système. Elles peuvent être ou non vectorielles ;
- $M_X = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ et $M_Y = \{\beta_1, \beta_2, \ldots, \beta_m\}$ sont respectivement les ensembles de modalités potentielles de X et de Y ;
- p_{ij} est la probabilité d'occurence conjointe des modalités α_i et β_j ;
- $p(\alpha_i) = p_{i.} = \sum_j p_{ij}$ pour $i = 1, \ldots, n$;
- $p(\beta_j) = p_{.j} = \sum_i p_{ij}$ pour $j = 1, \ldots, m$.

On note aussi les probabilités conditionnelles :

$$p(\alpha_i/\beta_j) = p_{i/j} = \frac{p_{ij}}{p_{.j}}$$

$$p(\beta_j/\alpha_i) = p_{j/i} = \frac{p_{ij}}{p_{i.}}$$

On note:

$$B(\alpha_i) = \{\beta_j \in M_Y | p_{j/i} > 0\}$$
$$A(\beta_j) = \{\alpha_i \in M_X | p_{i/j} > 0\}$$

Sur la base de ces notations et du tableau de contingence, on peut déduire des règles de fonctionnement sous la forme:

$$X = \alpha_i \Longrightarrow Y = B(\alpha_i) \quad \text{pour } i = 1, \ldots, n$$
$$Y = \beta_j \Longrightarrow X = A(\beta_j) \quad \text{pour } j = 1, \ldots, m$$

Ceci nous donne un modèle du système composé de $n + m$ règles qui constituent le modèle atomique.

3.6.2 Qualité du modèle

On dit [SBS92] que ce modèle est:

- *VRAI*: car il ne modifie pas la réalité observée (tableau de contingence);
- *FAUX*: car il décrit le couple (système-bruit);
- *PAS SIMPLE*: chaque modalité de X fournit une règle;
- *PAS EFFICACE*: $Y = B(\alpha_i)$ ne détermine pas Y car $B(\alpha_i)$ comporte plusieurs modalités résultant éventuellement d'erreurs.

On peut donc affiner ce modèle en associant à chacune des modalités de la variable à expliquer un coefficient de pondération de la façon suivante:

$$X = \alpha_i \Longrightarrow (Y = B(\alpha_i), P(Y/\alpha_i)) \quad i = 1, \ldots, n$$

où $P(Y/\alpha_i)$ est la distribution de probabilité $p_{j/i}$. Cela ne modifie en rien les caractéristiques du modèle (vrai, faux, non simple, non efficace).

3.6.3 L'opérateur de contraste

Pour augmenter l'efficacité de ces règles, M. BARBOUCHA [BSA87, SB91] considère que le signal est composé d'un signal « pur » (comportement déterministe) sur lequel vient s'ajouter un comportement aléatoire dû aux bruits... Il propose de procéder à un filtrage qui consiste à garder le comportement déterministe (ou considéré comme tel) et à éliminer le comportement aléatoire. Pour chaque règle, il divise ainsi l'ensemble $B(\alpha_i)$ en deux classes:

- B_{σ_i} = les modalités de Y considérées comme la résultante du comportement déterministe;

- $\overline{B_{\sigma_i}}$ = les résidus avec $B_{\sigma_i}(\alpha_i) \cup \overline{B_{\sigma_i}}(\alpha_i) = B(\alpha_i)$.

dans lequel σ_i représente un niveau de contraste de la règle i :

pour chaque règle, on ne conserve que les σ_i modalités les plus significatives de la distribution conditionnelle $P(Y/X = \alpha_i)$.

Ce filtrage fait donc apparaître un seuil que l'utilisateur se fixe *a priori.*

L'opération de contraste entraîne alors la détermination de deux nouvelles distributions :

- $R_{\sigma_i} = R(Y/X = \alpha_i)$, distribution de probabilité de B_{σ_i}
- $Q_{\sigma_i} = Q(Y/X = \alpha_i)$, distribution de probabilité de $\overline{B_{\sigma_i}}$.

Le remplacement de P par R_{σ_i} est d'autant plus légitime que R_{σ_i} est proche de P. On s'intéresse alors à la quantité d'information perdue en remplaçant P par R_{σ_i}, et on la note $d(P, R_{\sigma_i})$.

En utilisant la théorie de l'information, on sait que $d(P, R_{\sigma_i})$ est la variation de l'entropie de SHANNON :

$$d(P, R_{\sigma_i}) = H(P) - H(R_{\sigma_i}).$$

Et on montre qu'en minimisant $d(P, R_{\sigma_i})$ suivant différents critères, on obtient la distribution optimale de R_{σ_i} :

$$\begin{aligned} R^*(Y/X = \alpha_i) &= r_{j/i} = \frac{p_{j/i}}{\sum_{k=1}^{\sigma_i} p_{k/i}} && j = 1, \ldots, \sigma_i \\ &= 0 && j = \sigma_i + 1, \ldots, m_i \end{aligned}$$

avec $m_i = \text{card}B(\alpha_i)$ et en ayant pris soin de reclasser les β_j dans l'ordre décroissant des $p_{j/i}$.

On obtient de la même façon la distribution de Q_{σ_i} :

$$\begin{aligned} Q(Y/X = \alpha_i) &= 0 && j = 1, \ldots, \sigma_i \\ &= q_{j/i} = \frac{p_{j/i}}{\sum_{k=\sigma_i+1}^{m_i} p_{k/i}} && j = \sigma_i + 1, \ldots, m_i \end{aligned}$$

Les règles obtenues à partir de ces distributions sont du type :

$$X = \alpha_i \Longrightarrow (Y = B_{\sigma_i}(\alpha_i), R(Y/X = \alpha_i)) \quad \alpha_i \in M_X.$$

- Un des critères permettant de juger une règle de niveau de contraste σ_i est la probabilité conditionnelle :

  La probabilité après contraste pour que la règle soit vraie dans la population d'apprentissage Ω est :

$$\sum_{j=1}^{\sigma_i} p_{j/i}$$

et la quantité $\sum_{j=\sigma_i+1}^{m_i} p_{j/i}$ représente la probabilité d'erreur.

- On peut aussi mesurer l'efficacité d'une règle.

  Une règle doit être d'efficacité maximale lorsqu'elle peut se traduire par:

  $$X = \alpha_i \Longrightarrow Y = \beta_j$$

  et dans ce cas $E(1) = 1$ (efficacité maximale).

  Le coefficient d'efficacité doit être une fonction décroissante de σ_i.

  Un exemple d'indice d'efficacité est donné par:

  $$E(\sigma_i) = 1 - \frac{H(R)}{\log(m_i)}$$

  Remarque :

  - pour une règle $X = \alpha_i \Longrightarrow Y = \beta_j$, $E = 1$;
  - pour une règle $X = \alpha_i \Longrightarrow Y = \beta_1 \vee \beta_2 \vee \ldots \vee \beta_{m_i}$ où les β_j sont équiprobables, alors $E = 0$ (car $H(R) = \log(m_i)$).

- On peut également mesurer l'utilité de l'information perdue: $U(\sigma_i)$

  Ce coefficient doit vérifier les conditions:

  - fonction décroissante de $H(Q)$;
  - fonction croissante des poids des résidus;
  - fonction décroissante de σ_i.

  À titre d'exemple, M. BARBOUCHA propose:

  $$U(\sigma_i) = \sum_{j=\sigma_i+1}^{m_i} p_{j/i} \cdot [1 + \frac{(\log(m_i - \sigma_i) - H(Q))}{\log(m_i)}]$$

- On peut enfin mesurer le coefficient de vérité $V(\sigma_i)$ de chaque règle qui doit vérifier les propriétés suivantes:

  - inversement proportionnel aux poids des résidus;
  - à poids des résidus égaux, plus la distribution est uniforme, plus le modèle est vrai;
  - $V(\sigma_i)max = 1$ si $U(\sigma_i) = 0$.

  M. BARBOUCHA propose de prendre: $V(\sigma_i) = 1 - U(\sigma_i)$

  $$V(\sigma_i) = \sum_{j=1}^{\sigma_i} p_{j/i} - \sum_{j=\sigma_i+1}^{m_i} p_{j/i} \cdot [\frac{(\log(m_i - \sigma_i) - H(Q))}{\log m_i}]$$

3.6.4 Simplification du modèle

Lorsque n (cardM_X) est grand, le nombre de règles atomiques est trop important et le modèle devient trop lourd à gérer. Une simplification du modèle atomique est alors souhaitable et celle-ci passe par une phase de généralisation du modèle. Cette généralisation passe par une phase d'agrégation des règles ayant des conclusions identiques ou très proches, et qui se traduit par un gain de vérité et une perte d'efficacité du modèle.

Basés sur $H(Y/X)$ et $H(Y/S)$, deux indices de « modélisabilité » sont utilisés :

$$m(Y/X) = 1 - \frac{H(Y/X)}{H(Y)}$$

mesure la *modélisabilité* de Y par X en utilisant les données d'apprentissage. C'est une mesure de leur incohérence [Tor82];

$$q(Y/S) = \frac{H(Y) - H(Y/S)}{H(Y) - H(Y/X)}$$

mesure la *qualité* du modèle plus simple $Y = \tilde{f}(S)$ par rapport à la qualité du modèle $Y = f(X)$.

Dans le cadre de cette généralisation [PSS92], nous utiliserons pour notre part des méthodes d'induction par arbre de décision.

3.7 Conclusion sur l'utilisation des outils de la théorie de l'information dans le cadre de l'analyse structurale des systèmes

Bien que la théorie de l'information soit relativement ancienne, elle suscite toujours autant d'intérêt sur le plan scientifique. Nous nous proposons de l'utiliser dans le cadre de notre travail, c'est-à-dire dans le cadre de l'analyse structurale des systèmes, et plus particulièrement dans le cadre de l'aide au diagnostic.

Après avoir introduit l'entropie de Shannon monovariable, la généralisation à plusieurs variables a été entreprise ainsi qu'une définition de l'entropie conditionnelle représentant la quantité d'information qu'il nous faudrait pour connaître complètement l'état d'une variable (Y avec nos notations) ayant pris connaissance de l'état d'autres variables (appelées X).

Appliquant les outils de la théorie de l'information au problème de la recherche d'explication et sur la base d'un tableau de contingence, M. BARBOUCHA propose d'établir un modèle atomique permettant de prendre en compte le comportement déterministe du système par l'intermédiaire d'un opérateur de contraste. Ce modèle étant trop lourd à gérer, on se propose de le simplifier. Un bon critère probabiliste dans le cadre de la simplification de ce modèle semble être l'entropie conditionnelle qui permet de définir un indice mesurant la modélisabilité ainsi qu'un indice de qualité de modèle relativement simple.

Deux intérêts principaux semblent ressortir de l'utilisation de la théorie de l'information dans le cadre de l'étude d'un système :

Le premier est que nous pouvons considérer à la fois des variables quantitatives, mais également des variables qualitatives (ordonnées ou pas). Il suffit dès lors de considérer la contrainte d'ordre sur les modalités de la (ou des) variable(s) quantitative(s) considérée(s).

Le second intérêt est qu'aucune relation mathématique entre les variables n'est considérée. Seules des relations à base de règles sont considérées dans notre travail, ce qui nous permet de traiter à la fois des relations (supposées) linéaires ainsi que des relations (supposées) non linéaires.

Le critère étant défini, il nous reste à définir la façon de construire un modèle simple sur la base du modèle atomique. C'est ce que nous nous proposons d'établir dans le chapitre 4.

Chapitre 4

Construction d'un modèle de comportement par arbres de décision

4.1 Introduction

Dans une démarche d'explication, le *modèle atomique* permet de rendre compte du comportement déterministe du processus (que l'on cherche à modéliser), en filtrant le comportement aléatoire dû aux perturbations et aux bruits polluant les données (filtrage du tableau de contingence [SB91]). Lorsque le nombre de modalités de X (cardM_X) est important, ce modèle est inutilisable.

Un modèle plus simple pourra être trouvé en généralisant le modèle atomique.

Dans le cadre de cette simplification du modèle atomique, nous nous sommes intéressés au système *ID3* développé par J.R. QUINLAN [Qui83, Qui86, Qui93]. C'est un système d'inférence inductive à partir d'exemples. Et plus particulièrement, c'est une méthode d'apprentissage par arbre de décision (cf [HMS66]).

Nous nous sommes aperçus que cet algorithme n'était qu'un cas particulier d'algorithme s'inscrivant dans une démarche plus générale de construction des arbres de décision.

Dans ce cadre, nous examinerons les critères utilisables afin de construire des arbres de décision en soulignant les inconvénients mais également les avantages de chacun d'eux.

Dès lors, deux approches nous permettent de construire des arbres de décision, à savoir :

- une approche utilisant toute la population d'apprentissage. On parlera alors « d'approche globale » vis-à-vis de la population d'apprentissage. À un niveau de l'arbre de décision, on ne considérera qu'une seule variable (on parlera dès lors de construction des arbres de décision « par niveau » ou de discrimination des conclusions « axe par axe »).

- une approche plus locale permettant une spécialisation locale sur un sous-ensemble de la population d'apprentissage non encore discriminé. On parlera alors « d'approche locale » de construction de l'arbre, encore appelée dans ce travail construction de l'arbre « par nœud ».

De plus, l'approche permettant de construire les arbres de décision est une approche *descendante*, et il est préférable dans certains cas d'utiliser une démarche *ascendante* préliminaire de sélection des variables pertinentes à tester.

Enfin, les deux points de vue précédents peuvent être associés à une démarche dite *agrégative* dans laquelle l'algorithme agrège les variables trouvées, mais également à une démarche dite *désagrégative* dans laquelle l'algorithme désagrège petit à petit l'ensemble des variables supposées pertinentes.

La raison de ces différentes méthodes est la suivante: il peut arriver que le pouvoir explicatif de chaque variable d'un ensemble S soit faible, alors que celui de l'ensemble S considéré globalement est important. Dans ce cas, un algorithme utilisant un critère prenant en compte chaque variable séparément, (ou ne prenant pas en compte tout le pouvoir explicatif du vecteur S), ne permettra pas de mettre en évidence certaines relations existant entre ces variables, et donc ne les choisira pas comme variables discriminantes.

L'intérêt d'un algorithme *agrégatif* ou à plus forte raison *désagrégatif* (on calcule d'abord le critère sur l'ensemble des variables constituant X, puis au fur et à mesure de la progression dans l'algorithme on désagrège cet ensemble) sera mis en évidence: cette démarche permet d'écarter des variables faussement informatives, et de conserver les variables réellement pertinentes.

Dans ce chapitre, ces trois points de vue de construction des arbres de décision seront développés. La combinaison de ceux-ci nous permettra de proposer des algorithmes de plus en plus performants.

Par ailleurs, ces modèles sont construits de manière à expliquer toutes les modalités de Y *via* un critère utilisé de façon à discriminer l'ensemble des modalités de Y. Le critère consiste donc toujours à « moyenner » le pouvoir discriminant sur toutes les modalités de Y. Mais il est possible d'utiliser ce critère de façon plus locale sur Y, afin d'expliquer une modalité particulière β_j de Y « *contre toutes les autres* ». En d'autres termes, pour chaque modalité de Y (pour chaque mode de fonctionnement dans le cas d'un diagnostic) nous chercherons une description (*une fonction de reconnaissance*) de celle-ci la plus correcte possible:

$\forall \beta_j \in M_Y$ nous cherchons f_j sous la forme de règles | $\beta_j \approx f_j(S)$ où $S \subset \tilde{X}$. Ce qui revient à chercher une fonction de reconnaissance de la modalité β_j de Y ne couvrant pas l'ensemble de toutes les autres modalités. Le modèle considéré pourra dès lors conclure β_j ou $\overline{\beta_j}$. Notre démarche générale consistant à travailler sur $\{\beta_1, \beta_2, ..., \beta_m\}$ s'applique donc

bien à ce cas en considérant : $Y = \{\beta_j, \overline{\beta_j}\}$, pour lequel :

$$\overline{\beta_j} = \bigcup_{\substack{k=1 \\ k \neq j}}^{m} \beta_k$$

Afin d'illustrer ce propos, dans le cadre de la détection (voir l'introduction générale), nous pouvons décider entre deux hypothèses de travail, à savoir H_0 (*fonctionnement normal*) et H_1 (*fonctionnement anormal du système*). Si plusieurs états de dysfonctionnement apparaissent, le problème de détection consiste à discriminer le mode de fonctionnement normal contre tous les autres modes de dysfonctionnement.

4.2 Démarche générale de construction des arbres de décision

Formellement, un arbre de décision est un arbre [Pic72] tel que :

- une *feuille* (ou nœud réponse) contient un nom de classe ;
- un *nœud* (qui n'est pas une feuille, ou nœud décision) contient un test sur un attribut (une variable) avec une branche donnant naissance à un autre arbre de décision pour chaque valeur possible de l'attribut en question.

Le cas de variables numériques sera également traité dans le paragraphe suivant.

Dans un premier temps, nous nous proposons de définir le critère utilisé dans nos approches. Puis, nous exposerons trois points de vue qui seront combinés afin de construire des arbres de décision. Enfin, nous déterminerons les critères d'arrêt (les critères de sortie) de nos algorithmes.

4.2.1 Les critères utilisés afin de construire des arbres de décision

4.2.1.1 Les critères utilisés dans la littérature

Les systèmes d'induction par arbres de décision visent à l'optimisation d'un critère global afin de spécialiser les hypothèses [Ren86]. La démarche est la suivante :

On recherche la variable apportant le plus d'information sur les classes à expliquer, puis une deuxième prise parmi les variables restantes, puis une troisième, ...

Plusieurs critères peuvent être utilisés pour définir, à chaque étape, la variable supplémentaire à prendre en compte [Min89].

Les indices principaux sont :

- la mesure du χ^2, utilisée pour la première fois dans des algorithmes d'induction par [Har84] ; cette mesure représente l'écart à l'indépendance statistique entre les lignes et les colonnes :

$$\chi^2 = \sum_i \sum_j \frac{(p_{ij} - p_{i.} \cdot p_{.j})^2}{(p_{i.} \cdot p_{.j})}$$

  la variable apportant le plus d'information sera alors celle qui maximisera la mesure du χ^2 ;

- l'indice GINI (ou mesure de l'impureté), proposé par [BFOS84] ;

$$i = \sum_i \sum_j \frac{p_{ij}^2}{p_{i.}} - \sum_j (p_{.j})^2$$

  la variable apportant le plus d'information sera alors celle qui maximisera cet indice ;

- la mesure d'information de J.R. QUINLAN : elle est utilisée dans le système *ID3*, et dans de nombreuses variantes de celui-ci.

Dans ce travail, nous nous limiterons à l'étude du critère utilisé par *ID3*, qui est en fait une mesure issue de la théorie de l'information présentée au chapitre 3.

4.2.1.2 Description du critère utilisé par *ID3*

Pour simplifier le problème, on ne considère que deux classes notées P et N. P et N représentent respectivement l'ensemble des instances positives ($p = \text{card}P$) et l'ensemble des instances négatives ($n = \text{card}N$).

L'information moyenne apportée par P et N est :

$$I(p,n) = -\frac{p}{p+n} \log \frac{p}{p+n} - \frac{n}{p+n} \log \frac{n}{p+n}$$

Cette quantité représente l'entropie d'une variable à expliquer Y dont l'ensemble des modalités est séparé en deux classes notées N et P.

L'information moyenne apportée par l'arbre ayant x (un attribut ayant les modalités $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$) comme racine est :

$$E(x) = \sum_{i=1}^{r} \frac{p_i + n_i}{p+n} \cdot I(p_i, n_i)$$

Cette quantité représente l'entropie conditionnelle $H(Y/x)$.

J.R. QUINLAN définit alors le gain d'information qu'apporte l'attribut x par :

$$Gain(x) = I(p,n) - E(x).$$

qui représente la transinformation externe $\overleftrightarrow{I}(x : Y) = H(Y) - H(Y/x)$.

Le système *ID3* examine alors tous les candidats et choisit x qui maximise le $Gain(x)$, forme l'arbre à partir de celui-ci, et utilise le même processus récursivement (pour former l'arbre de décision).

Comme le dit J.R. QUINLAN [Qui86], maximiser le $Gain(x)$ revient à minimiser $E(x)$, car $I(p, n)$ est constant, quel que soit l'attribut x. Ceci revient, dans le cadre de la théorie de l'information et de l'entropie, à minimiser l'entropie conditionnelle $H(Y/X)$ où Y est une variable à expliquer, et X est un vecteur de variables explicatives.

Le choix de la maximisation du gain comme critère favorise les variables ayant beaucoup de modalités. En effet, dans le cas d'une variable qui prend autant de valeurs différentes qu'il y a d'individus dans le tableau de données, le gain est maximal ($H(Y/x) = 0$). QUINLAN propose donc de diviser $Gain(x)$ par ce qu'il appelle $SplitInfo(x)$ qui n'est autre que l'entropie $H(x)$. La quantité résultante est appelée $GainRatio(x)$ qu'il maximise.

$$GainRatio(x) = \frac{Gain(x)}{SplitInfo(x)} = \frac{H(Y) - H(Y/x)}{H(x)}$$

L'algorithme *C4.5* [Qui93] repose sur la même stratégie que *ID3*. Le principal apport de cet algorithme par rapport au précédent concerne les variables continues. Il procède à une discrétisation progressive de ces variables en fixant, à chaque nœud de l'arbre, un seuil pour chaque variable continue (cette procédure de seuillage est détaillée et améliorée dans [FI92]). Ces variables sont donc transformées en variables binaires, et peuvent dès lors être traitées comme des valeurs discrètes. Ce seuillage est effectué à chaque nœud de l'arbre. De plus, cet algorithme offre la possibilité de prendre en compte les valeurs manquantes des attributs, et permet également d'effectuer un élagage de l'arbre ainsi que des validations croisées.

Cependant, l'ajustement des seuils des variables continues conduit de nouveau à favoriser le choix des variables continues et/ou qui comportent beaucoup de valeurs différentes dans la population d'apprentissage, car cet ajustement conduit à une amélioration du gain pour les variables continues, ce qui n'est pas effectué pour les variables discrètes. Un moyen de remédier à ce problème, est proposé dans [Qui96], consistant à appliquer une pénalité aux tests continus potentiels, basée sur le principe MDL.

D'autre part, une récente proposition consiste à modéliser la construction des arbres de décision sous la forme d'un processus itératif de décision bayesienne [MS98]. Enfin, d'autres auteurs proposent une construction incrémentale des arbres de décision afin de faire évoluer la structure de l'arbre pendant l'apprentissage, voire la validation de celui-ci. Ainsi, UTGOFF présente dans [Utg95] deux approches basées sur la construction d'arbres de décision, le principe général étant de garder l'information sur la population d'apprentissage dans sa globalité à chaque nœud, ce qui permet de restructurer plus facilement l'arbre lors de l'ajout d'une nouvelle observation.

Remarques :

- Dans le cas où la population d'apprentissage est incohérente, à un nœud terminal (une feuille) correspondent plusieurs classes (plusieurs conclusions).
- *ID3* introduit les variables une à une en utilisant une démarche descendante de sélection des variables pertinentes et de construction de l'arbre de décision. Cependant V.M. TORO [Tor82] montre que :

$$gain(x_1, x_2, \ldots, x_p) \neq \sum_{i=1}^{p} gain(x_i)$$

Dans ce sens, il peut arriver que le pouvoir explicatif de chaque variable d'un vecteur S soit faible (elles ne seront alors pas retenues comme variables discriminantes à chaque pas de l'algorithme utilisant une approche descendante) alors que le pouvoir explicatif du vecteur S considéré globalement est important. La démarche définie par *ID3* sera alors incapable de mettre en évidence une relation de ce type entre la variable à expliquer et le vecteur de variables explicatives S.

À titre d'exemple, considérons le tableau 4.1.

x_1	x_2	x_3	Y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

TAB. 4.1: *Un exemple de tableau initial de données pour lequel Y représente le « bit de parité » de x_1 et de x_2*

et cherchons à expliquer Y.
Le calcul des entropies conditionnelles monovariables nous donne :
$H(Y/x_1) = H(Y/x_2) = 1$bit.
$H(Y/x_3) = 0.69$bit.
Notre choix va dès lors se porter sur x_3, alors que manifestement, Y est le bit de parité de x_1 et x_2.

Afin de résoudre ce problème, une solution consisterait à chercher le sous-ensemble de variables expliquant au mieux Y :

Dans ce cadre, nous avons $H(Y/(x_1, x_2, x_3)) = 0$, ce qui signifie qu'il existe une relation entre Y et un sous-ensemble (au sens large) de $\{x_1, x_2, x_3\}$.

De même, $H(Y/(x_1, x_2)) = 0$ alors que $H(Y/(x_1, x_3)) = H(Y/(x_2, x_3)) = 0.5$ bit.

Utiliser cette démarche nous permet donc de choisir le couple (x_1, x_2) apportant toute l'information nécessaire à la connaissance de Y.

4.2.1.3 Le critère utilisé dans nos algorithmes

Il est bien évident que n'importe quel critère défini dans le paragraphe précédent peut être utilisé dans nos algorithmes.

Au chapitre 3, nous avons montré que l'indice $H(Y/X)$ permettait de mesurer la liaison entre X et Y.

Nous nous proposons donc tout naturellement d'utiliser le critère consistant à minimiser l'entropie conditionnelle $H(Y/S)$, où S est une variable multidimensionnelle, avec $S \in \mathcal{P}(X)$.

Cette quantité représente en fait la quantité d'information non encore expliquée de Y, après avoir posé les questions concernant les variables composant S. Notons que l'on peut choisir de maximiser la quantité : $\frac{H(Y)-H(Y/S)}{H(Y)}$ qui représente le pourcentage expliqué de Y ($H(Y)$ étant une constante).

On peut également choisir de maximiser l'indice $q(Y/S) = \frac{H(Y)-H(Y/S)}{H(Y)-H(Y/X)}$ défini au chapitre 3, qui représente la qualité du modèle plus simple $Y = \tilde{f}(S)$ par rapport à la qualité du modèle $Y = f(X)$. L'intérêt de cet indice repose essentiellement sur le fait qu'il soit borné entre 0 et 1 et qu'il représente une fonction croissante de la quantité d'information apportée sur Y :

- 0 : S n'explique pas Y ;
- 1 : S apporte autant d'information sur Y que X ; l'ensemble des variables composant $X \backslash S$ représente l'ensemble des variables redondantes pour expliquer Y, ou n'apportant aucune information sur Y.

4.2.2 Approches globale et locale du problème

Le critère d'étude étant défini, dans le cadre des méthodes d'induction par arbres de décision, nous pouvons distinguer deux approches du problème de l'apprentissage :

- une approche dite « globale » ;
- et une approche dite « locale ».

4.2.2.1 L'approche globale

Cette approche consiste en l'utilisation du critère sur l'ensemble d'apprentissage dans sa globalité : à chaque niveau de l'arbre de décision correspond une et une seule variable répondant au critère. Deux façons de traiter le problème peuvent être utilisées :

- Nous pouvons traiter le problème « variable par variable » sans effet mémoire : à chaque niveau de l'arbre, la variable qui est retenue est celle qui répond au critère choisi, sans tenir compte des niveaux supérieurs (des variables testées précédemment). Il suffit alors de classer une fois pour toutes les variables par quantité d'information

apportée sur la conclusion, en commençant par la variable apportant le plus d'information. Il est clair que cette méthode ne sera pas optimale car l'information traitée à un niveau de l'arbre pourra l'être à un autre niveau si les variables concernées possèdent de l'information commune sur Y. Le pouvoir discriminant de l'arbre n'est par conséquent pas optimisé.

- Il semble plus intéressant de traiter le problème en tenant compte des variables déjà testées précédemment. La variable retenue est alors celle qui, en tenant compte des variables déjà testées, répond au critère choisi (et ceci de façon multidimensionnelle). Nous dirons ainsi que nous choisissons une méthode globale (dans le sens d'une utilisation du critère sur l'ensemble d'apprentissage entier) multidimensionnelle.

Jusqu'à présent, l'information a toujours été vue sous une approche « globale » en tenant compte de l'ensemble de la population d'apprentissage: caractérisation de l'information totale (ou moyenne) sur des variables (d'où le symbole « $\sum$ » dans les expressions).

Il serait pourtant intéressant d'étudier l'information de SHANNON partielle (sur quelques modalités), ceci afin de travailler de façon « locale ».

On peut en fait représenter ces deux approches de la façon suivante:

$$H_g : \mathbb{P}(\Omega) \longrightarrow \mathbb{R}^+$$

$$H_l : P(\Omega) \longrightarrow \mathbb{R}^+$$

En effet, l'entropie globale H_g nous permet de travailler sur des partitions de l'ensemble d'apprentissage plus ou moins fines, tandis que l'information locale H_l nous permettra de travailler sur des parties de l'ensemble d'apprentissage plus ou moins petites.

4.2.2.2 L'approche locale

Cette approche consiste en l'utilisation du critère sur une partie de la population d'apprentissage: à chaque nœud de l'arbre de décision correspond une variable, répondant au critère choisi (pris de façon locale) tout en prenant en compte les modalités des variables initialement testées (approche multidimensionnelle). À un niveau donné, plusieurs variables différentes peuvent alors être testées.

Afin de définir cette dernière approche, deux techniques peuvent être employées:

- La première consisterait à définir un chemin allant de la racine jusqu'au nœud considéré, ce qui nous amène dans le cadre du critère choisi à calculer une expression de l'entropie conditionnelle de la forme $H(Y/(X_i = \alpha_i \wedge X_j = \alpha_j \wedge \ldots \wedge X_k = \alpha_k \wedge X_l))$ où X_i est la première variable à tester (qui correspond à la racine de l'arbre) prenant la valeur α_i, X_j est la deuxième variable à tester prenant la valeur α_j, ..., et X_k est la dernière variable à tester prenant la valeur α_k. Ces conditions constituent donc le chemin à parcourir: $X_i = \alpha_i \wedge X_j = \alpha_j \wedge \ldots \wedge X_k = \alpha_k$ jusqu'au nœud considéré.

La nouvelle variable à tester X_l sera la variable pour laquelle l'indice ci-dessus est minimum. Il nous faut dès lors travailler avec des quantités d'information locales et des chemins différents, ce qui demande un conditionnement du critère assez lourd à gérer.

- La seconde approche du problème (qui revient au même que la précédente) consisterait à travailler sur des ensembles d'apprentissage de plus en plus restreints au fur et à mesure que l'on descend dans l'arbre. À chaque nœud, on cherchera alors la variable la plus discriminante (celle qui répond au critère consistant à minimiser $H(Y/X_i)$) dans la sous-population d'apprentissage considérée. C'est bien entendu cette deuxième formulation du problème que nous allons utiliser.

4.2.2.3 Découpage de l'espace des variables

Cette démarche qui consiste à utiliser une approche globale (par niveau) ou locale (par nœud) afin de construire des arbres de décision est illustrée par la figure 4.1.

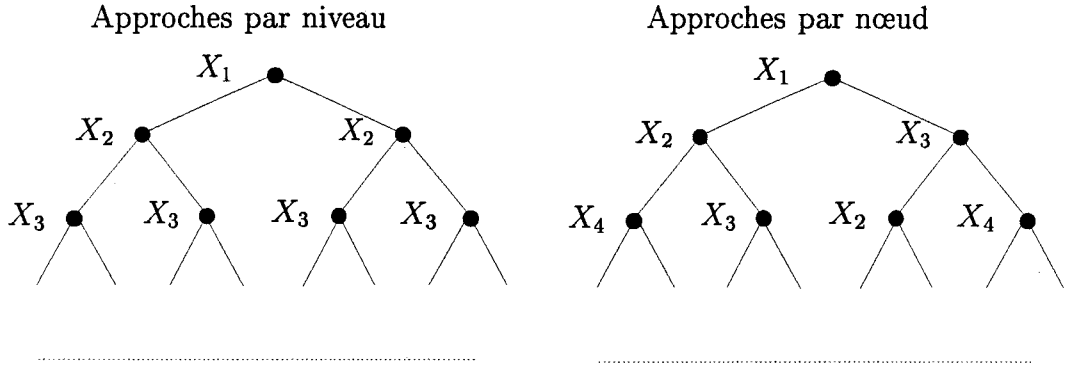

FIG. 4.1: *Approches de construction d'arbres de décision par niveau / par nœud*

En considérant l'approche globale, on trouvera dans l'arbre une seule variable par niveau. Pour cela, on travaille sur une partition de la population d'apprentissage Ω que l'on affine petit à petit jusqu'à ce que les conclusions soient discriminées au mieux (suivant le critère utilisé). On procède donc de manière *globale* sur l'ensemble d'apprentissage.

Par contre, en considérant l'approche locale, on trouvera dans l'arbre une variable par nœud. On travaille en fait sur des parties de Ω. Ces parties sont de plus en plus restreintes au fur et à mesure que l'on progresse dans l'arbre. On procède donc de manière *locale* sur l'ensemble d'apprentissage en utilisant une démarche de spécialisation.

Cette démarche constitue en fait le premier point de vue afin de construire des arbres de décision.

Afin de bien comprendre ces deux approches (globale et locale), considérons l'espace des variables à tester. La figure 4.2 représente le découpage de cet espace par un arbre de

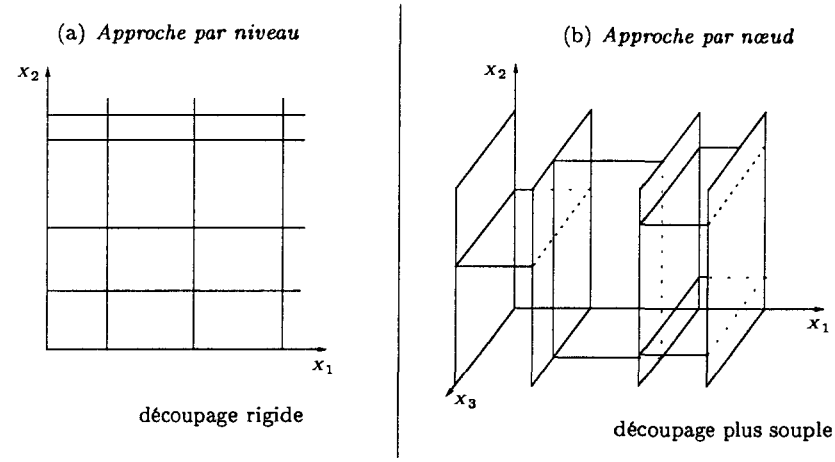

FIG. 4.2: *Découpage de l'espace des variables en considérant une approche par niveau (a) et une approche par nœud (b)*

décision, dans lequel x_1 est la première variable à tester, x_2 la deuxième variable à tester, et x_3, la troisième. Les séparateurs représentent des hyperplans.

La construction de l'arbre par niveau correspond à un découpage « axe par axe » de l'espace des variables, tandis que la construction de l'arbre par nœud correspond à un découpage par spécialisation de la règle aboutissant au nœud considéré.

L'approche locale est beaucoup plus souple que l'approche globale : la variable x_1 étant testée, la variable x_2 est alors testée en prenant en compte la première et la troisième modalité de x_1. Par contre, pour la deuxième modalité de x_1, on pourra préférer tester x_3 qui correspondrait à un choix plus judicieux.

4.2.3 Approches descendante et ascendante de sélection des variables

Dans ce paragraphe nous définirons les méthodes *descendantes* de sélection des variables à tester, en opposition aux méthodes *ascendantes* (figure 4.3) ; constituant ainsi le deuxième point de vue afin de construire des arbres de décision.

Afin de présenter clairement ces concepts, nous appellerons (nous renumérotterons) la première variable trouvée par l'algorithme « X_1 », la deuxième variable trouvée par l'algorithme « X_2 », etc...

Pour procéder au choix des différentes variables discriminantes dans l'arbre, on peut procéder de deux manières.

On peut tout naturellement commencer par chercher la variable « la plus discriminante » au sens du critère choisi et la placer en haut de l'arbre (elle sera renommée « X_1 »). Puis, on cherchera la deuxième variable qui, associée à la première variable, répond au critère choisi. Elle constituera une variable à tester au deuxième niveau de l'arbre et sera renommée « X_2 »... et ainsi de suite jusqu'au bas de l'arbre. Par voie de conséquence, la

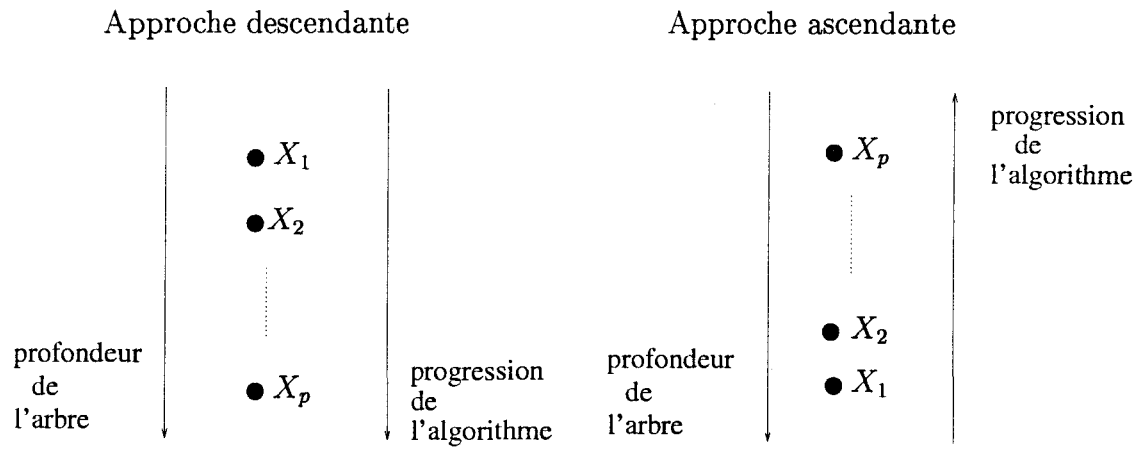

FIG. 4.3: *Approches descendante et ascendante*

phase de sélection des variables à tester ainsi que la phase de construction de l'arbre de décision pourront être simultanées. Sur la figure 4.3, la profondeur de l'arbre peut être représentée par un axe orienté dans le même sens que la progression dans l'algorithme. On parlera alors d'algorithme *descendant*, encore appelé « *model-driven algorithm* » dans le sens où cette approche nécessite l'utilisation d'un modèle qui est ici un arbre de décision.

À l'inverse, on peut commencer par chercher la variable « la moins discriminante » au sens d'un critère choisi (elle sera renommée « X_1 »). Cette variable constituera la dernière variable à tester dans l'arbre. De la même façon on recherchera l'avant-dernière variable à tester (renommée alors « X_2 »), ..., jusqu'à trouver la variable « la plus discriminante » qui constituera la racine de l'arbre (variable renommée « X_p »). La phase permettant de déterminer l'ordre des variables à tester représente ici, *a fortiori*, une phase préliminaire à la phase de construction de l'arbre de décision. Sur la figure 4.3, la profondeur de l'arbre peut être représentée par un axe orienté dans le sens inverse de la progression dans l'algorithme. On parlera ici d'algorithme *ascendant* (ou data-driven). En effet, retirer progressivement les variables non pertinentes (non discriminantes) consiste à généraliser petit à petit la description des données.

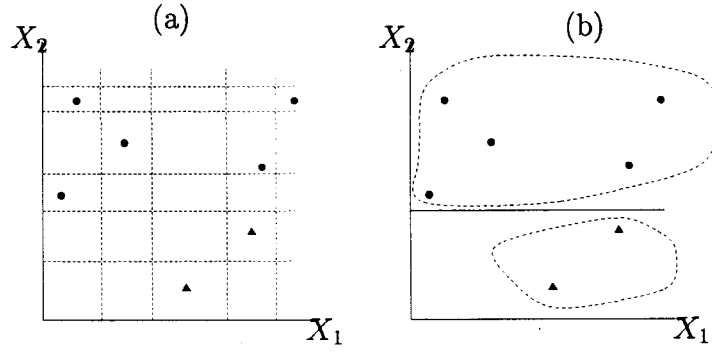

FIG. 4.4: *Approche ascendante, encore appelée « data-driven approach »*

Dans l'exemple de la figure 4.4, en prenant en compte (X_1, X_2), toutes les observations sont discriminées. Le fait de ne considérer que X_2 permet de généraliser les observations. Cette démarche est donc basée sur l'approche « data-driven » présentée au chapitre 2.

Cette approche de construction des arbres de décision constitue le deuxième point de vue.

4.2.4 Approches agrégative et désagrégative de sélection des variables

Nous définissons dans ce paragraphe, le troisième et dernier point de vue afin de construire des arbres de décision. Dans le cadre de la sélection des variables les plus pertinentes, nous distinguons les approches *agrégatives* des approches *désagrégatives* (figure 4.5).

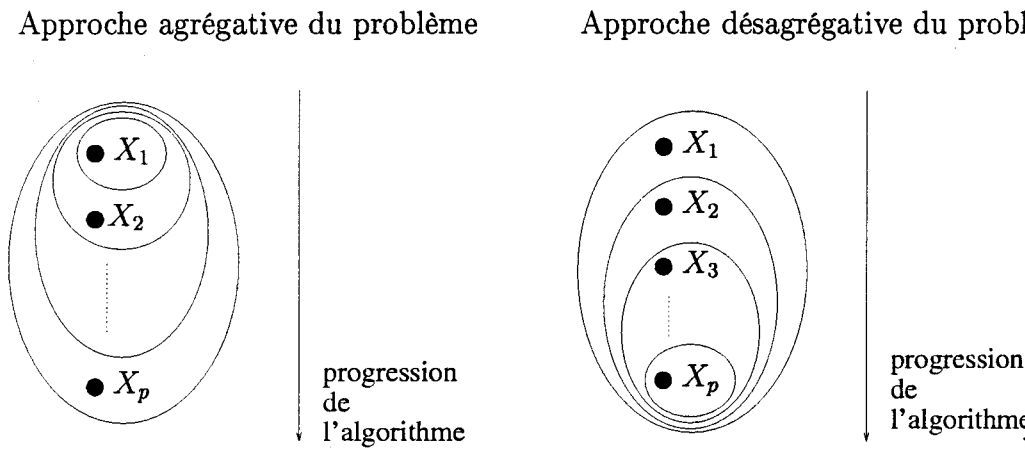

FIG. 4.5: *Approches agrégative et désagrégative du problème*

Afin d'expliciter simplement ce point de vue, nous pouvons scinder le vecteur des variables potentiellement explicatives X en deux vecteurs S et Res représentant respectivement le vecteur des variables considérées pertinentes pour l'explication de Y et le vecteur des variables résiduelles (non prises en compte).

La démarche agrégative de sélection des variables pertinentes consistera alors à chercher la variable la plus discriminante X_i et à l'associer au vecteur S obtenu à l'itération précédente de l'algorithme et par voie de conséquence, à retirer X_i du vecteur Res. Nous agrégeons ainsi petit à petit les variables les plus pertinentes, d'où le nom de cette approche.
Sur la figure 4.5, cette agrégation successive des variables pertinentes constituant S est représentée par des cercles emboîtés.

À l'inverse, la démarche désagrégative de sélection des variables pertinentes consistera à chercher la variable la moins discriminante X_i et à l'associer au vecteur Res obtenu à l'itération précédente de l'algorithme et par voie de conséquence, à retirer X_i du vecteur S constituant le vecteur des variables les plus explicatives. Nous excluons ainsi petit à petit

du vecteur S les variables redondantes ou n'apportant aucune information sur Y, d'où le nom de cette approche.

Sur la figure 4.5, cette désagrégation successive des variables non pertinentes constituant S (*i.e.* cette agrégation successive des variables non pertinentes constituant *Res*) est représentée par des cercles emboîtés.

Nous nous baserons ainsi sur ces trois points de vue, en les combinant de manière à obtenir des algorithmes performants.

4.2.5 Les critères d'arrêt

Les critères d'arrêt de ces approches seront les suivants :

- À un niveau donné, les conclusions peuvent être toutes discriminées (dans le cas d'une population d'apprentissage cohérente). Les nœuds constituants le niveau considéré sont alors appelés des feuilles. À chaque feuille de l'arbre correspond une conclusion et une seule, et dans ce cas on a $H(Y/S) = 0$. Y est complètement expliquée par S.

  Dans le cas d'une population incohérente, il est préférable d'utiliser l'indice $q(Y/S) = \frac{H(Y)-H(Y/S)}{H(Y)-H(Y/X)}$ défini dans le chapitre précédent. On a $q(Y/S) = 1$ si S apporte autant d'information sur Y que X ; c'est-à-dire que toute l'information sur Y contenue dans la population d'apprentissage est résumée par S.

  Pour des raisons qui lui sont propres, l'utilisateur pourra se contenter d'un certain pourcentage d'explication de Y, surtout si les données d'apprentissage sont incohérentes. À titre d'exemple, il pourra se contenter de 95 % d'explication de Y. Il lui faudra alors définir un seuil q_{min} au delà duquel il considère qu'il a expliqué correctement Y, en tenant compte des imperfections (de l'incohérence) de la population d'apprentissage.

- L'utilisateur, pour une raison quelconque (pouvant être par exemple la conséquence d'une contrainte matérielle), pourra choisir un nombre déterminé de variables à tester. À titre d'exemple, il pourra désirer ne tester que les 5 variables les plus discriminantes sur les 100 variables disponibles. On appellera « nb_{max} » ce nombre maximal de variables à tester.

L'utilisateur pourra choisir l'un des deux critères d'arrêt définis ci-dessus, ou éventuellement les deux. S'il ne désire pas utiliser « nb_{max} », il suffit de fixer cette constante à $p = \text{card}(X)$. S'il ne désire pas utiliser q_{min} dans ces algorithmes comme critère d'arrêt, il suffit de le fixer à 100 %.

Ces critères d'arrêt sont bien adaptés aux approches globales par niveau.

Il est tout à fait possible de les employer dans une approche plus locale du problème, et dans ce cas, l'utilisateur pourra choisir l'une des deux démarches suivantes :

- La première démarche consiste à reprendre les critères d'arrêt tels quels en considérant toute la population d'apprentissage et à chaque nœud, la variable retenue sera

celle qui répond au critère de sélection des variables vu de « façon locale » (voir le paragraphe 4.2.1.3) sur la partie de la population d'apprentissage considérée.

- La deuxième démarche consiste à utiliser le critère d'arrêt sur le nœud en question en prenant en compte la partie de la population d'apprentissage considérée au nœud. La variable retenue sera également celle qui répond au critère vu de « façon locale » sur la partie de la population d'apprentissage considérée. En effet, il suffit de reprendre l'indice $q(Y/S)$ conditionné à chaque nœud de l'arbre sur la partie de la population d'apprentissage correspondante et non pas globalement sur la totalité de l'ensemble d'apprentissage. Cette démarche nécessite un conditionnement simple du critère utilisé. On pourra alors se fixer un seuil q_{min} (par exemple 95 %) au delà duquel l'explication est suffisante pour ce nœud. Dans ce cas, on ne poursuit pas la recherche et une feuille est construite. Cet indice est maximal (=1) à ce nœud si aucune variable ne permet de mieux discriminer les conclusions.

Les critères d'arrêt étant définis, nous allons étudier dans un premier temps les approches de construction de l'arbre de décision par niveau, afin de souligner l'intérêt des approches ascendantes désagrégatives. Dans un deuxième temps on affinera certaines de ces méthodes en utilisant des approches de construction par nœud.

Afin d'illustrer les différentes approches développées et dans l'objectif de mettre en évidence les propriétés de chacune d'entre elles, nous proposons un petit exemple de tableau de données sur lequel nous testerons les algorithmes. Nous pourrons ainsi comparer les différents arbres obtenus. Il s'agit d'un système composé de 5 variables explicatives a_1, a_2, a_3, a_4 et a_5, et d'une variable à expliquer Y. Ces données sont générées à partir des variables a_1, a_2, a_3 par l'intermédiaire du circuit combinatoire de la figure 4.6.

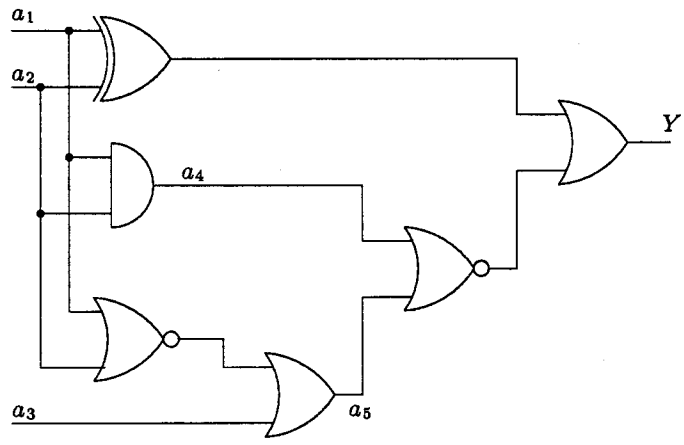

FIG. 4.6: *Circuit combinatoire*

En supposant que a_1, a_2 et a_3 soient des variables binaires générées de manière aléatoire, on obtient le tableau initial des données 4.2.

a_1	a_2	a_3	a_4	a_5	Y
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	0

TAB. 4.2: *Exemple de tableau initial des données*

La particularité de cet exemple est que la variable Y est expliquée complètement par a_1 et a_2. En effet, on remarque que $Y = a_1 \oplus a_2$. Les autres variables apportent de l'information redondante, voire parasite. Ainsi, si l'on calcule les entropies conditionnelles de chaque variable (voir tableau B.1 donné en annexe), on constate que les variables a_1 et a_2 prises séparément, n'apportent aucune information sur Y alors que les variables a_4 et a_5 seules semblent plus informatives. Par contre, si on associe les deux variables a_1 et a_2, alors la variable Y est complètement expliquée.

Dans ces conditions, nous pourrons étudier la capacité des algorithmes à sélectionner les variables les plus discriminantes.

Afin d'être cohérent dans les différentes démarches proposées, nous renommerons « X_1 » la première variable trouvée par l'algorithme, « X_2 » la deuxième variable, etc...

Dans la présentation des algorithmes, nous noterons S le vecteur des variables utilisées pour construire l'arbre de décision et nous noterons *Res* le vecteur des variables restantes.

4.3 Les méthodes de construction par niveau

Les algorithmes de construction de l'arbre de décision par nœud sont en général plus performants que les algorithmes de construction de l'arbre par niveau. Ces derniers sont présentés dans ce travail afin de mettre en évidence certaines propriétés intéressantes des approches ascendantes désagrégatives que nous utiliserons dans les algorithmes de construction par nœud.

4.3.1 Approche triviale (sans effet mémoire)

Pour construire l'arbre de décision, une manière triviale de procéder est d'évaluer le critère de sélection des variables, en considérant une seule variable à la fois. C'est-à-dire que l'on choisit de placer en haut de l'arbre la variable qui apporte le plus d'information sur Y. Cette variable sera renommée « X_1 ». Puis, au deuxième pas de l'algorithme, parmi les variables restantes, on choisit celle qui apporte le plus d'information sans tenir compte

de la variable précédente. Cette variable est renommée « X_2 » et correspondra à la variable à tester au deuxième niveau de l'arbre... et ainsi de suite jusqu'à épuisement des variables. La méthode consiste donc à :

$$\min_{X_i \in (X \setminus S_{k-1})} (H(Y/X_i))$$

$$\begin{aligned} S_0 &= \emptyset \\ S_k &= S_{k-1} \times X_i \end{aligned}$$

ce qui revient, en utilisant l'indice $q(Y/S)$, à :

$$\max_{X_i \in (X \setminus S_{k-1})} (q(Y/X_i))$$

$$\begin{aligned} S_0 &= \emptyset \\ S_k &= S_{k-1} \times X_i \end{aligned}$$

où S_0 représente le vecteur S initial ; et S_k, le vecteur S au $k^{ième}$ pas de l'algorithme.

On peut illustrer la démarche de la façon suivante :

Étant donné X le vecteur des variables potentiellement explicatives, on cherche à établir une liste de variables (appartenant à X) à tester dans un ordre précis afin d'expliquer la variable Y.

Au $k^{ième}$ pas de l'algorithme, celui-ci a déterminé les k premières variables à tester. La figure 4.7 illustre cette approche.

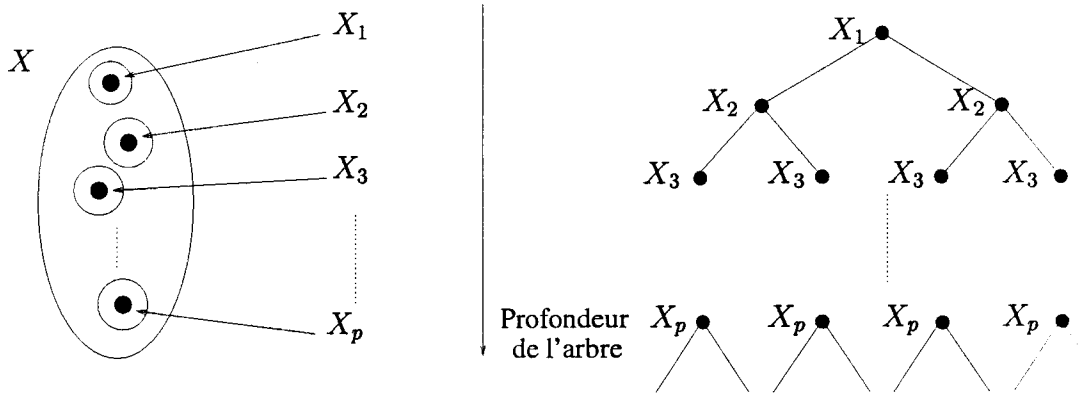

FIG. 4.7: *Approche triviale*

Le principal intérêt de cette méthode est sa simplicité. En effet, pour chaque variable X_i ($i = 1 \ldots p$), on calcule l'entropie conditionnelle de SHANNON une seule fois et il suffit de classer ces variables par ordre croissant de l'entropie. L'inconvénient majeur est qu'à un niveau de l'arbre, on ne prend pas en compte l'information déjà apportée aux niveaux supérieurs. Il y a donc de fortes chances pour que nous traitions la même information (la

même sous-population d'apprentissage pourra être discriminée plusieurs fois) à des niveaux différents.

En appliquant cette méthode sur notre exemple didactique, on obtient l'arbre de la figure 4.8.

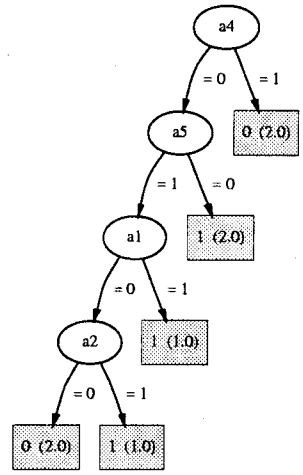

FIG. 4.8: *Arbre obtenu avec l'approche triviale*

Étant donné que les variables sont triées par ordre croissant de l'entropie conditionnelle, les variables a_4 et a_5 sont évidemment placées en haut de l'arbre et on ne teste a_1 et a_2 qu'ensuite. Cette démarche n'apparaît pas satisfaisante.

Nous proposons ainsi d'autres méthodes d'induction par niveau.

4.3.2 Approche descendante agrégative par niveau

La démarche consiste ici à chercher la variable X_i apportant le plus d'information sur Y. Cette variable X_i est renommée par l'algorithme X_1 et est placée à la racine de l'arbre de décision.

Dans un deuxième temps, l'algorithme cherche une autre variable qui, associée à la variable X_1, apporte le plus d'information sur Y. Cette variable est renommée X_2 et représente la variable-test placée au deuxième niveau de l'arbre. La même démarche est appliquée afin de trouver la variable testée au troisième niveau de l'arbre (renommée X_3) ; et ainsi de suite itérativement... jusqu'à validation du critère d'arrêt.

Cette démarche est différente de l'approche triviale (sans effet mémoire) dans le sens où à un niveau donné de l'arbre, afin de déterminer la variable à tester, on tient compte de l'information déjà traitée plus haut. Cette démarche correspond en fait à affiner la partition de la population d'apprentissage Ω suivant les modalités de la nouvelle variable à tester.

Cette méthode a été appelée « méthode agrégative » car elle consiste à agréger successivement les variables pertinentes. Le terme « descendant », quant à lui, vient du fait que

la (les) première(s) variable(s) trouvée(s) par l'algorithme est (sont) la (les) première(s) variable(s) à tester.

L'approche consiste donc à :

$$\min_{X_i \in (X \setminus S_{k-1})} (H(Y/(S_{k-1} \times X_i)))$$

$$\begin{aligned} S_0 &= \emptyset \\ S_k &= S_{k-1} \times X_i \end{aligned}$$

ou :

$$\max_{X_i \in (X \setminus S_{k-1})} (q(Y/(S_{k-1} \times X_i)))$$

$$\begin{aligned} S_0 &= \emptyset \\ S_k &= S_{k-1} \times X_i \end{aligned}$$

où S_0 représente le vecteur S initial ; et S_k, le vecteur S au $k^{ième}$ pas de l'algorithme.

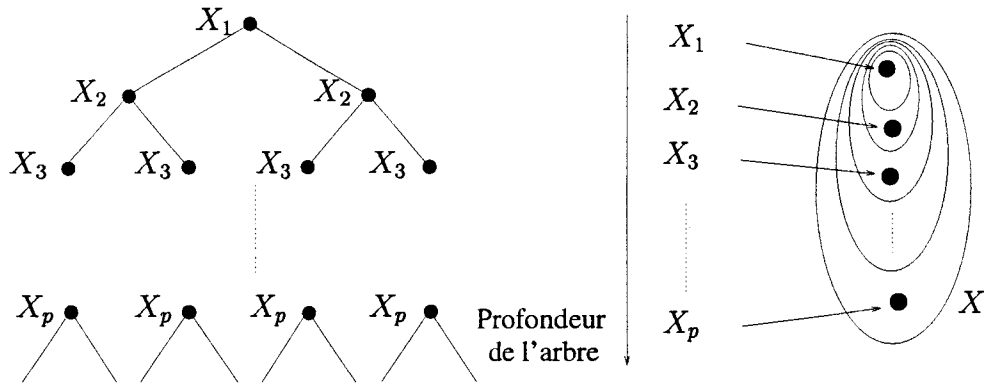

FIG. 4.9: *Approche descendante agrégative*

La figure 4.9 illustre cette approche. À chaque niveau de l'arbre, on trouvera une et une seule variable. La construction de l'arbre est arrêtée dès lors qu'un des critères d'arrêt est respecté. Les cercles symbolisent l'agrégation des différentes variables prises en compte dans l'arbre.

Nous proposons ainsi l'algorithme suivant :

Programme APPROCHE DESCENDANTE AGRÉGATIVE PAR NIVEAU
; initialisation
nb_{max} ; q_{min} ; $S \leftarrow \{\}$
; boucle
Répéter
calculer les $q(Y/S \times X_i)$ avec $X_i \notin S$
retenir X_i qui maximise $q(Y/S \times X_i)$
$S \leftarrow S \times (X_i)$
Jusqu'à $(q(Y/S) > q_{min})$ ou $(\text{card}S = nb_{max})$
tracer l'arbre de décision et donner $q(Y/S)$
Fin

L'avantage de cet algorithme est que son implantation et sa compréhension sont très simples. Les critères d'arrêt sont définis par l'utilisateur (nb_{max}, q_{min}). S correspond initialement à l'ensemble vide (aucune variable n'est prise en compte) et est mis à jour itérativement à chaque pas de l'algorithme.

En appliquant cette méthode sur notre exemple, on obtient l'arbre de la figure 4.10.

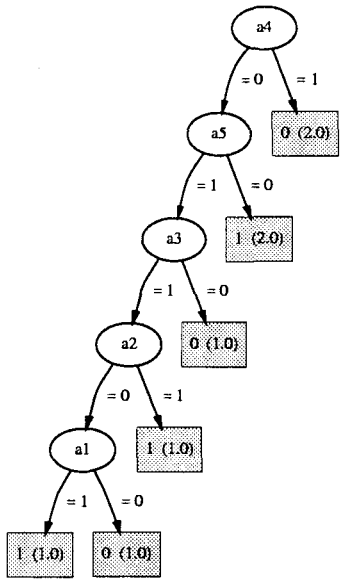

FIG. 4.10: *Arbre obtenu avec l'approche descendante agrégative*

Comme l'approche triviale, la première variable testée est la variable a_4. Puis, la variable a_5 est choisie car elle est l'une des variables qui, associée à a_4, apporte le plus d'information sur Y (en fait, dans cet exemple, plusieurs choix sont possibles pour cette variable car on obtient plusieurs entropies égales en raison de la petite taille du tableau de données). Et ainsi de suite, pour finalement tester a_2 et a_1. L'arbre n'est donc pas optimal.

4.3.3 Approche descendante désagrégative par niveau

Dans cette démarche, au lieu de chercher directement la variable qui explique le plus Y, on se propose d'écarter le sous-vecteur constitué de $p-1$ variables expliquant le moins Y. La variable restante ne sera donc plus celle qui, à elle toute seule, apporte le plus d'information, mais celle qui, combinée avec d'autres variables, apporte beaucoup d'information sur Y. En effet, elle n'appartient pas au sous-vecteur le moins explicatif, mais appartient à tous les autres. Cette variable constitue la racine de l'arbre et est renommée X_1.

Le deuxième pas de l'algorithme consiste alors à réitérer l'opération après avoir ôté X_1 de l'ensemble des variables résiduelles. On se propose ainsi d'écarter le sous-vecteur constitué de $p-2$ variables expliquant le moins Y. La variable restant est alors placée au deuxième niveau de l'arbre et est renommée X_2.

Et ainsi de suite...

On a donc pris en compte les sous-vecteurs constitués de $p-1$ variables à la première itération pour l'évaluation du critère. Puis les sous-vecteurs constitués de $p-2$ variables etc... C'est en ce sens que cet algorithme a été qualifié d'algorithme « désagrégatif » (on désagrège petit à petit l'ensemble des variables résiduelles).

La méthode consiste alors à :

$$\max_{X_i \in S_{k-1}} (H(Y/(S_{k-1} \backslash X_i)))$$

$$\begin{aligned} S_0 &= X \\ S_k &= S_{k-1} \backslash X_i \end{aligned}$$

ou, ce qui revient au même, à :

$$\min_{X_i \in S_{k-1}} (q(Y/(S_{k-1} \backslash X_i)))$$

$$\begin{aligned} S_0 &= X \\ S_k &= S_{k-1} \backslash X_i \end{aligned}$$

où S_0 représente le vecteur S initial ; et S_k, le vecteur S au $k^{ième}$ pas de l'algorithme.

On cherche le sous-vecteur constitué de $p-1$ variables qui explique le moins Y (que nous renuméroterons $(X_2, \ldots, X_p)$), puis le sous-vecteur constitué de $p-2$ variables qui explique le moins Y (que nous renuméroterons $(X_3, \ldots, X_p)$), et ainsi de suite jusqu'à ce que le critère d'arrêt soit validé. On désagrège bien ici le vecteur X en sous-vecteurs les moins explicatifs possible (figure 4.11).

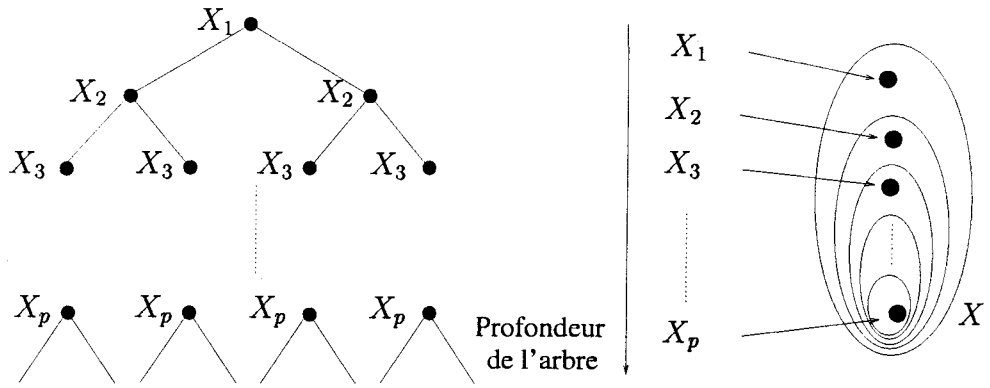

FIG. 4.11: *Approche descendante désagrégative*

Dans ce cadre, nous proposons l'algorithme suivant :

Programme APPROCHE DESCENDANTE DÉSAGRÉGATIVE PAR NIVEAU
; initialisation
nb_{max} ; q_{min} ; $S \leftarrow \{\}$; $Res \leftarrow X$
; boucle
Répéter
calculer les $q(Y/Res \backslash X_i)$ avec $X_i \in Res$
retenir X_i qui minimise $q(Y/Res \backslash X_i)$
$S \leftarrow S \times (X_i)$; $Res \leftarrow Res \backslash X_i$
Jusqu'à $(q(Y/S) > q_{min})$ ou $(\text{card}S = nb_{max})$
tracer l'arbre de décision et donner $q(Y/S)$
Fin

De même que pour l'algorithme précédent, les critères d'arrêt sont définis par l'utilisateur (nb_{max}, q_{min}).

S correspond initialement à l'ensemble vide (aucune variable n'est prise en compte) tandis que *Res* (le vecteur des variables résiduelles) est fixé à X (ce qui signifie que toutes les variables sont stockées dans *Res*). Ces deux vecteurs sont mis à jour itérativement à chaque pas de l'algorithme : le vecteur S est mis à jour en agrégeant petit à petit les variables à tester, et le vecteur *Res* des variables résiduelles est par contre désagrégé itérativement.

En reprenant notre exemple, cette méthode fournit l'arbre de la figure 4.12.

Dans un premier temps, l'algorithme a cherché le sous-ensemble de quatre variables le moins explicatif afin d'écarter la variable qui appartient à tous les autres sous-ensembles de quatre variables plus explicatifs que ce dernier. Il est donc logique que l'on écarte l'une des deux variables a_1 ou a_2 car, précisément, à elle deux elles expliquent totalement Y. Elles ne peuvent donc pas appartenir toutes les deux au sous-ensemble le moins explicatif. Une de ces deux variables (ici, a_2) est donc placée en haut de l'arbre, et un début de solution

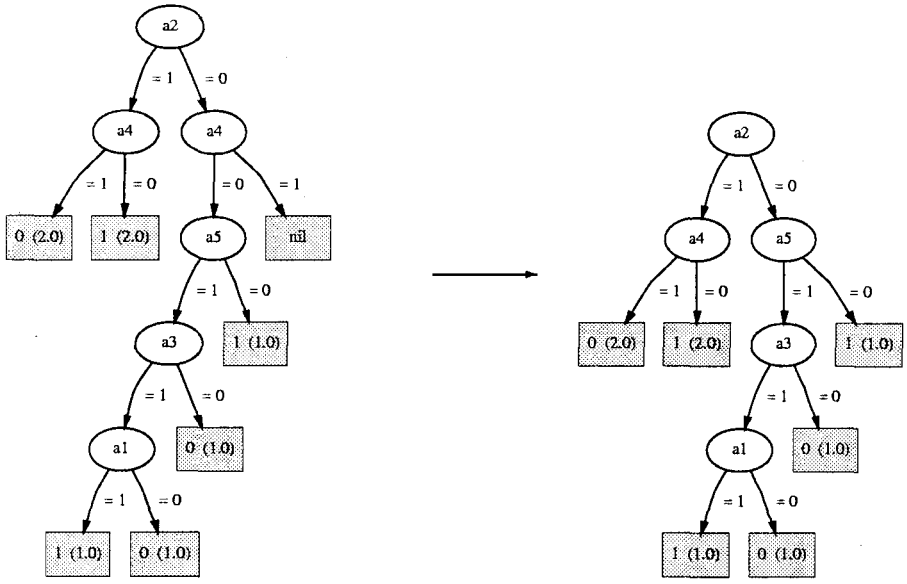

FIG. 4.12: *Arbre obtenu avec l'approche descendante désagrégative*

apparaît. Néanmoins, au deuxième pas de l'algorithme, cette variable est ôtée de l'ensemble des variables candidates pour le deuxième niveau de l'arbre. Dissociée de a_2, la variable a_1 peut alors très bien faire partie du sous-ensemble de trois variables le moins explicatif. C'est ici le cas, et elle n'est donc pas choisie. Nous n'obtenons donc pas un résultat satisfaisant.

D'autre part, il est à remarquer dans cet exemple, qu'au deuxième niveau de l'arbre, on obtient au deuxième nœud un test inutile car lorsque $a_2 = 0$, alors $a_4 = 0$. Dans la pratique, on supprime ce test en remontant le sous-arbre inférieur (ou fils) d'un niveau.

4.3.4 Approche ascendante agrégative par niveau

Cette approche consiste à chercher dans un premier temps la variable « la moins explicative », celle que l'on placera en bas de l'arbre. C'est dans ce sens que cette approche est qualifiée ici d'approche « ascendante ». L'algorithme cherche ainsi la variable X_i qui apporte le moins d'information sur Y ; elle est renommée X_1.

Dans un deuxième temps, il cherche la variable qui, associée à X_1 (la variable mise en bas de l'arbre) apporte le moins d'information sur Y. Cette variable est renommée X_2 et constitue l'avant-dernier niveau de l'arbre... Et ainsi de suite jusqu'à ce qu'il ne reste plus qu'une variable, qui constituera la racine de l'arbre. L'approche consiste donc à :

$$\max_{X_i \in (X \setminus S_{k-1})} (H(Y/(S_{k-1} \times X_i)))$$

$$\begin{aligned} S_0 &= \emptyset \\ S_k &= S_{k-1} \times X_i \end{aligned}$$

ou :

$$\min_{X_i \in (X \setminus S_{k-1})} (q(Y/(S_{k-1} \times X_i)))$$

$$\begin{aligned} S_0 &= \emptyset \\ S_k &= S_{k-1} \times X_i \end{aligned}$$

où S_0 représente le vecteur S initial ; et S_k, le vecteur S au $k^{ième}$ pas de l'algorithme.

Cette démarche est illustrée sur la figure 4.13 :

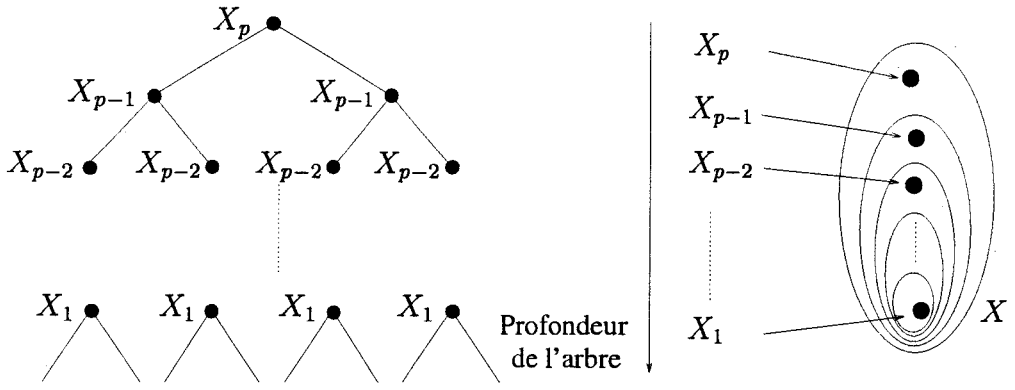

FIG. 4.13: *Approche ascendante agrégative*

La première variable trouvée est renommée X_1 par l'algorithme et constitue la dernière variable à tester. Elle représente la variable apportant le moins d'information sur Y (elle n'en apporte peut être pas du tout, et dans ce cas $H(Y/X_1) = H(Y)$, c'est-à-dire que l'incertitude sur Y est aussi importante en connaissant les valeurs prises par X_1 qu'en ne les connaissant pas) parmi toutes celles retenues, c'est-à-dire toutes celles composant le vecteur X.

La même remarque peut être établie à l'avant-dernier niveau (pour X_2), et de façon général, pour tous les niveaux les plus bas de l'arbre de décision (voir figure 4.13).

Cette phase de détermination de l'ordre des variables à tester est en fait une phase préliminaire à la construction de l'arbre. Cette remarque implique que les variables placées dans les derniers niveaux de l'arbre pourront ne pas être testées. En effet, une fois construit, l'arbre pourra peut être permettre de répondre aux critères d'arrêt fixés par l'utilisateur bien avant de tester les derniers niveaux.

Nous proposons ainsi l'algorithme suivant :

Programme APPROCHE ASCENDANTE AGRÉGATIVE PAR NIVEAU
; Cette procédure consiste à déterminer l'ordre dans lequel les variables seront testées dans l'arbre
; On utilise une approche agrégative...
; initialisation
$Res \leftarrow \{\}$; $S \leftarrow X$
; boucle
Répéter
calculer les $q(Y/Res \times X_i)$ avec $X_i \in S$
retenir X_i qui minimise $q(Y/Res \times X_i)$
$S \leftarrow S \backslash (X_i)$; $Res \leftarrow Res \times (X_i)$
Jusqu'à $S = \{\}$
Renumérotation des variables à tester dans l'ordre décroissant (X_p = 1ère / X_{p-1} = 2ème / ...)
; Cette procédure consiste à déterminer l'arbre...
; initialisation
nb_{max} ; q_{min}
; boucle
Répéter
$i \leftarrow i+1$; $S \leftarrow S \times (X_i)$; $Res \leftarrow Res \backslash X_i$
calculer $q(Y/S)$
Jusqu'à $(q(Y/S) > q_{min})$ ou $(i = \text{card}S = nb_{max})$
tracer l'arbre de décision et donner $q(Y/S)$
Fin

Cet algorithme est constitué de deux boucles, la première permettant de fixer l'ordre de test des variables (en utilisant ici une approche ascendante vis-à-vis de l'arbre que l'on construira), et la deuxième permettant de construire l'arbre (utilisant une approche classique de construction descendante) en prenant en compte la contrainte d'ordre des variables à tester déterminé par la première boucle.

L'ordre des variables à tester étant fixé, l'arbre est construit jusqu'à ce qu'il réponde aux critères d'arrêt ; et par conséquent, les premières variables trouvées par l'algorithme ne seront peut être pas testées.

Remarque : Un abus de langage dans la dénomination de cette méthode est réalisé dans ce travail. En effet, cette approche est *ascendante* lorsqu'on cherche à déterminer l'ordre des variables à tester, mais l'arbre est toujours construit de façon *descendante*, en tenant compte de l'ordre des variables à tester.

Sur l'exemple traité, l'algorithme fournit l'arbre de la figure 4.14.

On peut remarquer que cet arbre est identique à celui obtenu par la méthode précédente. Cette coïncidence ne se vérifie pas dans le cas général. Néanmoins, une démarche

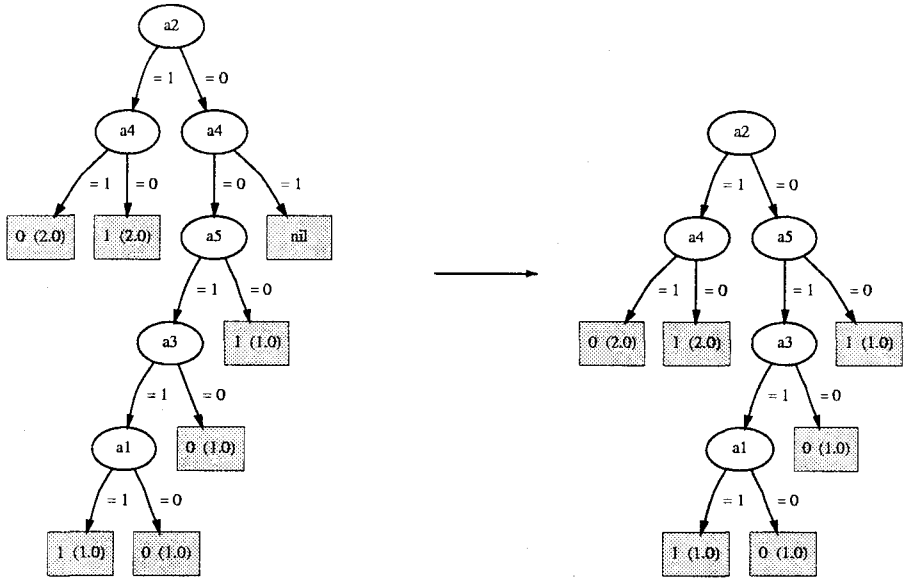

FIG. 4.14: *Arbre obtenu avec l'approche ascendante agrégative*

duale à la précédente est appliquée ici. En effet, l'algorithme cherche dans un premier temps la variable la moins informative afin de la placer au bas de l'arbre de décision. Étant donné que les variables a_1 et a_2 prises séparément n'apportent aucune information sur Y, il est logique que l'une de ces deux variables (ici a_1) soit choisie. Dans un deuxième temps, l'algorithme recherche les sous-ensembles les moins explicatifs. On s'aperçoit que la deuxième variable importante (a_2) ne peut pas être choisie puisque avec a_1 elle explique complètement Y. On la retrouve donc en haut de l'arbre.

D'autre part, la même remarque peut être faite au sujet du nœud inutile du deuxième niveau, qui est finalement supprimé.

4.3.5 Approche ascendante désagrégative par niveau

Dans cette approche, le sous-vecteur constitué des $p-1$ variables expliquant le plus Y sera écarté. La variable restante (renommée X_1) n'appartient pas au sous-vecteur le plus explicatif, et par conséquent, elle sera placée au bas de l'arbre. L'opération est réitérée sur les $p-1$ variables, en recherchant les $p-2$ variables les plus informatives. La variable restant (renommée X_2) constituera alors l'avant-dernier niveau de l'arbre ; et ainsi de suite jusqu'à trouver la variable « la plus explicative », constituant la racine de l'arbre (renommée X_p).

L'approche consiste donc à :

$$\min_{X_i \in S_{k-1}} (H(Y/(S_{k-1} \backslash X_i)))$$

$$\begin{aligned} S_0 &= X \\ S_k &= S_{k-1} \backslash X_i \end{aligned}$$

ou :

$$\max_{X_i \in S_{k-1}} (q(Y/(S_{k-1} \backslash X_i)))$$

$$\begin{aligned} S_0 &= X \\ S_k &= S_{k-1} \backslash X_i \end{aligned}$$

où S_0 représente le vecteur S initial ; et S_k, le vecteur S au $k^{ième}$ pas de l'algorithme.

Cette démarche est illustrée sur la figure 4.15. Les cercles représentent les variables les plus informatives à un niveau donné. À titre d'exemple, au niveau 3 de l'arbre, le vecteur de variables prises en compte est (X_p, X_{p-1}, X_{p-2}).

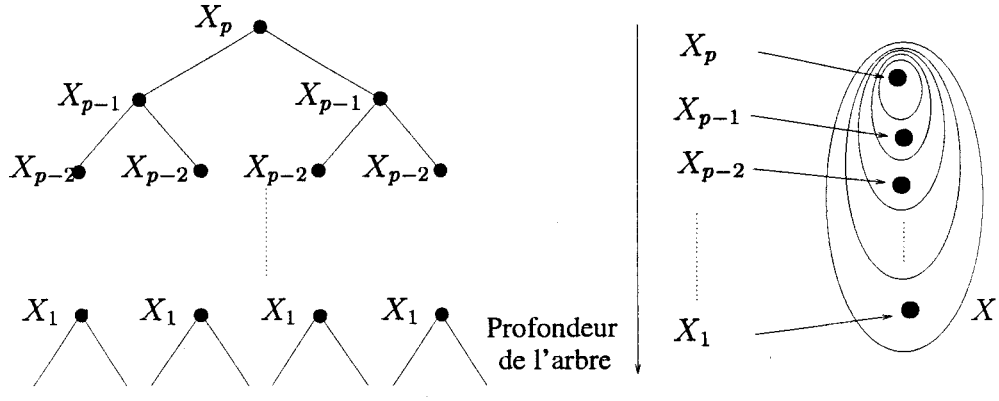

FIG. 4.15: *Approche ascendante désagrégative*

Nous proposons ainsi l'algorithme suivant :

Programme APPROCHE ASCENDANTE DÉSAGRÉGATIVE PAR NIVEAU
; Cette procédure consiste à déterminer l'ordre dans lequel les variables seront testées dans l'arbre
; On utilise une approche désagrégative...
; initialisation
$Res \leftarrow \{\}$; $S \leftarrow X$
; boucle
Répéter
calculer les $q(Y/S\backslash X_i)$ avec $X_i \in S$
retenir X_i qui maximise $q(Y/S\backslash X_i)$
$S \leftarrow S\backslash(X_i)$; $Res \leftarrow Res \times (X_i)$
Jusqu'à $S = \{\}$
Renumérotation des variables à tester dans l'ordre décroissant (X_p = 1ère / X_{p-1} = 2ème / ...)
; Cette procédure consiste à déterminer l'arbre...
; initialisation
nb_{max} ; q_{min}
; boucle
Répéter
$i \leftarrow i+1$; $S \leftarrow S \times (X_i)$; $Res \leftarrow Res\backslash X_i$
calculer $q(Y/S)$
Jusqu'à $(q(Y/S) > q_{min})$ ou $(i = \text{card}S = nb_{max})$
tracer l'arbre de décision et donner $q(Y/S)$
Fin

L'algorithme met bien en évidence deux phases. La première consiste à déterminer l'ordre des variables à tester en utilisant une approche désagrégative, et la deuxième phase consiste à construire l'arbre jusqu'à ce que l'un des deux critères soit validé.

Remarque : On retrouve le même abus de langage que pour l'approche précédente.

En appliquant cette dernière approche par niveau sur notre exemple, on obtient l'arbre de la figure 4.16.

Cette fois-ci, l'algorithme va rechercher les sous-ensembles de variables les plus explicatifs en les désagrégeant petit à petit. Comme les deux variables a_1 et a_2 composent le sous-ensemble de variables le plus explicatif, l'algorithme les conservent jusqu'à la fin. Elles se retrouvent donc en haut de l'arbre. Une fois ces deux variables testées, Y est complètement expliquée et les autres variables ne sont plus nécessaires. On obtient par conséquent un arbre à deux niveaux et à trois nœuds qui représente l'arbre optimal pour cet exemple.

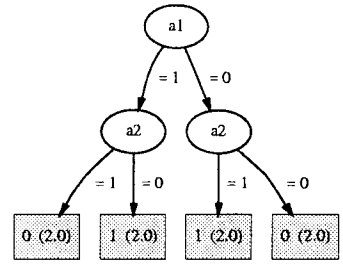

FIG. 4.16: *Arbre obtenu avec l'approche ascendante désagrégative*

4.3.6 Conclusion sur les méthodes de construction de l'arbre par niveau

L'approche ascendante désagrégative a été introduite dans [PPS95]. Cette approche, ainsi que quatre autres approches sont développées dans [PP97a]. Une démarche similaire à celle-ci est utilisée par ZIGHED et SEBBAN dans [ZS98],afin de réduire l'espace de représentation du système physique. Le critère utilisé n'est pas, comme ici, l'entropie conditionnelle, mais ce que l'auteur appelle le *test des arêtes* introduit dans [Seb96].

Afin de comparer ces différentes méthodes, intéressons nous à l'information apportée sur Y en fonction du nombre de variables testées. Notons Γ cette courbe représentative (figure 4.17).

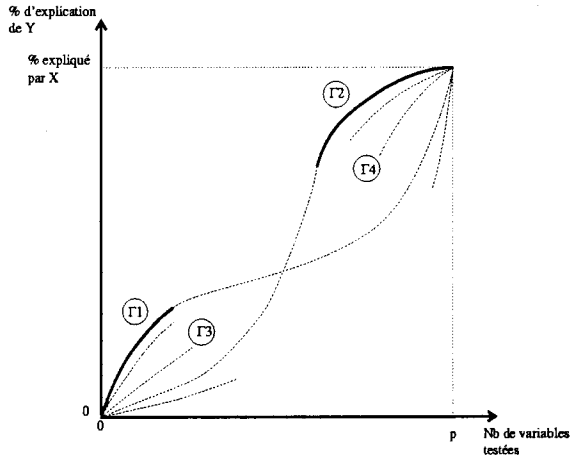

FIG. 4.17: *Comparaison des méthodes de construction de l'arbre de décision par niveau*

Les deux axes de cette figure représentent en fait les deux critères d'arrêt utilisés dans les algorithmes.

- *Approche descendante agrégative* (Γ_1) : La première variable testée est celle qui apporte le plus d'information à elle seule sur Y. Cette courbe sera donc caractérisée

par une pente importante lorsque le nombre de variables à tester est petit. Les autres méthodes ne seront pas aussi performantes dans cette zone. Elles seront caractérisées par une pente moins abrupte.

- *Approche ascendante désagrégative* (Γ_2) : Lorsque le nombre de variables à tester est égal à $p-1$ (où $p = card(X)$), cette approche nous amène à considérer le vecteur des $p-1$ variables apportant le plus d'information sur Y. La dernière variable apporte donc la dernière quantité d'information sur Y, et on obtient dans cette zone une courbe de faible pente. De même, dans cette zone, les courbes correspondant aux autres algorithmes ne passeront pas au-dessus de Γ_2.

- *Approche descendante désagrégative* (Γ_3) : Cette approche consiste à tester d'abord la variable qui n'appartient pas au sous-vecteur constitué des $p-1$ variables qui apporte le moins d'information sur Y. Cette démarche n'indique pas si la variable elle-même apporte beaucoup d'information ou pas. On sait seulement qu'elle apporte beaucoup d'information avec d'autres variables. Le début de courbe, lorsque le nombre de variables à tester est faible, se situera donc sous la limite imposée par la courbe Γ_1.

- *Approche ascendante agrégative* (Γ_4) : De même, en fin de courbe, la dernière variable apporte peu d'information sur Y, et rien n'indique que le sous-vecteur constitué des $p-1$ variables restantes est celui qui apporte le plus d'information sur Y. La seule certitude est que la fin de cette courbe se situera en-dessous de la courbe Γ_2.

Remarques :

- Il est à noter que si les variables explicatives n'apportent pas d'information commune deux à deux, alors les quatre courbes sont confondues avec la courbe représentative de la solution triviale.

- Pour un pourcentage d'explication E_i donné $(100-\delta_p)$ et proche de 100%, l'algorithme ascendant désagrégatif donnera de meilleurs résultats afin de trouver le plus petit ensemble de variables (parmi X) qui permette d'obtenir E_i.

Par ailleurs, si l'on cherche, à partir de p variables potentiellement explicatives, le sous-ensemble de p' variables ($p' < p$) qui explique le mieux Y, la solution optimale sera obtenue à coup sûr par une démarche exhaustive, en calculant

$$C_p^{p'} = \frac{p!}{p'!(p-p')!}$$

entropies différentes (algorithme de complexité exponentielle).

En revanche, cette même solution sera obtenue pour p' proche de p par une approche ascendante désagrégative en calculant

$$\sum_{i=0}^{p-p'-1} p-i$$

entropies (algorithme de complexité polynomiale). En effet, on calcule :

$$\begin{cases} p \text{ entropies } H(Y/(X\backslash X_i)) & \rightarrow S_1 = X\backslash X_i \text{ au premier pas de l'algorithme} \\ p-1 \text{ entropies } H(Y/(S_1\backslash X_i)) & \rightarrow S_2 = S_1\backslash X_i \text{ au deuxième pas de l'algorithme} \\ \vdots & \\ p'+1 \text{ entropies } H(Y/(S_{p-p'-1}\backslash X_i)) & \end{cases}$$

À titre d'exemple, pour $p = 10$ et $p' = 8$, la recherche exhaustive calcule 45 entropies, alors que l'approche ascendante désagrégative en calcule 19.

Dans le cadre de l'explication, l'intérêt principal des méthodes ascendantes désagrégatives consiste à éviter de choisir une variable qui *a priori* apporte beaucoup d'information à elle seule, alors qu'un sous-vecteur de variables peut expliquer presque totalement Y, bien que ces variables prises séparément n'apportent que très peu d'information.

Afin d'illustrer ce phénomène, nous pouvons reprendre les résultats obtenus sur l'exemple du tableau de données 4.2. Y représentait le bit de parité de a_1 et de a_2, et les autres variables étaient des variables supplémentaires inutiles pour connaître la valeur de Y.

Dans cet exemple, les variables a_4 et a_5 apportent plus d'information à elles seules (prises séparément) que les variables a_1 ou a_2 (voir tableau B.1), alors que le couple (a_1, a_2) explique complètement Y, c'est-à-dire que la connaissance des valeurs de (a_1, a_2) nous permet d'en déduire la valeur de Y.

Appliquons les différentes méthodes développées sur ce tableau. Les résultats sont donnés dans le tableau 4.3.

	Algorithme	Ordre des variables testées
	descendant agrégatif	$a_4\ a_5\ a_3\ a_2\ a_1$
	descendant désagrégatif	$a_2\ a_5\ a_4\ a_3\ a_1$
	ascendant agrégatif	$a_2\ a_5\ a_4\ a_3\ a_1$
$\longrightarrow$	ascendant désagrégatif	$a_1\ a_2\ \cancel{a_3\ a_4\ a_5}$

TAB. 4.3: *Résultats obtenus sur l'exemple traité*

La méthode triviale (sans effet mémoire) choisira alors la variable a_4 comme la plus explicative de Y, puis a_5, etc...

L'algorithme descendant agrégatif va également choisir a_4 comme variable à tester en premier, puis une des variables restantes, etc... L'arbre obtenu n'est pas optimal.

Par contre, l'algorithme ascendant désagrégatif nous donnera les variables a_1 et a_2 à tester en premier puisqu'il cherche directement le sous-vecteur le plus explicatif. On obtient dans ce cas l'arbre optimal de la figure 4.16.

Les deux autres algorithmes vont quant à eux apporter une solution partielle au problème. En effet, l'algorithme descendant désagrégatif va écarter le sous-vecteur de variables le moins explicatif. Il va donc choisir a_1 ou a_2 en premier. Puis, comme une des deux variables du sous-vecteur le plus explicatif est, à la deuxième itération, absente de l'ensemble des variables prises en compte dans le calcul du critère, on ne peut plus l'associer à la seconde et cette dernière peut alors appartenir au sous-vecteur le moins explicatif. Cette variable n'est donc pas choisie. Cet algorithme possède donc un bon comportement à la première itération ; par contre, lors des itérations suivantes, les résultats sont moins probants.

L'algorithme ascendant agrégatif va, quant à lui, choisir la variable la moins explicative a_1 (ou a_2), pour la placer au dernier niveau de l'arbre de décision. Le sous-vecteur le moins explicatif contenant cette variable va ensuite être choisi. La deuxième variable a_2 sera donc choisie en dernier puisque le couple (a_1, a_2) est le plus explicatif. Elle sera placée en haut de l'arbre. Cet algorithme possède donc un bon comportement dans les dernières itérations, mais pas dans les premières.

L'approche ascendante désagrégative obtient donc logiquement les meilleurs résultats au niveau de la sélection des variables.

L'étude des méthodes de construction par niveau nous a permis de mettre en évidence cette caractéristique. Mais dans la pratique, l'utilisation d'un arbre de décision par niveau n'est pas très judicieuse car elle aboutit à un découpage rigide de l'espace des variables (figure 4.2). C'est pourquoi nous proposons des méthodes de construction par nœud tout en conservant les caractéristiques de l'approche ascendante désagrégative.

4.4 Les méthodes de construction par nœud

Après avoir souligné l'intérêt d'une approche ascendante désagrégative, on peut affiner cette méthode en l'appliquant de manière plus locale afin de construire l'arbre [PP97b], c'est-à-dire en tenant compte, à chaque nœud engendré, du chemin déjà parcouru. On peut retrouver les mêmes algorithmes que précédemment (ou tout au moins en ce qui concerne les algorithmes descendants), en intégrant le chemin parcouru dans l'arbre à chaque nœud (c'est-à-dire les modalités prises par chaque variable testée pour parvenir à ce nœud). Il s'agit en fait de conditionner le critère employé en prenant en compte les modalités prises par les variables déjà choisies. Cela revient à considérer le sous-ensemble d'apprentissage correspondant au nœud considéré de l'arbre.

La figure 4.18 illustre ce propos. Dans cet exemple, les variables tests du troisième niveau sont X_4, X_3, X_2 et X_4. Elles sont déterminées de façon à engendrer une partition de l'ensemble d'apprentissage Ω permettant une discrimination accrue des conclusions.

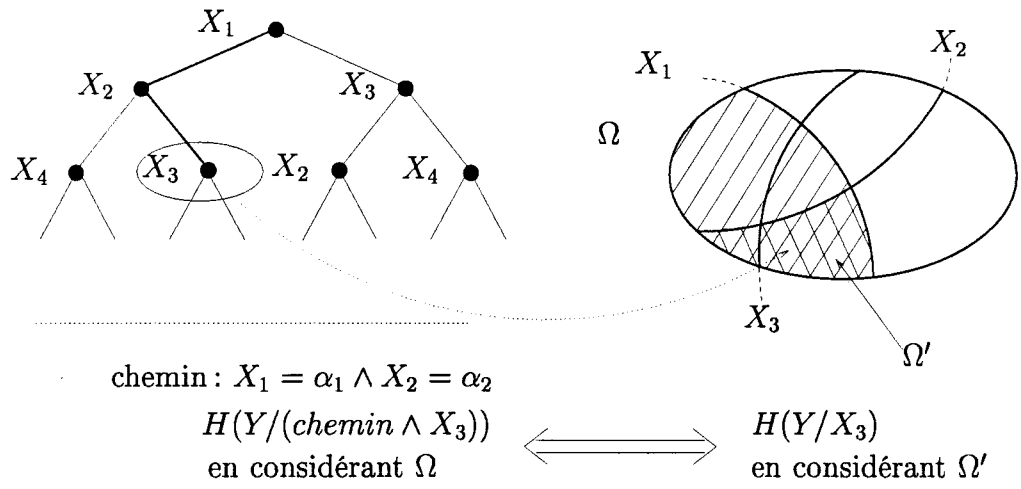

FIG. 4.18: *Le chemin parcouru consiste à prendre en compte une partie de* Ω

Afin de tester la variable X_3 au troisième niveau de l'arbre, il faut emprunter le chemin $X_1 = \alpha_1$ et $X_2 = \alpha_2$. Ce chemin permet de déterminer une classe Ω' de la partition de Ω, et la variable X_3 est déterminée de façon à discriminer au mieux les conclusions de cette classe.

Considérer une quantité d'information $H(Y/(chemin \wedge X_3))$ sur la population d'apprentissage entière est donc équivalent à considérer $H(Y/X_3)$ en considérant la partie de la population d'apprentissage déterminée par le chemin.

4.4.1 Approche descendante agrégative par nœud

Avant de présenter une approche ascendante désagrégative, attardons nous sur l'approche descendante agrégative, car elle est très utilisée dans la littérature. En effet, c'est dans cette catégorie d'algorithmes que l'on trouve *ID3* et son successeur *C4.5* [Qui83, Qui86, Qui93].

Dans ces conditions, le critère suivant est appliqué à chaque nœud de l'arbre :

$$\min_{X_i \notin chemin} (H(Y/(chemin\, parcouru \wedge X_i)))$$

Si l'indice $q(Y/S)$ est utilisé en considérant les sous-populations d'apprentissage, le critère est équivalent à :

$$\max_{X_i \notin chemin} (q(Y/X_i))$$

On peut noter qu'à la première itération, ce critère est équivalent à celui de la méthode descendante agrégative globale.

La démarche consistera alors à appeler une procédure récursive : APPROCHE DESCENDANTE AGRÉGATIVE PAR NŒUD(Ω, X), c'est-à-dire à appeler cette procédure

en considérant l'ensemble d'apprentissage entier Ω, et le vecteur complet des variables potentiellement explicatives X.

Nous définissons ainsi la procédure récursive suivante :

```
Procédure APPROCHE DESCENDANTE AGRÉGATIVE PAR
NŒUD(Ω', Res)
    Ω* est une variable locale à la procédure

    calculer les q_Ω'(Y/X_i) avec X_i ∈ Res
    retenir X_i qui maximise q_Ω'(Y/X_i)
    Res ← Res\X_i
    Pour j : = 1 à cardM_Xi Faire
        Ω* ← Ω' ∩ {X_i^-1(α_j^i)} où α_j^i ∈ M_Xi
        Si (q_Ω*(Y/X_i) > q_min) ou (cardRes = 0) Alors
            tracer la feuille correspondant à la classe
            majoritaire pour Ω*
        Sinon
            APPROCHE DESCENDANTE AGRÉGATIVE PAR
            NŒUD(Ω*, Res)
        Fin Si
    Fin Pour
Fin
```

La procédure possède deux paramètres d'entrée, à savoir Ω' et *Res* représentant respectivement une partie de la population d'apprentissage et un vecteur de variables résiduelles, c'est-à-dire un vecteur de variables pouvant encore être testées.

Ω^* représente une variable locale à la procédure, et correspond à une partie de la population d'apprentissage.

L'indice $q_{\Omega'}(Y/X_i)$ désigne l'indice $q(Y/X_i)$ évalué sur la sous-population d'apprentissage Ω'.

Cette procédure a été réalisée avec la contrainte qui consiste à appliquer le critère d'arrêt sur chaque nœud (donc sur une partie de la population d'apprentissage), et non pas de façon globale sur la population d'apprentissage entière. On spécialise alors chaque nœud obtenu jusqu'à ce que le critère d'arrêt soit validé. En utilisant cette démarche, on discrimine les conclusions tant que c'est possible.

Nous pouvons reprendre l'exemple du tableau de données 4.2 afin d'illustrer cette approche. On obtient l'arbre de la figure 4.19.

Nous pouvons remarquer que, comme l'approche par niveau, cet algorithme choisit en premier la variable a_4 qui apporte plus d'information à elle toute seule que la variable a_1 ou a_2. Sur cet exemple, l'arbre obtenu est identique à celui obtenu par l'approche descendante agrégative par niveau. Ceci est dû à la petite taille du tableau de données considéré. Dans

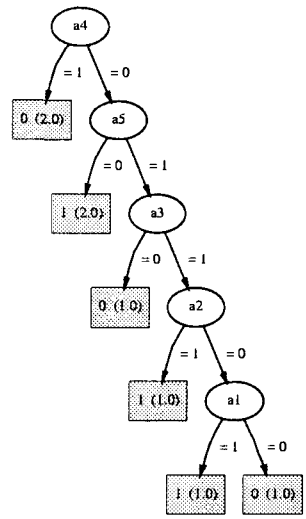

FIG. 4.19: *Arbre obtenu avec l'approche descendante agrégative par nœud*

le cas général, les arbres obtenus sont différents. En effet, avec l'approche par nœud, nous pouvons obtenir différentes variables sur un même niveau.

4.4.2 Approche ascendante désagrégative par nœud

Nous avons montré précédemment tout l'intérêt des approches désagrégatives par niveau permettant d'isoler des groupes de variables explicatives.

Ainsi, si la variable à expliquer Y prend une certaine valeur lorsque les deux variables explicatives X_1 et X_2 prennent chacune la même valeur ($X_1 = X_2$), alors la connaissance de X_1 et X_2 sera primordiale pour expliquer Y même si X_1 et X_2 prises séparément n'apportent que très peu d'information.

L'approche ascendante permettait de déterminer l'ordre inverse des variables à tester, c'est-à-dire que l'algorithme trouve d'abord la dernière variable à tester (constituant le dernier niveau de l'arbre), puis l'avant-dernière (constituant l'avant-dernier niveau de l'arbre), et ainsi de suite jusqu'à la racine de l'arbre. Cette approche permet en fait d'écarter les variables faussement informatives.

Nous nous proposons donc tout naturellement d'adapter cette approche en ayant une démarche plus locale.

L'adaptation de cette méthode consiste à appliquer l'algorithme ascendant désagrégatif global pour trouver la variable à placer à la racine. Chaque modalité de la variable va donner naissance à un nœud qui peut être vu comme la racine d'un sous-arbre, en prenant en compte la sous-population associée à cette modalité (ce qui revient à prendre en compte le chemin parcouru dans l'arbre). L'algorithme ascendant désagrégatif global peut être réitéré sur cette sous-population afin de trouver la variable à placer au nœud correspondant, ..., et ainsi de suite jusqu'à ce que le critère d'arrêt soit validé, dans ce cas le nœud est une

feuille.

Cette approche permet de conserver les propriétés de l'algorithme ascendant désagrégatif global pour chaque nœud.

À chaque nœud, la méthode consiste à :

$$\min_{X_i \in S_{k-1}} (H(Y/chemin \wedge (S_{k-1} \backslash X_i)))$$

$$\begin{aligned} S_0 &= X \backslash (X_j \in chemin) \\ S_k &= S_{k-1} \backslash X_i \end{aligned}$$

ou, si l'on utilise l'indice $q(Y/S)$ et si l'on considère les sous-populations d'apprentissage :

$$\max_{X_i \in S_{k-1}} (q(Y/(S_{k-1} \backslash X_i)))$$

$$\begin{aligned} S_0 &= X \backslash (X_j \in chemin) \\ S_k &= S_{k-1} \backslash X_i \end{aligned}$$

Nous proposons l'algorithme suivant :

Procédure APPROCHE ASCENDANTE DESAGRÉGATIVE PAR NŒUD(Ω', Res)
  Ω^*, Res^* et S^* sont des variables locales à la procédure

  $Res^* \leftarrow \{\}$; $S^* \leftarrow Res$
  ; boucle
  Répéter
    calculer les $q_{\Omega'}(Y/S^* \backslash X_i)$ avec $X_i \in S^*$
    retenir X_i qui maximise $q_{\Omega'}(Y/S^* \backslash X_i)$
    $S^* \leftarrow S^* \backslash (X_i)$; $Res^* \leftarrow Res^* \times (X_i)$
  Jusqu'à card$(S^*) = 1$
  $Res \leftarrow Res \backslash S^*$
  Pour $j := 1$ à cardM_{X_i} **Faire**
    $\Omega^* \leftarrow \Omega' \cap \{X_i^{-1}(\alpha_j^i)\}$ où $\alpha_j^i \in M_{X_i}$
    Si $(q_{\Omega^*}(Y/X_i) > q_{min})$ ou (card$Res = 0$) **Alors**
      tracer la feuille correspondant à la classe majoritaire pour Ω^*
    Sinon
      APPROCHE ASCENDANTE DESAGRÉGATIVE PAR NŒUD(Ω^*, Res)
    Fin Si
  Fin Pour
Fin

La procédure consiste à utiliser une approche ascendante désagrégative afin de déterminer la variable à tester à chaque nœud, ce qui nous permet de conserver les propriétés de cette approche. Cette variable correspond dans l'algorithme au cas où S^* est un singleton ($\text{card}(S^*) = 1$).

À chaque nœud, la variable la plus discriminante étant déterminée, il suffit de construire les branches donnant naissance à de nouveaux nœuds.

Il suffit alors de réitérer l'opération tant que le critère d'arrêt n'est pas validé.

En reprenant l'exemple précédent, nous obtenons l'arbre de la figure 4.20.

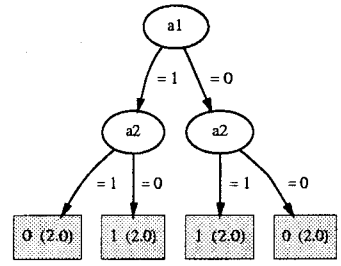

FIG. 4.20: *Arbre obtenu avec l'approche ascendante désagrégative par nœud*

Comme pour l'approche par niveau, l'arbre optimal est obtenu.

4.5 Les méthodes de construction par nœud et par modalité

On s'est intéressé jusqu'à présent à chercher, à un niveau ou à un nœud donné, la variable qui satisfaisait au critère choisi. C'est donc la variable qui « minimise » la quantité d'information qu'il reste à apporter sur Y lorsque l'on connaît la modalité de la variable explicative, et ceci, en moyenne sur la totalité des modalités qu'elle peut prendre.

La quantité d'information moyenne qu'il reste à apporter sur Y une fois que l'on connaît X est représentée par $H(Y/X)$, qui peut être définie par :

$$H(Y/X) = \sum_i p_{i.} \cdot H(Y/X = \alpha_i)$$

$H(Y/X)$ est donc la moyenne sur α_i des $H(Y/X = \alpha_i)$. La grandeur $H(Y/X = \alpha_i)$ représente la quantité d'information moyenne qu'il reste à apporter sur Y sachant que $X = \alpha_i$. Cette quantité représente en fait l'incertitude de Y (notée $H(Y)$), si l'on considère la sous-population d'apprentissage correspondant au chemin : $X = \alpha_i$.

On peut dès lors s'intéresser à la quantité d'information qu'il reste à apporter sur Y une fois que l'on sait que la variable explicative prend une valeur particulière (ou appartient à une partie de l'ensemble des modalités dans le cas d'une variable numérique).

À un nœud donné de l'arbre, on obtient donc un test sur une variable suivant une modalité particulière. Si la réponse à ce test est positive, on continue la progression dans l'arbre dans l'une des branches, sinon, on s'autorise, dans l'autre branche, à tester une autre variable qui apporterait plus d'information, même si l'on ne connaît pas encore la valeur de la variable testée auparavant. Ceci est illustré sur la figure 4.21.

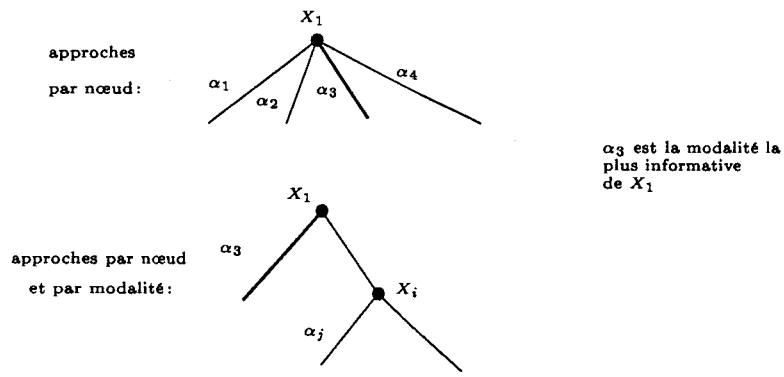

FIG. 4.21: *Approche par nœud et par modalité*

L'approche par nœud testera au pire toutes les modalités de X_1. En adoptant une approche par nœud et par modalité, on pourra ne tester que la modalité de la variable la plus informative (ici α_3).

Pour réaliser cette démarche, il suffirait de minimiser $H(Y/X = \alpha_i)$. Néanmoins, il est nécessaire de « normaliser » ces quantités d'information afin de prendre en compte des sous-populations d'effectif différent. Nous minimiserons donc la quantité suivante :

$$\frac{1}{p_{i.}} \cdot H(Y/X = \alpha_i)$$

Afin de justifier l'utilisation de cette mesure d'information, considérons la quantité :

$$H(Y/X = \alpha_i) = -\sum_j p_{j/i} \cdot \log p_{j/i}$$

La minimisation de cette somme, en regard de l'allure de la fonction $f(p) = -p \cdot \log p$ (figure 4.22) et du fait que la somme de ces probabilités $p_{j/i}$ vaut 1 sur une ligne du tableau de contingence, va nous conduire à privilégier dans celui-ci des lignes pour lesquelles il n'y a qu'une seule valeur non nulle et de plus égale à 1, ou des situations s'en approchant.

Les méthodes de construction de l'arbre de décision par nœud et par modalité consistent à déterminer une modalité α_i (ou un ensemble de modalités) de la variable X_j apportant de l'information. Au nœud considéré, on testera alors si X_j est égal ou différent de α_i. Cette méthode ne consiste pas à tester toutes les modalités de X_j mais un sous-ensemble ; on conditionne donc les modalités de X_j en deux ensembles, ce qui induit une transformation

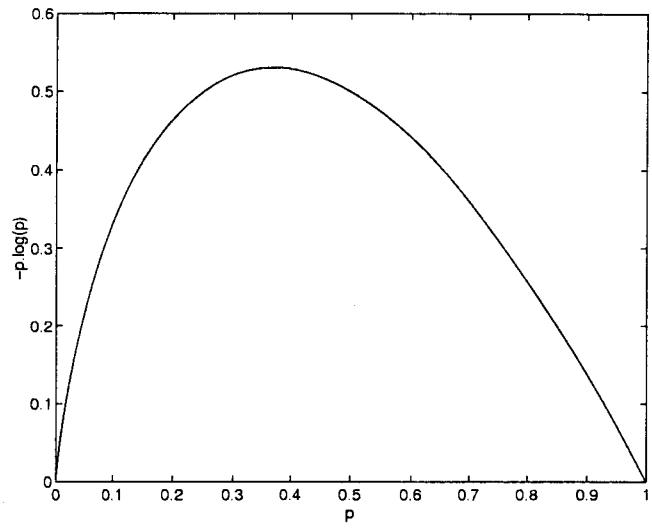

FIG. 4.22: $f(p) = -p \cdot \log p$

binaire de la variable X_j. L'application à des variables quantitatives est directe car il suffit d'adapter un seuil à chaque nœud.

En prenant en compte ce conditionnement, on peut réutiliser une approche descendante agrégative ayant de bonnes performances pour un petit nombre de variables à tester.

L'approche ascendante désagrégative introduit une contrainte combinatoire en général trop lourde à gérer. En effet, il faudrait combiner toutes les transformations binaires de toutes les variables non encore testées, afin de ne garder que celle répondant au critère.

Par contre, nous pouvons introduire une approche descendante désagrégative, qui n'a pas été présentée dans les méthodes de construction de l'arbre par nœud, car elle ne présente qu'un intérêt restreint (voir la figure 4.17).

4.5.1 Approche descendante agrégative par nœud et par modalité

Il s'agit en fait de l'adaptation de l'approche descendante agrégative par nœud. La méthode consiste donc à:

$$\min_{\substack{i \text{ et } j \\ [X_i=\alpha_j] \not\subset chemin}} \left(\frac{H(Y/chemin \wedge (X_i = \alpha_j))}{p_{ij/chemin}}\right)$$

La démarche est en tout point identique à l'approche descendante agrégative par nœud, en considérant toutes les transformations possibles des variables en variables binaires pour lesquelles il suffit de tester si la variable est égale à une modalité (à un seuil dans le cas de variables numériques).

Si on applique cette approche sur notre exemple, nous obtenons l'arbre de la figure 4.23.

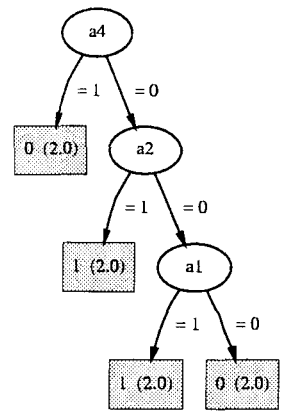

FIG. 4.23: *Arbre obtenu avec l'approche descendante agrégative par nœud et par modalité*

Pour cet exemple, l'assertion la plus informative est $a_4 = 1$. En effet, lorsque $a_4 = 1$, on est sûr que $Y = 1$. Il est donc logique que la variable a_4 apparaisse en haut de l'arbre.

4.5.2 Approche descendante désagrégative par nœud et par modalité

Il s'agit ici de l'adaptation de l'approche descendante désagrégative par niveau.
La méthode consiste alors à:

$$\max_{\substack{i \text{ et } j \\ [X_i=\alpha_j] \notin chemin}} \left(\frac{H(Y/chemin \wedge (S \backslash X_i) \wedge (X_i \neq \alpha_j))}{p_{i\bar{j}/chemin}}\right)$$

S étant l'ensemble des variables non encore testées complètement dans chemin, $p_{i\bar{j}}$ étant la probabilité $p(X_i \neq \alpha_j)$.

En reprenant le même exemple, nous obtenons l'arbre de la figure 4.24.
L'intérêt de ces méthodes n'est pas mis en évidence sur ce petit exemple. Il le sera beaucoup plus dans le chapitre 5 où nous traiterons des tableaux de données plus volumineux, constitués de variables ayant plus de deux modalités.

Ces deux dernières approches peuvent être généralisées en prenant en compte un sous-ensemble de modalités de X_i.

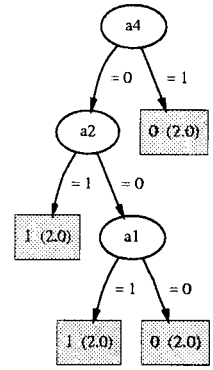

FIG. 4.24: *Arbre obtenu avec l'approche descendante désagrégative par nœud et par modalité*

4.6 Comparaison des méthodes

Lorsque l'utilisateur n'est pas limité concernant le nombre de variables à tester, il est clair qu'il a tout intérêt à utiliser une méthode ascendante désagrégative plutôt qu'une méthode descendante agrégative comme *ID3*. On obtient d'ailleurs en général de meilleurs résultats avec ces méthodes (voir chapitre 5).

L'intérêt de l'approche ascendante désagrégative par nœud par rapport à l'approche par niveau tient dans le fait qu'on ne se limite plus à une variable par niveau. On peut trouver dans le cas de l'approche ascendante désagrégative une variable différente à chaque nœud, et cette approche constitue donc une version affinée de l'approche par niveau.

Les méthodes descendantes désagrégatives n'apportent qu'une solution peu satisfaisante au problème qui consiste à prendre en compte la quantité d'information apportée par un ensemble de variables. Elles donnent donc des résultats moins performants.

L'intérêt des approches par nœud et par modalité se vérifie lorsque les relations que l'on essaie de caractériser entre les variables sont d'un type bien particulier. En effet, lorsque les relations sont plutôt locales, c'est-à-dire lorsque celles-ci sont définies sur certaines modalités des variables explicatives et des variables à expliquer, alors elles donneront de meilleurs résultats. Cette remarque est illustrée dans le paragraphe 5.1.3.

Lorsque les conclusions sont facilement (ou visuellement) séparables (situation (a) de la figure 4.25), les méthodes « globales » sont plus efficaces car les relations en présence sont plutôt « globales ». *A contrario* lorsque les données sont plus hétérogènes et que les relations sont plus contraignantes, *i.e.* lorsque les relations sont plutôt du type : $Y = 1$ si $X_1 = 1$ ou $X_2 = 1$ et $X_3 = 2$ ... (situation (b)) alors les approches de construction de l'arbre par nœud et par modalité sont plus efficaces.

Que ce soit dans le cadre des approches globales ou des approches plus locales, nous pouvons noter que l'introduction de variables supplémentaires permet de disposer de partitions de plus en plus fines de la population d'apprentissage et donc d'approcher de mieux en mieux la partition associée à la variable à expliquer (voir chapitre 2). Cette introduction

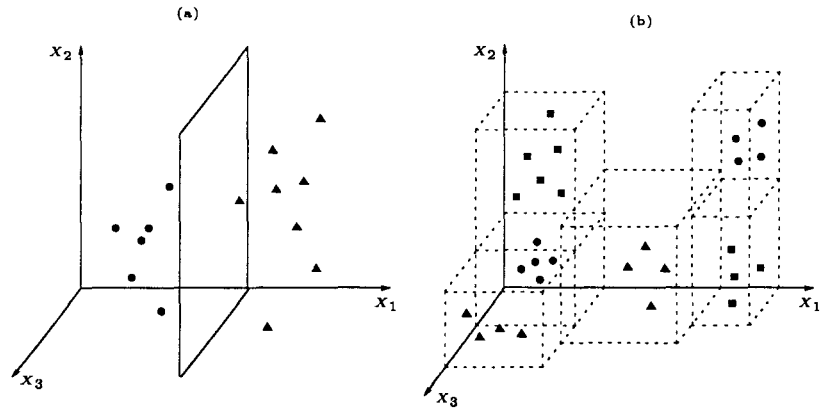

FIG. 4.25: *Données homogènes / hétérogènes*

trouve cependant ses limites dans la diminution de l'effectif des classes de la partition de la population d'apprentissage, qui peut dès lors constituer un critère d'arrêt supplémentaire de nos algorithmes.

Les différents algorithmes ayant été présentés, nous nous proposons dans le chapitre suivant de les valider sur des cas concrets.

Chapitre 5

Validation des méthodes développées sur des cas concrets

Nous nous proposons dans ce chapitre de valider les méthodes sur des cas concrets.

Dans un premier temps, nous testerons ces méthodes sur quelques tableaux de données issus de la base de données UCI Data Repository. Le choix de ces tableaux a été guidé par le fait que de nombreuses méthodes d'apprentissage issues de la littérature les utilisent afin de comparer les résultats obtenus à travers diverses publications. Ces tableaux représentent un peu une référence dans le domaine. Nous comparerons ainsi les méthodes par niveaux afin de faire ressortir les performances de l'approche ascendante désagrégative. Nous comparerons alors les méthodes par nœud entre elles, ainsi qu'avec la meilleure des approches par niveau, et là encore, les méthodes basées sur une approche ascendante désagrégative obtiennent les meilleurs résultats.

Afin de se replacer dans le cadre du diagnostic, nous nous proposons dans un deuxième temps, de tester *C4.5* ainsi que la meilleure méthode ascendante désagrégative, à savoir l'approche ascendante désagrégative par nœud, sur une base de données issues d'un moteur asynchrone. L'objectif de cette étude est de discriminer l'état de fonctionnement normal du moteur, ainsi qu'un état de dysfonctionnement, à savoir un dysfonctionnement de type « capteur », une dérive de la résistance dite « rotorique » étant considérée comme un fonctionnement normal.

5.1 Application des méthodes sur des tableaux de données issus de la littérature

5.1.1 Présentation des tableaux

Afin de tester les différentes méthodes et de les comparer, nous allons utiliser quelques tableaux de données issus de la base de données UCI Data Repository [MM96].

Les principales caractéristiques de ces tableaux sont décrites ci-dessous :

Nom	Nombre d'observations (apprentissage)	Nombre de variables explicatives	Nombre de classes
monks-1	124	6	2
monks-2	169	6	2
monks-3nb	122	6	2
monks-3	122	6	2
hayes-roth	132	4	3

TAB. 5.1: *Caractéristiques des tableaux utilisés*

Dans ce tableau, le nombre de classes représente le nombre de modalités de la variable à expliquer Y.

Le tableau « hayes-roth » représente les résultats d'une enquête sur une population constituée de 132 personnes. Les variables explicatives sont les *loisirs* (3 modalités), l'*âge* (4 modalités), le *niveau d'éducation* (4 modalités) et la *situation maritale* (4 modalités). Ces individus sont répartis en 3 classes de la façon suivante :

- la variable *loisirs* n'intervient pas dans la classification ;
- l'individu appartient à la classe 3 si l'un des trois attributs *âge*, *niveau d'éducation*, *situation maritale* vaut 4 ;
- l'individu appartient à la classe 1 si le nombre de 1 est supérieur au nombre de 2 pour ces trois attributs ;
- l'individu appartient à la classe 2 si le nombre de 2 est supérieur au nombre de 1 pour ces trois attributs ;
- si le nombre de 1 est égal au nombre de 2 pour ces trois attributs l'individu sera alors aléatoirement de classe 1 ou 2.

Les tableaux « monks » représentent, quant à eux, des relations bien définies entre les différentes variables. On dispose en fait de trois tableaux d'individus tirés aléatoirement dans une même base de données composées de 432 individus au total qui représentent tous les cas possibles. Ils sont alors répartis pour chacun des tableaux en deux classes suivant la règle définie. Les six attributs décrivent des robots. La variable X_1 qui est la « forme de la tête » a trois modalités (ronde, carrée, octogonale). La variable X_2 est la « forme du corps » et a également trois modalités (ronde, carrée, octogonale). La troisième variable X_3 a deux modalités et vaut « yes » si le robot sourit et « no » dans le cas contraire. X_4 décrit l'objet que tient le robot (épée, ballon, drapeau), X_5 est la « couleur de la veste » du robot (rouge, jaune, verte, bleue). Enfin, la variable X_6 vaut « yes » si le robot porte une cravate, et « no » dans le cas contraire.

Le tableau « monks-1 » représente la règle « $Y = 1$ lorsque $X_1 = X_2$ ou $X_5 = 1$ ».

Le tableau « monks-2 » représente la règle « $Y = 1$ lorsque l'on a exactement deux des assertions : $\{X_1 = 1, X_2 = 1, X_3 = 1, X_4 = 1, X_5 = 1, X_6 = 1\}$ ».

Le tableau « monks-3nb » représente la règle « $Y = 1$ lorsque $(X_3 = 3$ et $X_4 = 1)$ ou $(X_5 \neq 4$ et $X_2 \neq 3)$ ».

Le tableau « monks-3 » est le tableau « monks-3nb » avec 5 % de bruit (sur 100 valeurs de Y, 5 sont erronées) superposé à la variable à expliquer.

Ces tableaux de données sont accompagnés de fichiers tests composés de 432 individus pour les tableaux « monks » (tous les cas possibles) et 28 individus pour le tableau « hayes-roth », que l'on peut utiliser pour comparer les différentes méthodes proposées. Pour ce faire, nous allons construire les arbres de décision pour chaque approche en utilisant les ensembles d'apprentissage. Pour ces essais, nous fixerons le seuil q_{min} à 100 % et le seuil nb_{max} à sa valeur maximale (card$X$). Nous n'effectuerons pas d'élagage de l'arbre non plus.

Sur la population d'apprentissage, nous évaluerons la « taille » de l'arbre en calculant le « nombre moyen de questions » nécessaires pour aboutir à une feuille. Puis, nous calculerons la « précision » de l'arbre construit sur la population test. Enfin, nous représenterons pour chaque tableau, la précision obtenue en fonction du nombre de questions moyen. Afin d'obtenir un bon compromis entre complexité et précision du modèle obtenu, l'approche étudiée doit se situer le plus haut et le plus à gauche possible du graphe.

5.1.2 Comparaison des approches par niveau

En ce qui concerne les approches par niveau, nous obtenons les résultats suivants :

Approche \ Tableau		hayes-roth	monks-1	monks-2	monks-3nb	monks-3
sans effet mémoire	Précision	75 % ± 16.33	92.59 %±2.47	67.82 %±4.41	96.76 %±1.67	93.52 %±2.32
	NQMoy	5.05 ± 0.13	5.46 ± 0.20	7.27 ± 0.14	4 ± 0.14	4.82 ± 0.19
desc. agrég.	Précision	75 % ± 16.33	100 %	68.06 %±4.40	100 %	93.52 %±2.32
	NQMoy	5.05 ± 0.12	4.52 ± 0.11	6.82 ± 0.15	3.89 ± 0.07	4.67 ± 0.14
desc. désagrég.	Précision	75 % ± 16.33	79.17 %±3.83	67.82 %±4.41	100 %	93.52 %±2.32
	NQMoy	5.05 ± 0.13	6 ± 0.20	7.24 ± 0.15	3.53 ± 0.10	4.80 ± 0.22
asc. agrég.	Précision	75 % ± 16.33	92.59 %±2.47	67.82 %±4.41	96.76 %±1.67	94.44 %±2.16
	NQMoy	5.05 ± 0.12	5.77 ± 0.18	7.24 ± 0.15	3.64 ± 0.20	4.82 ± 0.22
asc. désagrég.	Précision	89.29 %±11.67	100 %	68.06 %±4.40	100 %	93.52 %±2.32
	NQMoy	5.05 ± 0.13	4.44 ± 0.12	6.82 ± 0.15	3.89 ± 0.07	4.67 ± 0.14

TAB. 5.2: *Résultats des approches par niveau*

Ces résultats sont donnés avec un intervalle de confiance à 95 %, en supposant que les données sont distribuées normalement. Néanmoins, nous ne les représenterons pas sur les graphes afin de ne pas surcharger ces derniers.

- Tableau « hayes-roth » (figure 5.1) :

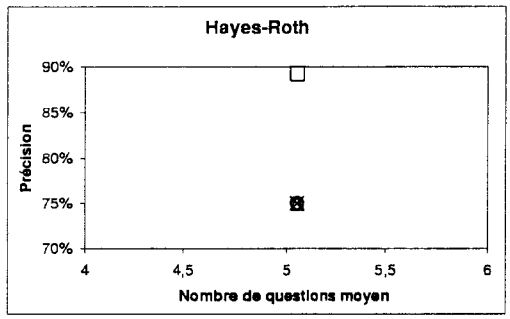

FIG. 5.1: *Résultats des approches par niveau sur le tableau « hayes-roth »*

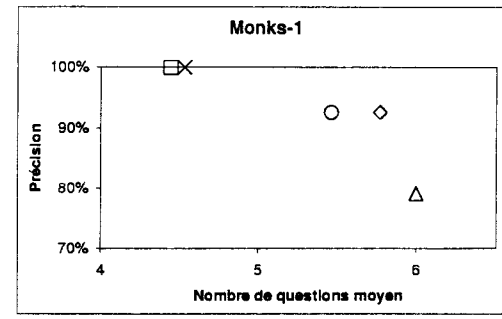

FIG. 5.2: *Résultats des approches par niveau sur le tableau « monks-1 »*

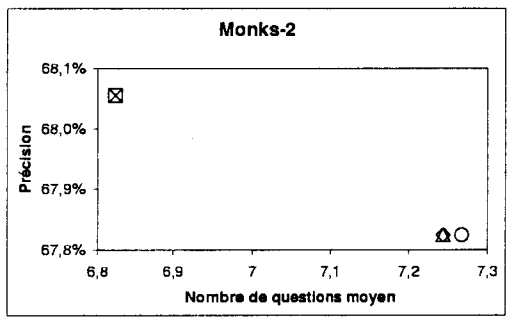

FIG. 5.3: *Résultats des approches par niveau sur le tableau « monks-2 »*

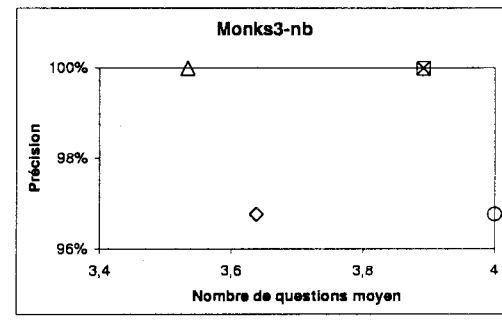

FIG. 5.4: *Résultats des approches par niveau sur le tableau « monks-3nb »*

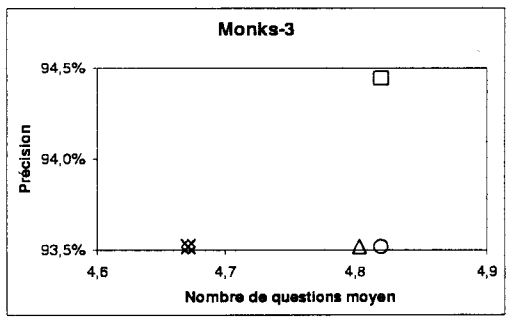

FIG. 5.5: *Résultats des approches par niveau sur le tableau « monks-3 »*

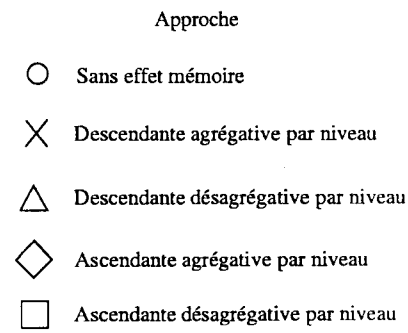

FIG. 5.6: *Légende*

Pour ce tableau, l'algorithme qui présente les meilleures performances est l'approche ascendante désagrégative. En effet, elle obtient une précision de 90 % contre 75 %

pour les autres méthodes.

L'ordre de test des variables obtenu par les différents algorithmes est récapitulé dans le tableau 5.3.

Algorithme	Ordre des variables testées
sans effet mémoire	$X_3\ X_2\ X_4\ X_1$
descendant agrégatif	$X_2\ X_3\ X_4\ X_1$
descendant désagrégatif	$X_3\ X_2\ X_4\ X_1$
ascendant agrégatif	$X_3\ X_2\ X_4\ X_1$
ascendant désagrégatif	$X_4\ X_2\ X_3\ X_1$

TAB. 5.3: *Ordre de test des variables pour le tableau hayes-roth*

- Tableau « monks-1 » (figure 5.2) :

Pour ce dernier, l'approche ascendante désagrégative obtient également les meilleures performances, suivie de près par l'approche descendante agrégative. Toutes deux ont une précision de 100 % sur la population test.

Algorithme	Ordre des variables testées
sans effet mémoire	$X_5\ X_1\ X_4\ X_2\ X_3\ X_6$
descendant agrégatif	$X_5\ X_1\ X_2\ X_6\ X_4\ X_3$
descendant désagrégatif	$X_1\ X_5\ X_4\ X_6\ X_2\ X_3$
ascendant agrégatif	$X_1\ X_5\ X_4\ X_2\ X_3\ X_6$
ascendant désagrégatif	$X_1\ X_2\ X_5\ X_3\ X_4\ X_6$

TAB. 5.4: *Ordre de test des variables pour le tableau monks-1*

Dans le tableau 5.4 donnant l'ordre des variables testées, les zones grisées indiquent que les variables concernées ne sont pas utilisées car à ce niveau, toutes les observations sont discriminées. À la vue de ce tableau, on remarque bien que les approches qui n'ont pas obtenu une précision de 100 % sont celles qui ont choisi d'autres variables que X_1, X_2 et X_5. De plus, on vérifie bien dans cet exemple que l'approche ascendante désagrégative a placé en haut de l'arbre les deux variables qui apportent de l'information commune sur Y bien qu'elles en apportent peu si on les prend en compte séparément. En effet, ici $Y = 1$ lorsque $X_1 = X_2$. L'approche descendante agrégative a privilégié la variable qui apporte beaucoup d'information à elle seule. Quant aux deux approches restantes, on peut également vérifier qu'elles ont un bon comportement au début qui se dégrade vers la fin, ou inversement, ce qui conduit à séparer les deux variables X_1 et X_2.

- Tableau « monks-2 » (figure 5.3) :

  On observe une assez faible dispersion des résultats avec des performances légèrement meilleures pour les approches ascendantes désagrégatives et descendantes agrégatives.

Algorithme	Ordre des variables testées
sans effet mémoire	$X_5\ X_4\ X_6\ X_1\ X_2\ X_3$
descendant agrégatif	$X_5\ X_3\ X_2\ X_4\ X_6\ X_1$
descendant désagrégatif	$X_5\ X_6\ X_4\ X_1\ X_2\ X_3$
ascendant agrégatif	$X_5\ X_6\ X_4\ X_1\ X_2\ X_3$
ascendant désagrégatif	$X_5\ X_3\ X_2\ X_4\ X_6\ X_1$

TAB. 5.5: *Ordre de test des variables pour le tableau monks-2*

Comme le montre le tableau 5.5, on est obligé d'utiliser dans l'arbre toutes les variables. En effet, pour chaque individu, il faut connaître les modalités de chaque variable afin d'expliquer Y. Il est donc logique que les approches par niveau obtiennent des résultats à peu près similaires pour ce type de données. Néanmoins, la connaissance de toutes les modalités des variables explicatives n'est pas nécessaire. Seule la modalité « 1 » de ces variables est importante. Nous verrons dans le paragraphe suivant l'intérêt d'une approche par nœud et par modalité pour de tels tableaux.

- Tableau « monks-3nb » (figure 5.4) :

  La dispersion des résultats est ici également assez faible, avec une précision de 100 % pour la plupart des approches.

  On peut tout de même remarquer sur le tableau 5.6 la faiblesse d'une approche ascendante agrégative qui dissocie une fois de plus deux variables importantes et place de ce fait, la variable X_4 au bas de l'arbre de décision.

Algorithme	Ordre des variables testées
sans effet mémoire	$X_2\ X_5\ X_1\ X_3\ X_6\ X_4$
descendant agrégatif	$X_2\ X_5\ X_4\ X_6\ X_3\ X_1$
descendant désagrégatif	$X_5\ X_2\ X_4\ X_1\ X_3\ X_6$
ascendant agrégatif	$X_5\ X_2\ X_1\ X_6\ X_3\ X_4$
ascendant désagrégatif	$X_2\ X_5\ X_4\ X_1\ X_3\ X_6$

TAB. 5.6: *Ordre de test des variables pour le tableau monks-3nb*

- Tableau « monks-3 » (figure 5.5) :

  En présence de bruit, la dispersion des résultats est encore réduite, et on perd en précision. Ce résultat semble cohérent, car plus un tableau est bruité et plus on

perd d'information pertinente. Utiliser l'une ou l'autre des méthodes n'apportera par conséquent que des résultats peu satisfaisants.

Ceci se traduit également dans la structure des arbres obtenus. En effet, cette fois-ci, d'autres variables que X_2, X_4 ou X_5 sont employées (tableau 5.7).

Algorithme	Ordre des variables testées
sans effet mémoire	$X_2\ X_5\ X_1\ X_6\ X_4\ X_3$
descendant agrégatif	$X_2\ X_5\ X_4\ X_1\ X_3\ X_6$
descendant désagrégatif	$X_5\ X_2\ X_1\ X_4\ X_6\ X_3$
ascendant agrégatif	$X_5\ X_2\ X_4\ X_6\ X_1\ X_3$
ascendant désagrégatif	$X_2\ X_5\ X_4\ X_1\ X_6\ X_3$

TAB. 5.7: *Ordre de test des variables pour le tableau monks-3*

On peut donc dire que lorsque les différences de performances sont significatives, la meilleure approche est l'ascendante désagrégative, suivie ensuite par l'approche descendante agrégative. Ce résultat confirme bien l'intuition exposée au chapitre 4.

5.1.3 Comparaison des approches par nœud

Nous nous proposons d'utiliser sur les mêmes tableaux les approches par nœud ainsi que l'approche ascendante désagrégative par niveau afin de les comparer entre elles.

Les résultats obtenus sont résumés dans le tableau 5.8.

Approche \ Tableau		hayes-roth	monks-1	monks-2	monks-3nb	monks-3
asc. désag. par niv.	Précision	89.29 %±11.67	100 %	68.06 %±4.40	100 %	93.52 %±2.32
	NQMoy	5.05 ± 0.13	4.44 ± 0.12	6.82 ± 0.15	3.89 ± 0.07	4.67 ± 0.14
desc. agr. par nœ.	Précision	82.14 %±14.45	85.42 %±3.33	68.06 %±4.40	100 %	94.91 %±2.08
	NQMoy	5.05 ± 0.12	5.15 ± 0.23	6.48 ± 0.16	3.89 ± 0.07	4.39 ± 0.15
desc. désag. par nœ.	Précision	64.29 %±18.07	74.77 %±4.10	70.14 %±4.32	97.22 %±1.55	92.13 %±2.54
	NQMoy	5.24 ± 0.12	6.63 ± 0.18	7.09 ± 0.14	3.59 ± 0.16	4.80 ± 0.21
asc. désag. par nœ.	Précision	82.14 %±14.45	100 %	67.59 %±4.42	100 %	93.98 %±2.25
	NQMoy	5.05 ± 0.12	4.44 ± 0.12	6.41 ± 0.15	3.89 ± 0.07	4.40 ± 0.15
desc. agr. par nœ. et mod.	Précision	85.71 %±13.20	75.69 %±4.05	79.63 %±3.80	98.61 %±1.10	95.14 %±2.03
	NQMoy	6.55 ± 0.22	5.75 ± 0.36	6.17 ± 0.19	2.65 ± 0.12	4.55 ± 0.25
desc. dés. par nœ. et mod.	Précision	89.29 %±11.67	75.69 %±4.05	79.63 %±3.80	98.61 %±1.10	94.91 %±2.08
	NQMoy	6.33 ± 0.22	5.98 ± 0.34	6.17 ± 0.19	3.41 ± 0.10	4.52 ± 0.24

TAB. 5.8: *Résultats des approches par nœud et ascendante désagrégative par niveau*

– Tableau « hayes-roth » (figure 5.7) :

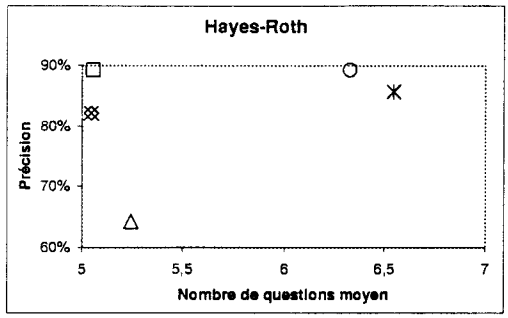

FIG. 5.7: *Résultats des approches par nœud sur le tableau « hayes-roth »*

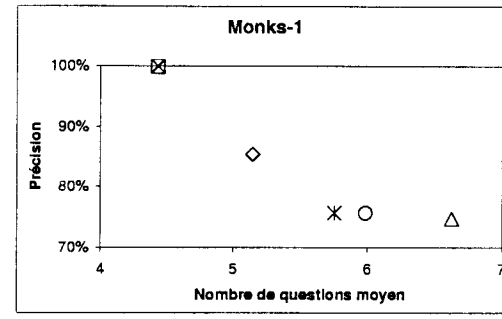

FIG. 5.8: *Résultats des approches par nœud sur le tableau « monks-1 »*

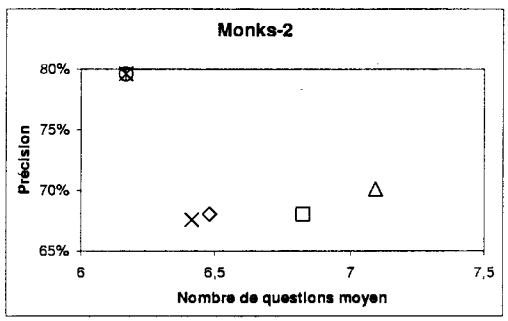

FIG. 5.9: *Résultats des approches par nœud sur le tableau « monks-2 »*

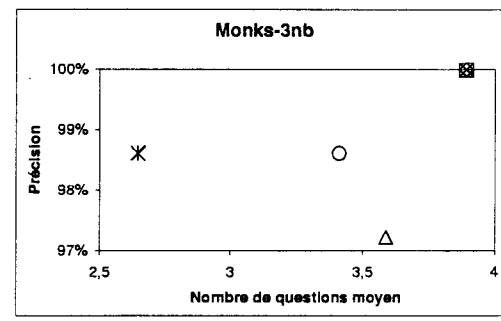

FIG. 5.10: *Résultats des approches par nœud sur le tableau « monks-3nb »*

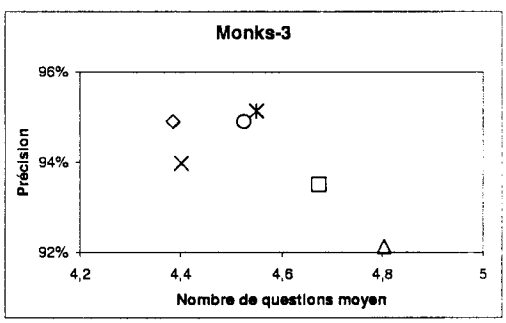

FIG. 5.11: *Résultats des approches par nœud sur le tableau « monks-3 »*

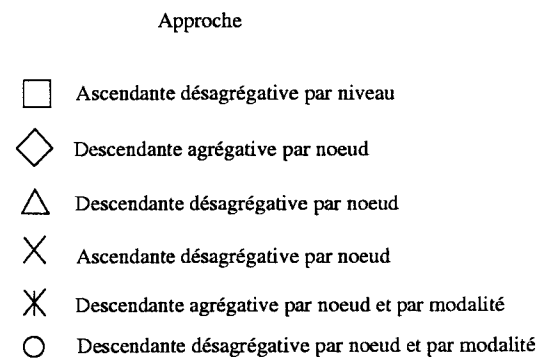

FIG. 5.12: *Légende*

Pour ce tableau, les approches par nœud ne sont pas meilleures que l'approche ascendante désagrégative par niveau. Mais l'approche ascendante désagrégative par nœud,

ainsi que l'approche descendante agrégative par nœud ont tout de même de bonnes performances.

- Tableau « monks-1 » (figure 5.8) :

  Dans ce cas, les meilleures approches sont clairement l'approche ascendante désagrégative par niveau, et l'approche ascendante désagrégative par nœud.

  À titre d'illustration, nous donnons en annexe, les arbres obtenus par l'approche descendante agrégative, figure C.1, et par l'approche ascendante désagrégative, figure C.2. On remarque aisément que l'arbre obtenu par l'approche ascendante désagrégative n'utilise que les variables nécessaires à l'explication, tandis que l'approche descendante agrégative propose à certains nœuds des tests sur des variables superflues telles que X_6, X_4 ou encore X_3, et ceci vers le haut de l'arbre, ce qui entraîne une plus forte complexité de celui-ci.

- Tableau « monks-2 » (figure 5.9) :

  Il est clair, dans ce cas, que les deux approches par nœud et par modalité sont les meilleures. En effet, ces approches permettent de ne tester que la modalité des variables qui apporte le plus d'information. Dans cet exemple, cette approche est particulièrement intéressante dans la mesure où l'on est en présence de relations fortement liées à la modalité « 1 » des variables.

- Tableau « monks-3nb » (figure 5.10) :

  Pour ce tableau, les résultats sont moins significatifs, mais on peut noter les bonnes performances en ce qui concerne la taille de l'arbre, des approches par nœud et par modalité.

- Tableau « monks-3 » (figure 5.11) :

  En présence de bruit, la dispersion des résultats est encore réduite.

On peut donc remarquer que de manière générale les approches ascendantes désagrégatives (l'approche par niveau ou l'approche par nœud) obtiennent les meilleurs résultats, en particulier lorsqu'il s'agit de retrouver des relations globales entre variables, comme c'est le cas pour le tableau « monks-1 ».

Les approches par nœud et par modalité, quant à elles, se distinguent lorsqu'il s'agit de contraintes sur certaines modalités, comme dans le cas du tableau « monks-2 ».

Le choix de l'algorithme peut alors être guidé par la connaissance *a priori* que possède l'utilisateur sur le type de relations (contraintes sur les modalités ou relations globales sur les variables) entre les variables.

5.2 Application des méthodes sur une base de données issues d'un moteur asynchrone

5.2.1 Présentation du problème

On considère un moteur asynchrone sous les hypothèses de fonctionnement suivantes :

- le circuit magnétique est non saturé ;
- la densité de courant est considérée uniforme dans la section des conducteurs élémentaires ;
- on ne considère que le premier harmonique d'espace dans la distribution des forces magnétomotrices ;
- les enroulements du rotor sont en court-circuit.

Le moteur est défini par les intensités, les flux et les tensions aux trois enroulements a, b et c du rotor et du stator, soit un total de 18 variables. On peut réduire la complexité du système en considérant un changement de repère, la transformation de Park. On associe aux trois grandeurs a, b et c (qu'il s'agisse des courants, tensions ou flux) des grandeurs d, q et o relatives au stator par :

$$\begin{bmatrix} \xi_{sd} \\ \xi_{sq} \\ \xi_{so} \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos\theta_s & \cos(\theta_s - \frac{2\pi}{3}) & \cos(\theta_s - \frac{4\pi}{3}) \\ \sin\theta_s & \sin(\theta_s - \frac{2\pi}{3}) & \sin(\theta_s - \frac{4\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \xi_{sa} \\ \xi_{sb} \\ \xi_{sc} \end{bmatrix}$$

La même opération est effectuée sur les grandeurs relatives au rotor :

$$\begin{bmatrix} \xi_{rd} \\ \xi_{rq} \\ \xi_{ro} \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos\theta_r & \cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r - \frac{4\pi}{3}) \\ \sin\theta_r & \sin(\theta_r - \frac{2\pi}{3}) & \sin(\theta_r - \frac{4\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \xi_{ra} \\ \xi_{rb} \\ \xi_{rc} \end{bmatrix}$$

Les angles θ_s et θ_r définissent le choix du repère (d, q). Ils sont reliés par la relation :

$$\omega_r = (\frac{d\theta_s}{dt}) - (\frac{d\theta_r}{dt})$$

où ω_r désigne la vitesse électrique de rotation du rotor. Elle est reliée à la vitesse mécanique du rotor Ω par la relation :

$$\omega_r = p\Omega$$

où p désigne le nombre de paires de pôles de la machine.

Une fois dans le repère (d, q) les équations de la machine s'écrivent avec la convention récepteur pour les signes des courants et tensions aux bornes du stator et du rotor :

$$\begin{cases} v_{sd} = \frac{d\phi_{sd}}{dt} - \phi_{sq} \cdot \frac{d\theta_s}{dt} + R_s \cdot i_{sd} \\ v_{sq} = \frac{d\phi_{sq}}{dt} + \phi_{sd} \cdot \frac{d\theta_s}{dt} + R_s \cdot i_{sq} \\ 0 = \frac{d\phi_{rd}}{dt} - \phi_{rq} \cdot \frac{d\theta_r}{dt} + R_r \cdot i_{rd} \\ 0 = \frac{d\phi_{rq}}{dt} + \phi_{rd} \cdot \frac{d\theta_r}{dt} + R_r \cdot i_{rq} \end{cases} \quad (5.1)$$

avec les relations liant les différents flux aux différents courants :

$$\begin{bmatrix} \phi_{sd} \\ \phi_{rd} \\ \phi_{sq} \\ \phi_{rd} \end{bmatrix} = \begin{bmatrix} L_s & M & 0 & 0 \\ M & L_r & 0 & 0 \\ 0 & 0 & L_s & M \\ 0 & 0 & M & L_r \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{rd} \\ i_{sq} \\ i_{rq} \end{bmatrix}$$

et le couple électromagnétique :

$$\begin{aligned} C_e &= p \cdot [\phi_{sd} \cdot i_{sq} - \phi_{sq} \cdot i_{sd}] \\ &= p \cdot \tfrac{M}{L_r} \cdot [\phi_{rd} \cdot i_{sq} - \phi_{rq} \cdot i_{sd}] \end{aligned}$$

Les angles de Park sont choisis de telle manière que le repère soit solidaire du champ tournant et que ϕ_{rq} soit égal à 0. En supprimant ϕ_{sd} et ϕ_{sq} du système 5.1 et en respectant la relation :

$$\omega = \omega_s - \omega_r$$

Le système suivant est obtenu :

$$\begin{cases} V_d = R_s i_{sd} + \sigma L_s \frac{d}{dt} i_{sd} + \frac{M}{L_r} \frac{d}{dt} \phi_{rd} - \sigma L_s \omega_s i_{sq} \\ V_q = R_s i_{sq} + \sigma L_s \frac{d}{dt} i_{sq} + \omega \frac{M}{L_r} \phi_{rd} + \sigma L_s \omega_s i_{sd} \\ M i_{sd} = \phi_{rd} + \frac{L_r}{R_r} \frac{d}{dt} \phi_{rd} \\ \omega_s = p\omega + \frac{M R_r}{L_r} \frac{i_{sq}}{\phi_{rd}} \\ C_e = p \frac{M}{L_r} \phi_{rd} i_{sq} \\ J \frac{d}{dt} \Omega = C_e - C_r - K\Omega \end{cases}$$

Avec :

- V_d projection de la tension d'alimentation sur l'axe d
- V_q projection de la tension d'alimentation sur l'axe q
- i_{sd} projection du courant statorique sur l'axe d
- i_{sq} projection du courant statorique sur l'axe q
- ϕ_{rd} projection du flux rotorique sur l'axe d
- ω_s pulsation statorique, ω vitesse électrique rotorique, Ω vitesse mécanique $\Omega = p\omega$
- C_e couple électromagnétique

- R_s et L_s résistance et inductance statorique
- R_r et L_r résistance et inductance rotorique
- M inductance mutuelle stator/rotor
- $\sigma(sig)$ coefficient de dispersion : $\sigma = 1 - \frac{M^2}{L_r Ls}$

Un modèle mathématique de la machine asynchrone sur la base de ce système d'équations et de sa commande vectorielle a été construit et est disponible sur le WEB à l'adresse : http://lailp2pc2.univ-lille1.fr/~vincent/. Des données de simulation y sont également disponibles.

À partir de ce modèle, il est possible de récupérer durant la simulation, différentes grandeurs telles que la vitesse de rotation du moteur, les trois tensions statoriques notées v_a, v_b et v_c, ainsi que les trois courants statoriques notés i_a, i_b et i_c. Ces six dernières grandeurs sont obtenues par une transformation de Park inverse à partir des tensions v_d et v_q d'une part, et des courants i_d et i_q d'autre part.

Dans le cadre de la surveillance de systèmes non linéaires appliquée ici à la machine asynchrone, des relations de redondance analytique peuvent être trouvées [CV97, CVCCS99]. En particulier, on veut détecter un défaut sur le capteur de vitesse. Dans ces travaux, deux résidus sont calculés à partir des grandeurs mesurées citées ci-dessus. Comme le montrent les figures 5.13 à 5.16, il est possible de détecter une défaillance du capteur de vitesse (biais de la forme de la figure 5.19 sur 3s.). Le problème est que ces deux résidus sont également sensibles à une variation de la résistance rotorique (figures 5.17 et 5.18). Et la variation de ces résidus est similaire pour les deux types de pannes (défaut du capteur de vitesse, variation de la résistance rotorique). Or, la variation de la résistance est inévitable lors d'un fonctionnement normal du moteur à cause des variations de température. Il s'avère donc difficile de dire s'il y a effectivement défaillance sur le capteur de vitesse ou simplement une variation de la résistance. C'est ce problème de discrimination que nous nous proposons de résoudre.

On se propose alors d'effectuer un apprentissage de ces deux modes de fonctionnement, afin de les distinguer, à partir des variables précitées, et des deux résidus.

5.2.2 Constitution de la population d'apprentissage Ω et des populations tests

Afin de constituer la population d'apprentissage, nous avons procédé de la façon suivante :

Nous avons choisi de prendre comme vitesse de référence, la vitesse nominale qui est de 1 415 tr/mn. La simulation dure 20 s. et on obtient une population de 20 000 observations. Pendant ces 20 s., il y a d'abord une phase de fonctionnement normal, puis un défaut sur le capteur de vitesse seul ($vitesse \rightarrow vitesse + 10\ tr/mn$), puis à nouveau fonctionnement normal, puis variation de la résistance ($R_r \rightarrow 2 \times R_r$), puis, en plus de la variation de

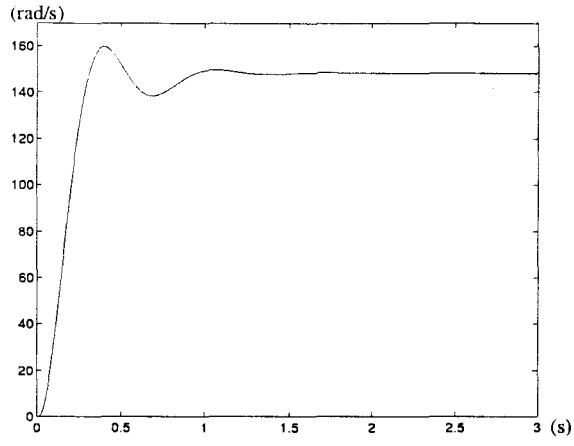

FIG. 5.13: *Vitesse du moteur sans défaut sur le capteur.*

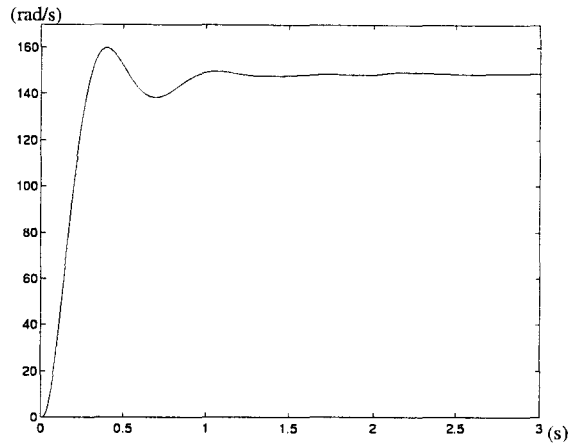

FIG. 5.14: *Vitesse du moteur avec défaut sur le capteur à partir de $t = 2$ s.*

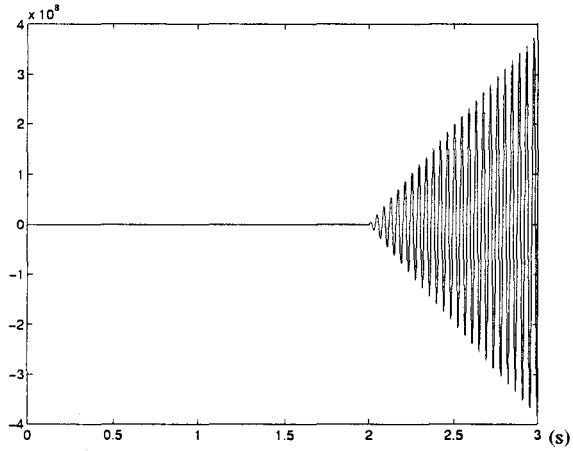

FIG. 5.15: *Résidu 1 (défaut capteur)*

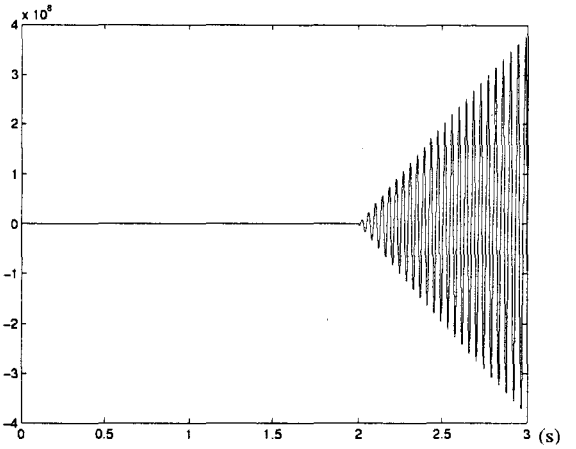

FIG. 5.16: *Résidu 2 (défaut capteur)*

la résistance, un défaut sur le capteur vitesse, et enfin variation de la résistance seule. Le défaut sur le capteur vitesse est ajouté à la sortie du moteur avant le bouclage du système. La variation de la résistance se fait par la variation d'un paramètre du bloc moteur. La forme de ces défauts est donnée sur les figures 5.19 et 5.20.

Afin de prendre en compte la dynamique du système nous avons considéré pour la constitution du tableau initial des données, les « *dérivées* » des variables v_a, v_b, v_c, i_a, i_b, i_c et de la vitesse Ω, notées dv_a, dv_b, dv_c, di_a, di_b, di_c et $d\Omega$, approximées par $dx = x(t) - x(t-1)$ où x est une de ces sept variables. À ces dernières on a ajouté pour chaque observation les deux résidus rs_1 et rs_2 dont le calcul est détaillé dans [CVCCS99]. Enfin, on a ajouté à ce tableau, la variable à expliquer *Mod* qui est le mode de fonctionnement du moteur. Elle vaut 0 en fonctionnement normal et 1 lorsqu'il y a un défaut sur le capteur de vitesse.

À partir de ces données, un arbre de décision a été construit à l'aide de *C4.5*, et un autre à l'aide de l'approche ascendante désagrégative par nœud. Des populations tests

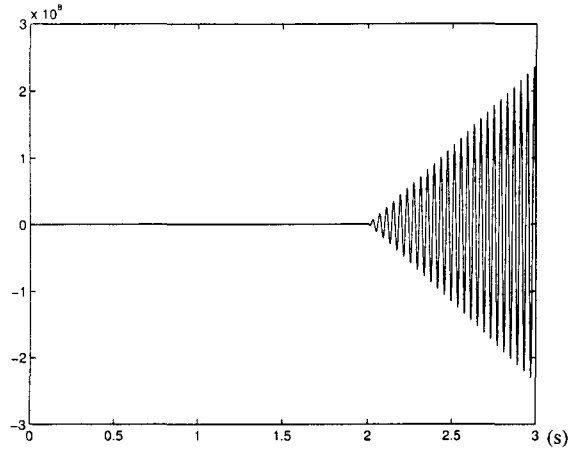

FIG. 5.17: *Résidu 1 (variation de la résistance à partir de $t = 2$ s)*

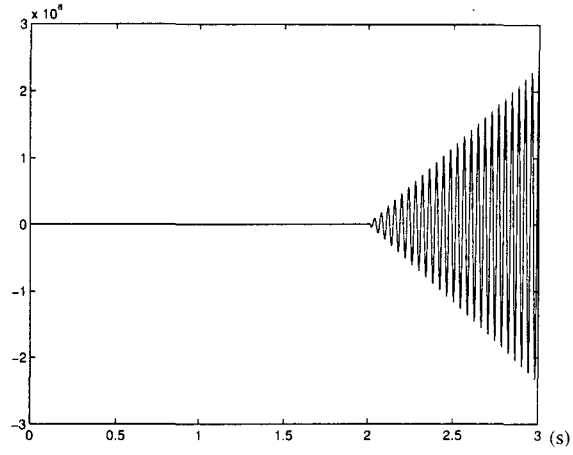

FIG. 5.18: *Résidu 2 (variation de la résistance à partir de $t = 2$ s)*

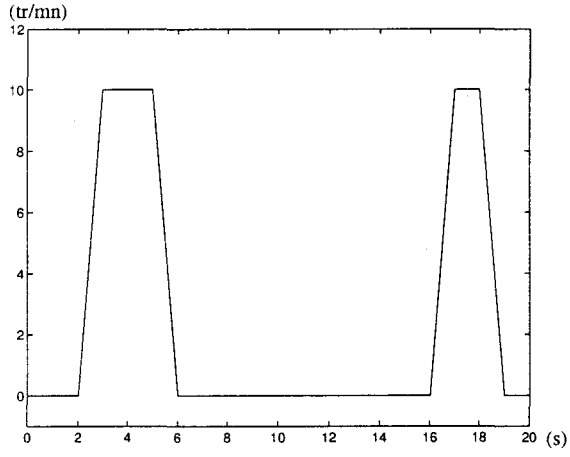

FIG. 5.19: *Forme du défaut sur le capteur vitesse*

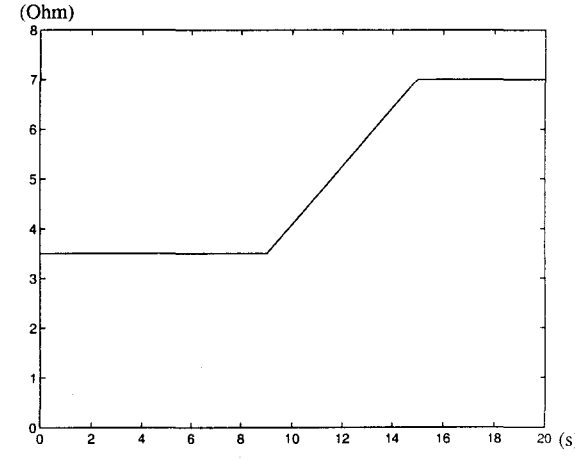

FIG. 5.20: *Forme de la variation de la résistance*

ont ensuite été créées de manière à valider le modèle obtenu. Ces populations tests sont constituées de 10 000 points chacune (10 s. de simulation dont 5 s. de défauts). Cette série de populations tests est composée de deux tableaux comprenant une variation de la résistance rotorique seule ($R_r \rightarrow 1.5 \times R_r$ et $R_r \rightarrow 2 \times R_r$), trois tableaux comprenant un défaut sur le capteur de vitesse seul ($vitesse \rightarrow vitesse + 3\ tr/mn$, $vitesse \rightarrow vitesse + 5\ tr/mn$ et $vitesse \rightarrow vitesse + 20\ tr/mn$) et six tableaux comprenant un défaut sur le capteur de vitesse et une variation de la résistance rotorique (combinaison des défauts précédents $\Rightarrow$ six tableaux). Ces 11 tableaux ont été construits pour une vitesse de référence de 1 415 tr/mn. La même opération a été réalisée pour des vitesses de référence de 1 500 tr/mn et de 900 tr/mn.

5.2.3 Analyse des résultats

En fait, quatre arbres ont été construits (deux pour chaque approche), un arbre avant *élagage*, et un arbre après *élagage*. La technique d'élagage est celle utilisée par *C4.5* et consiste à remplacer à un nœud donné le sous-arbre associé par une feuille lorsque le pourcentage d'erreur de prédiction est plus grand (avec une certaine tolérance prédéfinie) avec le sous-arbre qu'avec la feuille. L'élagage de l'arbre se justifie par le fait que, en général les méthodes d'induction par arbre de décision construisent des arbres qui tentent de refléter parfaitement les données d'apprentissage, quitte à compliquer exagérément la structure de l'arbre et compromettre ainsi la précision de tels modèles sur des populations d'individus non observés. Cet élagage permet de satisfaire le compromis « précision-simplicité ». En ce qui concerne les paramètres d'élagage, nous avons choisi les paramètres par défaut de C4.5, (« pruning confidence level » = 25 %).

Les arbres ainsi obtenus ont été testés sur les populations tests, et le pourcentage d'erreur a été calculé pour chacune d'entre elles.

Il est également important de souligner que pour cet exemple, les variables sont continues. Nous avons donc utilisé la même technique que C4.5 pour les discrétiser. À chaque nœud de l'arbre, avant de sélectionner la nouvelle variable, l'algorithme binarise toutes les variables candidates X_i en effectuant une adaptation de seuil de manière à minimiser l'entropie conditionnelle $H(Y/X_i)$ ou, de manière à maximiser l'indice $q(Y/X_i)$.

À titre indicatif, les arbres obtenus sont donnés en annexe (figures D.1, D.2, D.3 et D.4). Étant donné la taille de ceux-ci, il n'est pas possible de les rendre lisibles, mais on peut toutefois remarquer l'importante différence entre la taille des arbres obtenus par chaque approche. L'arbre obtenu par C4.5 est beaucoup plus volumineux que celui obtenu par l'approche ascendante désagrégative par nœud. On peut également remarquer que l'élagage n'a pas beaucoup d'influence sur ce dernier. Ceci dit, si l'on regarde de plus près la structure des arbres, on s'aperçoit que dans tous les cas, les premières variables testées sont les résidus rs_1 et rs_2. Ceci paraît tout à fait logique car ces variables « contiennent » l'information sur le modèle analytique à partir duquel ils ont été calculés.

En ce qui concerne la taille des arbres, on obtient avec *C4.5* 235 branches pour l'arbre non élagué et 217 pour l'arbre élagué. Avec l'approche ascendante désagrégative par nœud, on obtient 167 branches pour l'arbre non élagué et 157 pour l'arbre élagué.

En ce qui concerne le pourcentage d'erreur sur les populations test, on obtient en moyenne 6.31 % d'erreurs avec l'arbre élagué de *C4.5* et 4.32 % d'erreurs pour l'arbre élagué de l'approche ascendante désagrégative par nœud. Les résultats sont à peu près similaires pour les deux méthodes lorsque la vitesse de référence est la même que celle utilisée pour la population d'apprentissage avec un taux d'erreur qui n'excède pas 5.7 %. Dans les autres cas, l'approche ascendante désagrégative par nœud obtient sensiblement de meilleurs résultats, les moins bons résultats étant obtenus lorsque la vitesse de référence est supérieure à celle de l'apprentissage, en présence des deux défauts et avec un faible défaut sur le capteur de vitesse. Ces résultats sont détaillés en annexe (tableau D.1).

Il est à noter que ces taux d'erreur sont à relativiser, car les erreurs surviennent de façon ponctuelle et comme le montre le tableau D.2 donné en annexe, un simple filtrage

permettrait de diminuer considérablement ces taux déjà faibles.

5.3 Conclusions et perspectives

Dans ce chapitre, les méthodes d'induction par arbres de décision ont été testées sur des tableaux issus de la littérature et une application dans le domaine du diagnostic a pu être traitée.

Nous avons mis en évidence l'intérêt de l'utilisation des méthodes basées sur une approche ascendante désagrégative en traitant les tableaux issus de la littérature.

Le traitement des données issues du moteur asynchrone montre que les méthodes proposées sont validées sur des exemples plus industriels rencontrés par la communauté automaticienne. Sans aucune autre connaissance *a priori* du système que le tableau initial des données, l'algorithme apprend seul (sans aucune aide de l'utilisateur, d'où le terme « automatique ») un modèle du système sans avoir à manipuler les équations (d'où le terme « apprentissage ») afin de trouver **le** bon résidu qui nous fera discriminer plusieurs états de fonctionnement. Les résultats obtenus sont très convaincants car le taux d'erreur n'excède pas 5.7 % dans cette application. Nous rappelons également que dans cette approche, les résidus (donc la connaissance d'un certain modèle mathématique) ont été utilisés, et que nous n'excluons pas les approches basées sur un modèle mathématique et les approches sans modèle mathématique. Nous pensons que ces méthodes sont complémentaires.

Les justifications théoriques semblent impossibles à définir, et il nous faut alors comparer les méthodes entre elles sur des exemples communs. C'est ce qui est généralement fait dans la littérature. Il est bien évident qu'il ne faut pas se borner à une seule application industrielle pour en déduire des généralités, et que les perspectives prochaines seront de traiter d'autres exemples dans le cadre du diagnostic afin d'établir des règles solides de choix d'algorithmes...

Conclusions et perspectives générales

Dans un premier temps, la démarche générale de l'étude d'un système sans modèle mathématique de comportement a été présentée.

Dans le premier chapitre, certaines définitions ont été introduites. Les données d'apprentissage sur lesquelles se base l'analyse du système, ont ainsi été présentées. À ce titre, les notions de système physique et de système cognitif ont été définies, en spécifiant en particulier le vocabulaire utilisé d'une part, par la communauté de l'apprentissage, et par la communauté automaticienne d'autre part. Nous nous sommes intéressés ensuite à la phase préliminaire d'étude d'un système, dans laquelle l'expert est confronté à un choix de variables bien souvent guidé par son « savoir-faire », avant d'introduire la phase d'observation nécessaire à cette étude. La notion de finesse nous a permis de quantifier intuitivement l'information que peut apporter une variable sur le système. Certaines propriétés sur la relation de finesse entre variables ont alors été démontrées. Les notions de variables à expliquer et de variables potentiellement explicatives ont été ensuite définies en se focalisant tout spécialement sur les systèmes dynamiques. La notion d'incohérence des données a alors été introduite, et sur cette base, un tableau de contingence a pu être construit.

Dans le deuxième chapitre, les méthodes issues de l'apprentissage ont été présentées dans le cadre d'une analyse structurale d'un système physique plus ou moins complexe. Les problèmes de l'analyse structurale ont ainsi été exposés, à savoir les problèmes de visualisation, structuration, explication et prédiction, et des méthodes issues de l'apprentissage automatique ont été présentées afin de remédier à ces problèmes. Dans ce cadre, les notions de « modèle », de « connaissance », et de « prédiction » ont été définies ainsi que l'apprentissage supervisé particulièrement intéressant pour ce travail. Les différentes démarches proposées sont l'approche basée sur les données (ou *approche ascendante*), et l'approche basée plutôt sur la construction d'un modèle à base de règles préétabli (ou *approche descendante*). Dans ce contexte, les principaux algorithmes utilisés dans la littérature ont été présentés, et une méthodologie générale de validation de modèle permettant de mesurer la « qualité » du modèle a été présentée afin de ne garder que le meilleur des nombreux modèles obtenus.

Dans un deuxième temps, le cadre de travail ayant été présenté, une méthodologie d'étude d'un système physique par construction d'arbres de décision a été proposée.

Tout d'abord, la présentation des outils de la théorie de l'information appliquée à l'analyse structurale des systèmes nous a permis d'établir un critère d'étude utilisable dans les algorithmes développés. En effet, l'entropie conditionnelle s'est avérée particulièrement

bien adaptée au problème d'explication.

Puis, nous nous sommes proposé de construire un modèle de comportement par l'intermédiaire d'arbres de décision. *C4.5* qui est certainement l'algorithme de construction d'arbres de décision le plus usité a été décrit, afin de définir les principes généraux de construction des arbres de décision. Cependant, il peut arriver que le pouvoir explicatif de chaque variable d'un vecteur S soit faible, alors que celui du vecteur S considéré globalement est important. Dans ce cas, un algorithme utilisant un critère prenant en compte chaque variable séparément, (ou ne prenant pas en compte tout de suite l'ensemble des variables de S), ne permettra pas de mettre en évidence certaines relations existant entre ces variables, et donc ne les choisira pas comme variables discriminantes. C'est pourquoi plusieurs points de vue de construction d'arbres de décision ont été considérés et combinés afin d'obtenir des algorithmes permettant de pallier cet inconvénient. On voit immédiatement l'intérêt d'utiliser un algorithme *agrégatif* ou à plus forte raison *désagrégatif* (on calcule d'abord le critère sur l'ensemble S, puis au fur et à mesure de la progression dans l'algorithme on désagrège cet ensemble). Cette démarche permet d'écarter des variables faussement informatives, et de conserver les variables réellement pertinentes. Les méthodes de construction par niveau ont été présentées afin de mettre en évidence l'intérêt de l'approche ascendante désagrégative. Quelques approches par nœud ont été développées reprenant les caractéristiques de cette dernière, de manière à obtenir une version affinée et plus souple que l'approche par niveau.

Enfin, le dernier chapitre a permis de valider les méthodes développées sur des cas concrets. Nous avons mis en évidence l'intérêt de l'utilisation des méthodes basées sur une approche ascendante désagrégative en traitant des tableaux issus de la littérature. Le traitement des données issues du moteur asynchrone montre que les méthodes proposées sont validées sur des exemples plus industriels. Les résultats obtenus avec la méthode ascendante désagrégative par nœud sur les données issues du moteur asynchrone sont très convaincants. Elle obtient de manière générale de meilleurs résultats que l'algorithme *C4.5* en termes de taille de modèle ainsi qu'en termes de précision de celui-ci sur des données test...

Perspectives...

Les différentes approches proposées construisent des arbres de décision. Elles ont naturellement été comparées à *ID3* et *C4.5* qui sont des références dans ce domaine. Il serait intéressant de mesurer leurs performances à celles des algorithmes introduits dans le chapitre 2, à savoir, *AQ*, les algorithmes à base d'instances, les algorithmes utilisant le formalisme des treillis, ou même les réseaux de neurones.

D'autre part, les algorithmes de base ont été développés. Il reste à y ajouter les différentes fonctions périphériques ou supplémentaires telles que le traitement des données manquantes, l'élagage de l'arbre, etc...

Nous avons introduit les approches par nœud et par modalité qui permettent de sélec-

tionner un test sur une modalité d'une variable, plutôt que sur la totalité des modalités de la variable choisie. Nous pouvons de surcroît nous intéresser à un sous-ensemble de modalités plutôt qu'à une modalité particulière, de la même façon que l'on effectue une adaptation de seuil pour les données numériques. Cette démarche pourrait atténuer le problème de biais (entraîné par ce seuillage des variables continues) vers les tests numériques.

Enfin, en ce qui concerne les données dynamiques, le choix des techniques de dérivation ou de prise en compte des variables retardées, peut être amélioré de manière un peu plus structurée. Les dynamiques des variables n'étant pas forcément identiques, on peut d'ailleurs se demander jusqu'à quel ordre il est nécessaire d'observer le système pour obtenir de bonnes performances. Dans ce sens, on peut envisager dans le cadre d'une démarche de visualisation, d'utiliser l'approche ascendante désagrégative par niveau avec l'ensemble des variables :

$$\Sigma = \bigcup_{i=0}^{o_{max}} \Sigma^i$$

(voir chapitre 1) (avec o_{max} fixé *a priori*), afin de déterminer les ordres d'observation maximum de chaque variable. Une autre étude a été entreprise [CCPR97] qui permettra de distinguer l'évolution d'une variable à court terme de son évolution à long terme.

Annexe A

Tableau extrait de [Mar87]

(proposé par G. SAPORTA)

27 races de chiens ont été caractérisées par 7 paramètres: *Taille*, *Poids*, *Vélocité*, *Intelligence*, *Affection*, *Agressivité* et *Fonction*. Ces caractéristiques ont été rassemblées dans le tableau A.1

où :

- -, 0, + signifient respectivement « *faible* », « *moyen* » et « *fort* » ;
- Utilitaire, Chasse, Compagnie signifient respectivement « *chien utilitaire* », « *chien de chasse* » et « *chien de compagnie* ».

À titre d'exemple, le *chihuahua* est caractérisé par une taille, un poids, une vélocité, une intelligence et une agressivité faibles, par une affection forte et par une fonction de chien de compagnie.

On cherchera par exemple à caractériser les concepts « *chien utilitaire* », « *chien de chasse* » et « *chien de compagnie* ».

En prenant la *fonction* comme variable à expliquer, et l'ensemble des autres variables comme variable explicative, on peut établir le tableau de contingence A.2.

		Taille -, 0, +	Poids -, 0, +	Vélocité -, 0, +	Intelligence -, 0, +	Affection -, +	Agressivité -, +	Fonction Ut, Ch, Co
1	beauceron	+	0	+	0	+	+	Utilitaire
2	basset	-	-	-	-	-	+	Chasse
3	berger allemand	+	0	+	+	+	+	Utilitaire
4	boxer	0	0	0	0	+	+	Compagnie
5	bull-dog	-	-	-	0	+	-	Compagnie
6	bull-mastiff	+	+	-	+	-	+	Utilitaire
7	caniche	-	-	0	+	+	-	Compagnie
8	chihuahua	-	-	-	-	+	-	Compagnie
9	cocker	0	-	-	0	+	+	Compagnie
10	colley	+	0	+	0	+	-	Compagnie
11	dalmatien	0	0	0	0	+	-	Compagnie
12	doberman	+	0	+	+	-	+	Utilitaire
13	dogue allemand	+	+	+	-	-	+	Utilitaire
14	épagneul breton	0	0	0	+	+	-	Chasse
15	épagneul français	+	0	0	0	-	-	Chasse
16	fox-hound	+	0	+	-	-	+	Chasse
17	fox-terrier	-	-	0	0	+	+	Compagnie
18	grand bleu	+	0	0	-	-	+	Chasse
19	labrador	0	0	0	0	+	-	Chasse
20	lévrier	+	0	+	-	-	-	Chasse
21	mastiff	+	+	-	-	-	+	Utilitaire
22	pékinois	-	-	-	-	+	-	Compagnie
23	pointer	+	0	+	+	-	-	Chasse
24	saint-bernard	+	+	-	0	-	+	Utilitaire
25	setter	+	0	+	0	-	-	Chasse
26	teckel	-	-	-	0	+	-	Compagnie
27	terre-neuve	+	+	-	0	-	-	Utilitaire

TAB. A.1: *Un exemple de tableau initial des données*

X						$Y = fonction$		
Taille	Poids	Vélocité	Intelligence	Affection	Agressivité	Utilitaire	Chasse	Compagnie
+	0	+	0	+	+	37	0	0
-	-	-	-	-	+	0	37	0
+	0	+	+	+	+	37	0	0
0	0	0	0	+	+	0	0	37
-	-	-	0	+	-	0	0	74
+	+	-	+	-	+	37	0	0
-	-	0	+	+	-	0	0	37
-	-	-	-	+	-	0	0	74
0	-	-	0	+	+	0	0	37
+	0	+	0	+	-	0	0	37
0	0	0	0	+	-	0	37	37
+	0	+	+	-	+	37	0	0
+	+	+	-	-	+	37	0	0
0	0	0	+	+	-	0	37	0
+	0	0	0	-	-	0	37	0
+	0	+	-	-	+	0	37	0
-	-	0	0	+	+	0	0	37
+	0	0	-	-	+	0	37	0
+	0	+	-	-	-	0	37	0
+	+	-	-	-	+	37	0	0
+	0	+	+	-	-	0	37	0
+	+	-	0	-	+	37	0	0
+	0	+	0	-	-	0	37	0
+	+	-	0	-	-	37	0	0

×1/1000

TAB. A.2: *Un exemple de tableau de contingence*

Annexe B

Tableau des entropies concernant l'exemple du chapitre 4

Dans cet exemple, $H(Y) = 1$ et $H(Y/X) = 0$, avec $X = (a_1, a_2, a_3, a_4, a_5)$. Il existe donc une relation entre X et Y.

S	$H(Y/S)$	$q(Y/S)$
a_1	1	0 %
a_2	1	0 %
a_3	1	0 %
a_4	0.689	31.13 %
a_5	0.951	4.88 %
$a_1\ a_2$	0	100 %
$a_1\ a_3$	1	0 %
$a_1\ a_4$	0.5	50 %
$a_1\ a_5$	0.844	15.56 %
$a_2\ a_3$	1	0 %
$a_2\ a_4$	0.5	50 %
$a_2\ a_5$	0.844	15.56 %
$a_3\ a_4$	0.689	31.13 %
$a_3\ a_5$	0.844	15.56 %
$a_4\ a_5$	0.5	50 %

S	$H(Y/S)$	$q(Y/S)$
$a_1\ a_2\ a_3$	0	100 %
$a_1\ a_2\ a_4$	0	100 %
$a_1\ a_2\ a_5$	0	100 %
$a_1\ a_3\ a_4$	0.5	50 %
$a_1\ a_3\ a_5$	0.75	25 %
$a_1\ a_4\ a_5$	0.344	65.56 %
$a_2\ a_3\ a_4$	0.5	50 %
$a_2\ a_3\ a_5$	0.75	25 %
$a_2\ a_4\ a_5$	0.344	65.56 %
$a_3\ a_4\ a_5$	0.344	65.56 %
$a_1\ a_2\ a_3\ a_4$	0	100 %
$a_1\ a_2\ a_3\ a_5$	0	100 %
$a_1\ a_2\ a_4\ a_5$	0	100 %
$a_1\ a_3\ a_4\ a_5$	0.25	75 %
$a_2\ a_3\ a_4\ a_5$	0.25	75 %
$a_1\ a_2\ a_3\ a_4\ a_5$	0	100 %

TAB. B.1: *Entropies et indices de modélisabilité*

Annexe C

Quelques exemples d'arbres obtenus

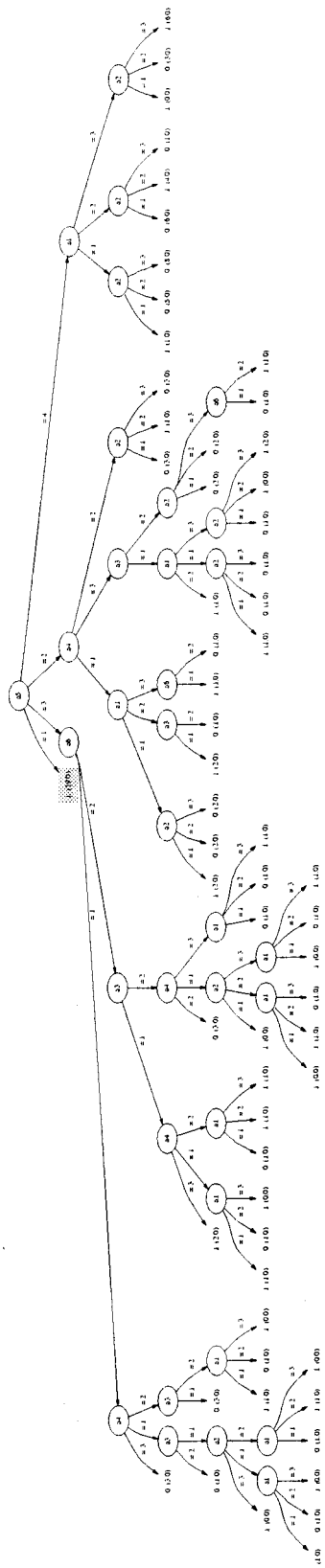

FIG. C.1: *Arbre obtenu avec l'approche descendante agrégative par nœud sur le tableau « monks-1 »*

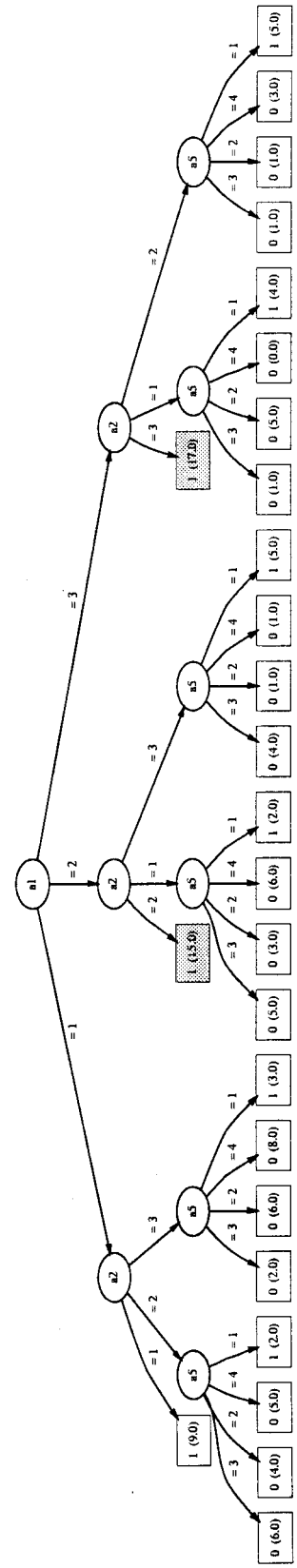

FIG. C.2: *Arbre obtenu avec l'approche ascendante désagrégative par nœud sur le tableau « monks-1 »*

Annexe D

Tableaux des résultats obtenus sur les données issues du moteur asynchrone

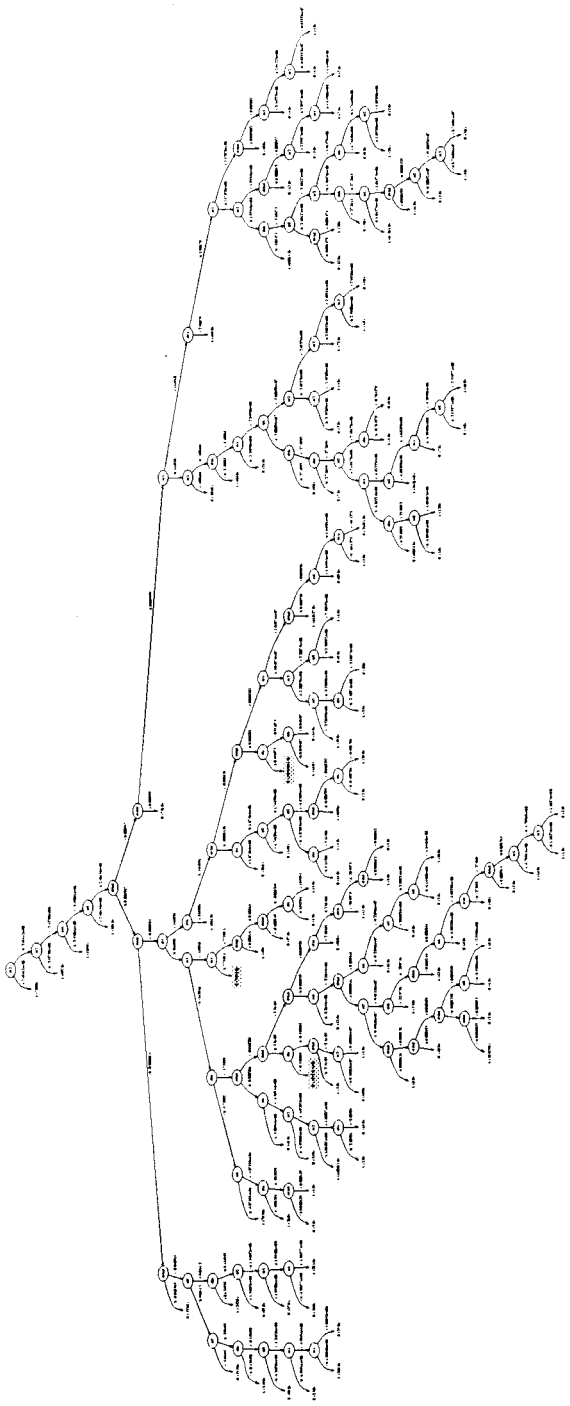

FIG. D.1: *Arbre obtenu avec C4.5 sur les données issues du moteur asynchrone*

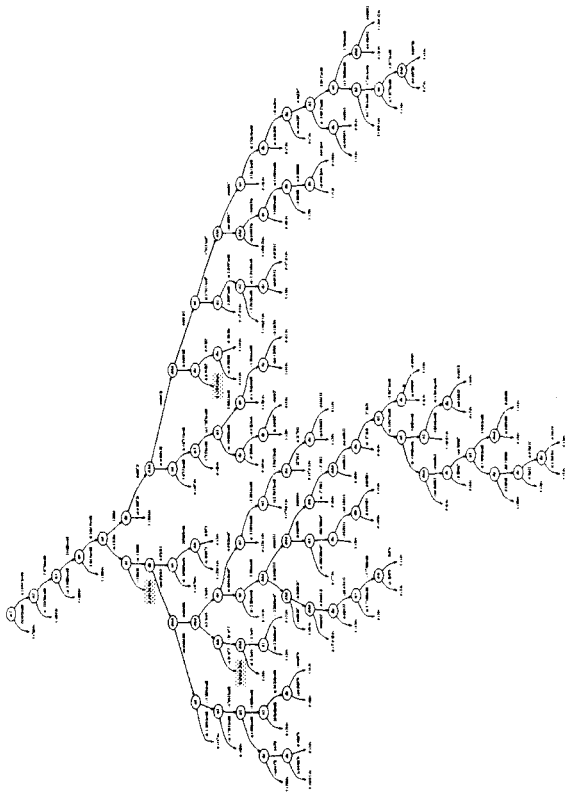

FIG. D.2: *Arbre obtenu avec l'approche ascendante désagrégative par nœud sur les données issues du moteur asynchrone*

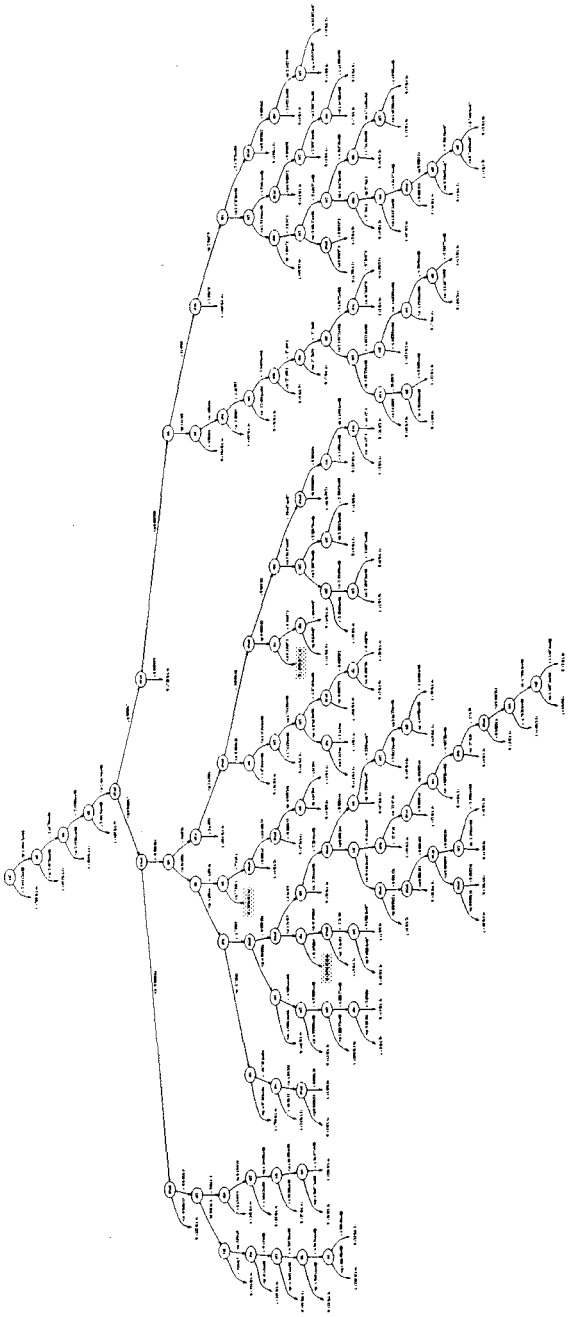

FIG. D.3: *Arbre obtenu avec C4.5 sur les données issues du moteur asynchrone après élagage*

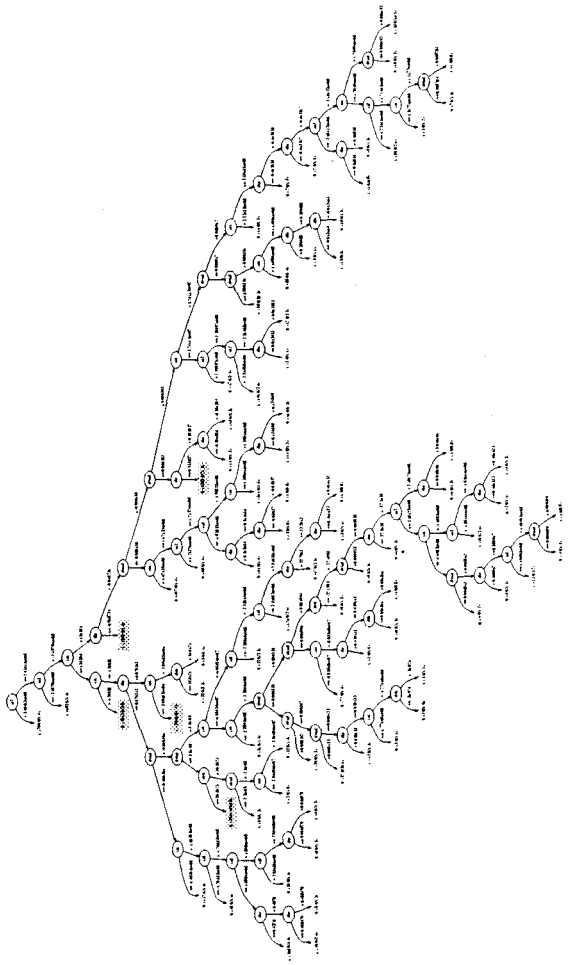

FIG. D.4: *Arbre obtenu avec l'approche ascendante désagrégative par nœud sur les données issues du moteur asynchrone après élagage*

			C4.5				Approche ascendante désagrégative par nœud			
			avant élagage		après élagage		avant élagage		après élagage	
			taille	taux d'erreurs	taille	taux d'erreurs	taille	taux d'erreurs	taille	taux d'erreurs
Population d'apprentissage 20 000 points (20 s.)	$Vref = 1415$ tr/mn $Vit \rightarrow Vit + 10$ $R_r \rightarrow 2 \times R_r$ $R_r \rightarrow 2 \times R_r$ et $Vit \rightarrow Vit + 10$		235	140 (0,7%)	217	145 (0,7%)	167	166 (0,8%)	157	168 (0,8%)
Populations test 10 000 points (10 s.) (défaut sur 5 s.)	$V_{ref} = 1415$ tr/mn	$R_r \rightarrow 1.5 \times R_r$	235	0 (0,0%)	217	0 (0,0%)	167	2 (0,0%)	157	2 (0,0%)
		$R_r \rightarrow 2 \times R_r$	235	1 (0,0%)	217	1 (0,0%)	167	4 (0,0%)	157	4 (0,0%)
		$Vit \rightarrow Vit + 3$	235	74 (0,7%)	217	74 (0,7%)	167	119 (1,2%)	157	119 (1,2%)
		$Vit \rightarrow Vit + 5$	235	82 (0,8%)	217	80 (0,8%)	167	101 (1,0%)	157	101 (1,0%)
		$Vit \rightarrow Vit + 20$	235	153 (1,5%)	217	152 (1,5%)	167	10 (0,1%)	157	10 (0,1%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 3$	235	561 (5,6%)	217	548 (5,5%)	167	555 (5,6%)	157	555 (5,6%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 5$	235	505 (5,1%)	217	495 (5,0%)	167	505 (5,1%)	157	505 (5,1%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	235	501 (5,0%)	217	501 (5,0%)	167	450 (4,5%)	157	450 (4,5%)
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 3$	235	471 (4,7%)	217	460 (4,6%)	167	568 (5,7%)	157	569 (5,7%)
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 5$	235	242 (2,4%)	217	228 (2,3%)	167	262 (2,6%)	157	260 (2,6%)
		$R_r \rightarrow 2 \times R_r$ (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	235	471 (4,7%)	217	471 (4,7%)	167	191 (1,9%)	157	143 (1,4%)
	$V_{ref} = 1500$ tr/mn	$R_r \rightarrow 1.5 \times R_r$	235	156 (1,6%)	217	156 (1,6%)	167	61 (0,6%)	157	61 (0,6%)
		$R_r \rightarrow 2 \times R_r$	235	1990 (19,9%)	217	1990 (19,9%)	167	1902 (19,0%)	157	987 (9,9%)
		$Vit \rightarrow Vit + 3$	235	283 (2,8%)	217	281 (2,8%)	167	221 (2,2%)	157	221 (2,2%)
		$Vit \rightarrow Vit + 5$	235	377 (3,8%)	217	377 (3,8%)	167	200 (2,0%)	157	200 (2,0%)
		$Vit \rightarrow Vit + 20$	235	428 (4,3%)	217	426 (4,3%)	167	68 (0,7%)	157	68 (0,7%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 3$	235	765 (7,7 %)	217	763 (7,6%)	167	729 (7,3%)	157	729 (7,3%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 5$	235	649 (6,5%)	217	661 (6,6%)	167	536 (5,4%)	157	536 (5,4%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	235	604 (6,0%)	217	610 (6,1%)	167	459 (4,6%)	157	459 (4,6%)
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 3$	235	2111 (21,1%)	217	2111 (21,1%)	167	2025 (20,3%)	157	1591 (15,9%)
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 5$	235	1686 (16,9%)	217	1731 (17,3%)	167	1613 (16,1%)	157	1142 (11,4%)
		$R_r \rightarrow 2 \times R_r$ (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	235	1598 (16,0%)	217	1598 (16,0%)	167	1267 (12,7%)	157	747 (7,5%)
	$V_{ref} = 900$ tr/mn	$R_r \rightarrow 1.5 \times R_r$	235	0 (0,0%)	217	0 (0,0%)	167	9 (0,1%)	157	9 (0,1%)
		$R_r \rightarrow 2 \times R_r$	235	0 (0,0%)	217	0 (0,0%)	167	21 (0,2%)	157	21 (0,2%)
		$Vit \rightarrow Vit + 3$	235	720 (7,2%)	217	719 (7,2%)	167	535 (5,4%)	157	535 (5,4%)
		$Vit \rightarrow Vit + 5$	235	656 (6,6%)	217	655 (6,6%)	167	479 (4,8%)	157	479 (4,8%)
		$Vit \rightarrow Vit + 20$	235	1079 (10,8%)	217	1087 (10,9%)	167	403 (4,0%)	157	403 (4,0%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 3$	235	605 (6,1%)	217	574 (5,7%)	167	556 (5,6%)	157	556 (5,6%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 5$	235	525 (5,3%)	217	520 (5,2%)	167	527 (5,3%)	157	527 (5,3%)
		$R_r \rightarrow 1.5 \times R_r$ (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	235	1162 (11,6%)	217	1166 (11,7%)	167	524 (5,2%)	157	524 (5,2%)
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 3$	235	733 (7,3%)	217	690 (6,9%)	167	687 (6,9%)	157	687 (6,9%)
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) $Vit \rightarrow Vit + 5$	235	548 (5,5%)	217	536 (5,4%)	167	533 (5,3%)	157	533 (5,3%)
		$R_r \rightarrow 2 \times R_r$ (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	235	1143 (11,4%)	217	1147 (11,5%)	167	512 (5,1%)	157	512 (5,1%)

TAB. D.1: *Résultats obtenus sur le moteur*

			C4.5					App. asc. désagrégative par nœud				
			$\%\epsilon_1$	p_{max}	of_{opt}	$\%\epsilon_{opt}$	$\%\epsilon_{10}$	$\%\epsilon_1$	p_{max}	of_{opt}	$\%\epsilon_{opt}$	$\%\epsilon_{10}$
Population d'apprentissage 20 000 points (20 s.)	$Vref = 1415$ tr/mn $Vit \rightarrow Vit + 10$ $R_r \rightarrow 2 \times R_r$ $R_r \rightarrow 2 \times R_r$ et $Vit \rightarrow Vit + 10$		0,7%	27	20	0,7%	0,7%	0,8%	27	21	0,6%	0,8%
Populations test 10 000 points (10 s.) (défaut sur 5 s.)	$V_{ref} = 1415$ tr/mn	$R_r \rightarrow 1.5 \times R_r$	0,0%	1	1	0,0%	0,0%	0,0%	1	2	0,0%	0,0%
		$R_r \rightarrow 2 \times R_r$	0,0%	1	2	0,0%	0,0%	0,0%	1	2	0,0%	0,0%
		$Vit \rightarrow Vit + 3$	0,7%	3	4	0,1%	0,2%	1,2%	4	4	0,1%	0,2%
		$Vit \rightarrow Vit + 5$	0,8%	3	2	0,1%	0,2%	1,0%	4	2	0,1%	0,2%
		$Vit \rightarrow Vit + 20$	1,5%	112	6	1,2%	1,3%	0,1%	2	3	0,1%	0,2%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 3$	5,5%	155	9	5,1%	5,1%	5,6%	137	11	4,6%	4,9%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 5$	5,0%	183	2	4,9%	5,0%	5,1%	193	6	4,9%	5,0%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	5,0%	169	54	4,6%	5,1%	4,5%	38	12	4,1%	4,4%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 3$	4,6%	81	11	2,9%	3,0%	5,7%	81	19	4,7%	5,2%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 5$	2,3%	44	7	1,2%	1,6%	2,6%	44	9	1,4%	1,4%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	4,7%	169	41	4,0%	6,1%	1,4%	16	16	0,2%	0,5%
	$V_{ref} = 1500$ tr/mn	$R_r \rightarrow 1.5 \times R_r$	1,6%	10	11	0,0%	0,4%	0,6%	9	10	0,0%	0,0%
		$R_r \rightarrow 2 \times R_r$	19,9%	10	11	0,0%	0,4%	9,9%	9	10	0,0%	0,0%
		$Vit \rightarrow Vit + 3$	2,8%	10	11	0,3%	0,9%	2,2%	9	10	0,2%	0,2%
		$Vit \rightarrow Vit + 5$	3,8%	10	11	0,2%	1,1%	2,0%	9	10	0,2%	0,2%
		$Vit \rightarrow Vit + 20$	4,3%	112	11	1,3%	2,0%	0,7%	9	10	0,2%	0,2%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 3$	7,6%	25	15	4,4%	5,5%	7,3%	34	12	5,2%	5,9%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 5$	6,6%	23	24	2,4%	5,7%	5,4%	34	13	4,0%	4,3%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	6,1%	150	15	3,4%	5,0%	4,6%	34	10	3,6%	3,6%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 3$	21,1%	57	11	9,9%	10,6%	15,9%	57	19	8,4%	12,5%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 5$	17,3%	44	13	4,9%	5,8%	11,4%	35	23	4,0%	6,0%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	16,0%	169	55	5,3%	9,3%	7,5%	15	14	1,2%	1,9%
	$V_{ref} = 900$ tr/mn	$R_r \rightarrow 1.5 \times R_r$	0,0%	1	1	0,0%	0,0%	0,1%	1	2	0,0%	0,0%
		$R_r \rightarrow 2 \times R_r$	0,0%	1	1	0,0%	0,0%	0,2%	1	2	0,0%	0,0%
		$Vit \rightarrow Vit + 3$	7,2%	9	10	0,2%	0,2%	5,4%	9	10	0,2%	0,2%
		$Vit \rightarrow Vit + 5$	6,6%	9	10	0,2%	0,2%	4,8%	11	12	0,2%	0,2%
		$Vit \rightarrow Vit + 20$	10,9%	113	114	3,4%	10,0%	4,0%	9	10	0,2%	0,2%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 3$	5,7%	111	11	5,1%	5,3%	5,6%	111	11	5,1%	5,3%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 5$	5,2%	104	24	4,9%	5,0%	5,3%	251	7	5,1%	5,1%
		$R_r \rightarrow 1.5 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	11,7%	185	59	11,0%	11,5%	5,2%	104	9	5,1%	5,1%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 3$	6,9%	171	68	5,7%	6,6%	6,9%	171	68	5,7%	6,6%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 5$	5,4%	138	18	4,9%	5,0%	5,3%	164	7	5,1%	5,1%
		$R_r \rightarrow 2 \times R_r$ (déf. R_r sur 6 s.) (déf. Vit sur 2 s.) $Vit \rightarrow Vit + 20$	11,5%	185	54	11,0%	11,5%	5,1%	157	4	5,0%	5,1%

TAB. D.2: *Résultats obtenus sur le moteur après filtrage*

où :

- $\%\epsilon_1$ est le taux d'erreurs initial ;
- p_{max} est la persistance maximale des erreurs ;
- of_{opt} est l'ordre de filtrage optimal ;
- $\%\epsilon_{opt}$ est le taux d'erreurs après filtrage optimal ;
- $\%\epsilon_{10}$ est le taux d'erreurs après filtrage d'ordre 10.

Bibliographie

[Agg74] N. L. AGGARWAL. « *Mesures d'information et questionnaires arborescents* ». Thèse de doctorat, Besançon, 1974.

[Agg76] N. L. AGGARWAL. « Mesures d'information : caractéristiques et propriétés ». Dans *Ecole de l'INRIA : Théorie de l'information*, 1976.

[AKA91] David W. AHA, Dennis KIBLER, et K. ALBERT. « Instance-Based Learning Algorithms ». *Machine Learning Journal*, 6:37–66, 1991.

[Ash56] W. R. ASHBY. *Introduction to Cybernetics.* Champan & Hall, London, 1956.

[Ash65a] R. ASH. *Information Theory.* John Wiley & Sons, 1965.

[Ash65b] W. R. ASHBY. « Measuring the Internal Informational Exchange in a System ». Dans *Cybernetica*, volume 8, Namur, Belgique, 1965.

[Ash69] W. R. ASHBY. « Two Tables of Identities Governing Information Flows within Large Systems ». *Communication of the American Society of Cybernetics*, 1, 1969.

[Bar87] M. BARBOUCHA. « Modélisation structurale des systèmes complexes - Extraction et validation des règles d'un système expert ». Thèse d'état, Univ. des Sciences et des Techniques de Lille Flandres Artois, Lille I, juin 1987.

[Ben80] J.-P. BENZECRI. *Analyse des données.* Dunod, 1980.

[BFOS84] L. BREIMAN, J. H. FRIEDMAN, R. A. OLSHEN, et C. J. STONE. *Classification and Regression Trees.* Wadsworth International Group, Belmont,CA, 1984.

[BG88] J. P. BARTHÉLÉMY et A. GUENOCHE. *Les arbres et les représentations des proximités.* Méthodes + Programmes. Masson, 1988.

[Bir67] G. BIRKHOFF. *Lattice Theory.* Providence. Amer. Math. Soc., 3e édition, 1967.

[BJL+90] J. BRUNET, D. JAUME, M. LABARRÈRE, A. RAULT, et M. VERGÉ. *Détection et diagnostic de pannes. Approches par modélisation.* Hermès, Paris, 1990.

[BM70] M. BARBUT et B. MONJARDET. *Ordre et classification - algèbre et combinatoire*, volume 1 & 2. Hachette Université, 1970.

[BMDD+90] B. BOUCHON-MEUNIER, S. DESPRES, D. DUBOIS, O. GASCUEL, A. GUENOCHE, et H. PRADE. « Aspects de l'interface entre symbolique et numérique ». Dans *3es journées nationales PRC-GDR intelligence artificielle*, pages 90–112, CNIT, Paris la Défense, 5-7 mars 1990. Hermes.

[BN93] Michèle BASSEVILLE et Igor V. NIKIFOROV. *Detection of Abrupt Changes - Theory and Application.* Information and System Sciences Series. Prentice Hall, 1993.

[BS89] J.-M. BOUROCHE et G. SAPORTA. *L'analyse des données.* Collection Que sais-je? Presses Universitaires de France, 4e édition, 1989.

[BSA87] M. BARBOUCHA, M. STAROSWIECKI, et P. AYGALINC. « Building Rules from Contingency Tables ». Dans *5es journées internationales « analyse des données et informatique »*, Versailles, septembre-octobre 1987. INRIA. tome 1.

[BU95] Carla E. BRODLEY et Paul E. UTGOFF. « Multivariate Decision Trees ». *Machine Learning Journal*, 19:45–77, 1995.

[CA70] R. C. CONANT et W. R. ASHBY. « Every Good Regulator of a System must be a Model of that System ». *International Journal of System Science*, 1(2), 1970.

[CCPR97] Daniel CALVELO, Marie-Christine CHAMBRIN, Denis POMORSKI, et P. RAVAUX. « Arbres de décision et analyse dynamique de données médicales: introduction de la notion d'échelle ». Dans *Colloque de Recherche Doctorale AGIS'97 « Automatique - Génie informatique - Image - Signal »*, pages 159–166, Angers, 9-11 décembre 1997.

[CDG+89] G. CELEUX, E. DIDAY, G. GOVAERT, Y. LECHEVALLIER, et H. RALAMBONDRAINY. *Classification automatique des données.* Dunod, 1989.

[CH67] T. M. COVER et P. E. HART. « Nearest Neighbour Pattern Classification ». *IEEE Trans. on Information Theory*, IT-13(1):21–27, janvier 1967.

[Cib87] Ph. CIBOIS. L'analyse factorielle. Dans *Que sais-je?* Presses Universitaires de France, 2e édition, 1987.

[CN89] Peter CLARK et Tim NIBLETT. « The CN2 Induction Algorithm ». *Machine Learning Journal*, 3(4):261–283, 1989.

[Con69] R. C. CONANT. « The Information Transfer Required in Regulatory Process ». *IEEE Trans. on Systems, Sciences and Cybernetics*, SSC-5(4), 1969.

[Con72] R. C. CONANT. « Detecting Subsystems of a Complex System ». *IEEE Trans. of Systems, Man and Cybernetics*, 2(4):550–553, septembre 1972.

[Con76] R. C. CONANT. « Laws of Information which Govern Systems ». *IEEE Trans. on Systems Man and Cybernetics*, SMC-6(4), 1976.

[Coo64] C. H. COOMBS. *A Theory of Data.* Wiley, New-York, 1964.

[CT95] John G. CLEARY et Leonard E. TRIGG. « K*: an Instance-based Learner Using an Entropic Distance Measure ». Dans *Proceedings of International Conference on Machine Learning.* Morgan Kaufmann, 1995.

[CV97] G. COMTET-VARGA. « *Surveillance des systèmes non linéaires – Application aux machines asynchrones* ». Thèse de doctorat, Université des Sciences et Technologies de Lille, Lille I, 2 décembre 1997.

[CVCCS99] G. COMTET-VARGA, C. CHRISTOPHE, V. COCQUEMPOT, et M. STAROSWIECKI. « F.D.I. for the Induction Motor Using Elimination Theory ». Dans *Procedings of the European Control Conference ECC'99*, 1999. soumis.

[DDG96] François DENIS, Cyrille D'HALLUIN, et Rémi GILLERON. « PAC Learning with Simple Examples ». Dans *Proceedings of the 13th Symposium on Theoretical Aspects of Computer Science STACS'96*, volume 1046 de *Lecture Notes in Computer Science*, pages 231–242, 1996.

[Del71] P. DELATTRE. « *Système, structure, fonction, évolution* », page 11. Maloine-Doin, Paris, 1971.

[DF96] Peter DEAN et A. FAMILI. « Comparative Performance of Rule Quality Measures in an Induction System ». *The Applied Intelligence Journal*, 1996.

[DG97] François DENIS et Rémi GILLERON. « PAC Learning under Helpful Distributions ». Dans *Proceedings of the 8th International Workshop on Algorithmic Learning Theory ALT'97*, volume 1316 de *Lecture Notes in Artificial Intelligence*, pages 132–145, 1997.

[DGS97] François DENIS, Rémi GILLERON, et Jean SIMON. « Apprentissage PAC avec enseignant ». Dans *JFA'97*, 1997.

[Did72] E. DIDAY. « *Nouvelles méthodes et nouveaux concepts en classification automatique et reconnaissance des formes* ». Thèse de doctorat, Univ. Paris 6, 4 décembre 1972.

[Dub90] B. DUBUISSON. *Diagnostic et reconnaissance des formes.* Hermès, Paris, 1990.

[Duf79] J. DUFOUR. « Méthodes et méthodologies d'analyse de systèmes complexes. Application aux procédés industriels et aux systèmes macroéconomiques ». Thèse d'état, Univ. Claude Bernard, Lyon, mai 1979.

[Dus80] A. DUSSAUCHOY. « Generalized Information Theory on some Ordered Sets and Structure of Systems ». Dans *14th Annual Conference on Information Sciences in Systems*, Princeton University, 1980.

[FI92] U. M. FAYYAD et K. B. IRANI. « On the Handling of Continuous-Valued Attributes in Decision Tree Generation ». *Machine Learning*, 8:87–102, 1992.

[For65] E. W. FORGY. « Cluster Analysis of Multivariate Data ». *Biometrics*, 21(3), septembre 1965.

[Fos95] Ian FOSTER. *Designing and Building Parallel Programs - Concepts and Tools for Software Engineering.* Addison-Wesley Publishing Company, 1995.

[Fou85] T. FOUCART. *Analyse Factorielle - programmation sur micro-ordinateurs.* Méthodes + Programmes. Masson, 1985.

[Gat72] Geoffrey E. GATES. « The Reduced Nearest Neighbour Rule ». *IEEE Trans. on Information Theory*, pages 431–433, 1972.

[Gui77] Silviu GUIASU. *Information Theory with Applications.* Mc Graw-Hill International Book Company, 1977.

[Har28] R. V. L. HARTLEY. « Transmission of Information ». *Bell System Technical Journal*, 7:535, 1928.

[Har68] Peter E. HART. « The Condensed Nearest Neighbour Rule ». *IEEE Trans. on Information Theory*, pages 516–517, mai 1968.

[Har84] A. HART. « Experience in the Use of an Inductive System in Knowledge Engineering ». Dans M. BRAMER, éditeur, *Research and Developments in Expert Systems*, Cambridge, 1984. Cambridge University Press.

[HMS66] E. HUNT, J. MARIN, et P. STONE. *Experiments in Induction.* Academic Press, New-York, 1966.

[Hot33] H. HOTELLING. « Analysis of a Complex of Statistical Variables into Principal Components ». *Journal of Educational Psychology*, 24:417–441, 498–520, 1933.

[HS94] M. HOLSHEIMER et A. P. J. M. SIEBES. « Data Mining : the Search for Knowledge in Databases ». Report cs-r9406, Computer Science / Departement of Algorithms and Architechture (Centrum voor Wiskunde en Informatica), Amsterdam, The Netherlands, 1994.

[Ise84] R. ISERMANN. « Process Fault Detection Based on Modeling and Estimation Methods ». *Automatica*, 30:387–404, 1984.

[JL78] M. JAMBU et M. O. LEBEAUX. *Classification automatique pour l'analyse des données*, volume 1 : Méthodes et algorithmes. Dunod Décision, 1978.

[Jum78] Guy JUMARIE. « Théorie relativiste de l'information et télécommunication. Perspectives. ». *Annales des télécommunications*, 33(1-2):13–27, janvier 1978.

[Jum79a] Guy JUMARIE. « Théorie relativiste de l'information II. Information de Shannon, entropie de Renyi, ensembles flous relativistes ». *Annales des télécommunications*, 34(9-10):491–507, septembre - octobre 1979.

[Jum79b] Guy JUMARIE. « Théorie relativiste de l'information III. Sur la signification et l'utilisation effective de l'entropie de Renyi dans les problèmes de codage ». *Annales des télécommunications*, 34(11-12):521–530, novembre - décembre 1979.

[Jum80] Guy JUMARIE. « Théorie relativiste de l'information IV. Sur l'introduction de facteurs subjectifs dans les processus de communication ». *Annales des télécommunications*, 35(7-8):281–296, juillet - août 1980.

[KB69] J. KAMPÉ DE FÉRIET et P. BENVENUTI. « Sur une classe d'informations ». *Comptes rendus Acad. Sciences*, 269:97–101, 1969. Série A.

[KB72] J. KAMPÉ DE FÉRIET et P. BENVENUTI. « Opération de composition régulière et ensembles de valeurs d'une information ». *Comptes rendus Acad. Sciences*, 274:655–659, 1972. Série A.

[KB78] A. KAUFMANN et G. BOULAYE. *Théorie des treillis en vue des applications.* Masson, Paris, 1978.

[KF67] J. KAMPÉ DE FÉRIET et B. FORTE. « Information et probabilité ». *Comptes rendus Acad. Sciences*, 265:110–114, 142–146, 350–353, 1967. Série A.

[KFB69] J. KAMPÉ DE FÉRIET, B. FORTE, et P. BENVENUTI. « Forme générale de l'opération de composition continue d'une information ». *Comptes rendus Acad. Sciences*, 269:529–534, 1969. Série A.

[Kli69] G. J. KLIR. *An Approach to General Systems Theory.* Van Nostrand Reinnhold, New-York, 1969.

[Kli75] G. J. KLIR. « On the Representation of Activity Arrays ». *International Journal of General Systems*, 2(3):59–71, 1975.

[Kli76] G. J. KLIR. « Identification of Generative Structures in Empirical Data ». *International Journal of General Systems*, 3(2):89–104, 1976.

[Kli77] G. J. KLIR. « On the Problem of Computer-aided Structure Identification : some Experimental Observations and Resulting Guidelines ». *International Journal of Man-Machine Studies*, 9:593–628, 1977.

[Lan73] Claude LANGRAND. « *Information généralisée. Estimation de sélection.* ». Thèse de doctorat, Université des Sciences et Techniques de Lille, 1973.

[Ler70] I. C. LERMAN. *Les bases de la classification automatique.* Programmation. Gauthier-Villars, Paris, 1970.

[LV97] M. LI et P. VITÁNYI. *An Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, 2e édition, 1997.

[Mac67] J. B. MAC QUEEN. « Some Methods for Classification and Analysis of Multivariate Observations ». Dans *5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1. Berkeley University of California Press, 1967.

[Mar87] F. MARCOTORCHINO. « Une approche unifiée des problèmes de sériation par blocs ». Dans *5es journées internationales « Analyse des données et informatique »*. INRIA, 29 sept - 2 oct 1987.

[Mep94] Engelbert MEPHU N'GUIFO. « Galois Lattice: A Framework for Concept Learning. Design, Evaluation and Refinement. ». Dans *the sixth International Conf. on Tools with Artificial Intelligence*, pages 461–467, New Orleans, Louisiana, LA, 6-9 novembre 1994. IEEE Press.

[Mil63] G. A. MILLER. « What is Information Measurement ». *American Psychologist*, 8(2):50–51, 1963.

[Min89] J. MINGERS. « An Empirical Comparison of Selection Measures for Decision-Tree Induction ». *Machine Learning Journal*, 4:319–342, 1989.

[MM96] C. MERZ et P. MURPHY. « UCI Repository of Machine Learning Databases ». http://www.ics.uci.edu/~mlearn/MLRepository.html, 1996. Irvine, CA: University of California, Department of Information and Computer Science.

[MMHL86a] Ryszard S. MICHALSKI, Igor MOZETIC, Jiarong HONG, et Nada LAVRAC. « The AQ15 Inductive Learning System : an Overview and Experiments ». Technical report uiucdcs-r-86-1260, University of Illinois, juillet 1986.

[MMHL86b] Ryszard S. MICHALSKI, Igor MOZETIC, Jiarong HONG, et Nada LAVRAC. « The Multi-purpose Incremental Learning System AQ15 and its Testing Application to three Medical Domains ». Dans *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 1041–1045, Philadelphia, 1986.

[MS98] Paul MUNTEANU et Jean-François SERIGNAT. « IDF : induction d'arbres de décision par l'approximation des fréquences ». *Revue Électronique sur l'Apprentissage par les Données*, 2(1):22–38, février 1998.

[Mur96] Kolluru Venkata Sreerama MURTHY. « *On Growing Better Decision Trees from Data* ». PhD thesis, The John Hopkins University, Baltimore, Maryland, 1996.

[Nik95] Igor V. NIKIFOROV. « A Generalized Change Detection Problem ». *IEEE Transactions on Information Theory*, 41(1), janvier 1995.

[NM98] Patrick NJIWOUA et Engelbert MEPHU N'GUIFO. « IGLUE: Un système d'apprentissage à base d'instances utilisant le formalisme des treillis ». Dans *Actes du 11^e Congrès RFIA '98*, Clermont-Ferrand, 20-22 janvier 1998. AFCET-AFIA.

[Nyq24] H. NYQUIST. « Certain Factors Affecting Telegraph Speed ». *Bell System Technical Journal*, 3:324, 1924.

[Osw86] G. OSWALD. *Théorie de l'information ou analyse diacritique des systèmes.* Masson, Paris, 1986.

[Par60] E. PARZEN. *Modern Probability Theory and its Application.* John Wiley and Sons, New-York, 1960.

[Pic72] C. F. PICARD. *Graphes et questionnaires*, volume 1 & 2 de *Programmation.* Gauthier-Villars, 1972.

[Pom91] D. POMORSKI. « *Apprentissage automatique symbolique / numérique. Construction et évaluation d'un ensemble de règles à partir des données* ». Thèse de doctorat, Univ. des Sciences et Technologies de Lille, Lille I, décembre 1991.

[Pos87] J.-G. POSTAIRE. *De l'image à la décision.* Dunod Informatique, Paris, 1987.

[PP97a] P.-B. PERCHE et D. POMORSKI. « Decision Tree Induction Methods Using an Entropy Criterion - I- Global Approaches ». Dans D. W.PEARSON, éditeur, *Proceedings of the Second International ICSC Symposium on Soft Computing SOCO'97*, pages 286–293, Nîmes, 17-19 septembre 1997. ICSC Academic Press.

[PP97b] P.-B. PERCHE et D. POMORSKI. « Decision Tree Induction Methods Using an Entropy Criterion - II- Local Approaches ». Dans D. W.PEARSON, éditeur, *Proceedings of the Second International ICSC Symposium on Soft Computing SOCO'97*, pages 294–299, Nîmes, 17-19 septembre 1997. ICSC Academic Press.

[PPS95] P.-B. PERCHE, D. POMORSKI, et M. STAROSWIECKI. « Méthodes d'induction par arbre de décision utilisant un critère entropique ». Dans Claude HUMBERT, éditeur, *2ème conférence internationale sur l'automatisation industrielle*, Nancy (France), juin 1995. CRAN, AIAI.

[PS91] Gregory PIATETSKY-SHAPIRO. « Discovery, Analysis, and Presentation of Strong Rules ». Dans Gregory PIATETSKY-SHAPIRO et William J. FRAWLEY, éditeurs, *Knowledge Discovery in Databases*, pages 229–248, Menlo Park, California, 1991. AAAI Press.

[PSS92] D. POMORSKI, M. SBAÏ, et M. STAROSWIECKI. « Automatic Learning from Incoherent Data ». Dans *Systems Science XI*, Wroclaw (Pologne), septembre 1992.

[PSS96] Denis POMORSKI, Mouloud SBAÏ, et Marcel STAROSWIECKI. « A Qualitative Approach to the Change Detection Problem with Different Types of Sampling ». Dans *IEEE International Conference on Systems, Man, and Cybernetics (SMC'96)*, Beijing, China, 14-17 octobre 1996.

[Qui83] J. R. QUINLAN. Learning Efficient Classification Procedures and their Application to Chess and Games. Dans R. S. MICHALSKI, J. G. CARBONELL, et T. M. MITCHELL, éditeurs, *Machine Learning: an Artificial Intelligence Approach*, pages 463–482. Tioga Publishing Company, 1983.

[Qui86] J. R. QUINLAN. « Induction of Decision Trees ». *Machine Learning Journal*, 1:81–106, 1986.

[Qui93] J. R. QUINLAN. *C4.5: Programs for Machine Learning.* Morgan Kauffman, 1993.

[Qui95] J. R. QUINLAN. « The MDL and Categorical Theories (Continued) ». Dans *Proceedings of the 12th International Conference on Machine Learning*, pages 464–470, 1995.

[Qui96] J. R. QUINLAN. « Improved Use of Continuous Attributes in C4.5 ». *Journal of Artificial Intelligence Research*, 4:77–90, mars 1996.

[Ral87] H. RALAMBONDRAINY. « GENREG un générateur de règles combinant techniques d'apprentissage et techniques d'analyse des données ». Dans *Journées « symbolique numérique » pour l'apprentissage de connaissances à partir des*

données, pages 40–44, Univ. Paris-Dauphine, décembre 1987. Diday and Kodratoff.

[Ral88] H. RALAMBONDRAINY. « The Algorithm GENREG for Generating Rules from Symbolic or Numerical Data ». rapport 910, INRIA-Rocquencourt, octobre 1988.

[Ren86] L. RENDELL. « A General Framework for Induction and a Study of Selective Induction ». *Machine Learning Journal*, 1:177–226, 1986.

[Ric75] M. RICHETIN. « Analyse structurale des systèmes complexes en vue d'une commande hiérarchisée ». Thèse d'état, Univ. Paul Sabatier, Toulouse, 1975.

[Ris78] J. RISSANEN. « Modeling by Shortest Data Description ». *Automatica*, 14:465–471, 1978.

[Ris83] J. RISSANEN. « A Universal Prior for Integers and Estimation by Minimum Description Length ». *Annals of Statistics*, 11:416–431, 1983.

[Rou85] M. ROUX. *Algorithmes de classification.* Méthode + Programmes. Masson, 1985.

[SB91] M. STAROSWIECKI et M. BARBOUCHA. « Apprentissage automatique par filtrage de tableaux de contingence ». Dans DIDAY et KODRATOFF, éditeurs, *Induction symbolique et numérique à partir des données*, Paris, 1991. Cepadues.

[Sba83] M. SBAÏ. « *Analyse structurale des systèmes complexes : méthodes d'explication et de partition* ». Thèse de doctorat, Univ. des Sciences et des Techniques de Lille Flandres Artois, Lille I, 29 septembre 1983.

[Sba93] Mouloud SBAÏ. « Modélisation structurale, apprentissage automatique et détection de rupture dans les systèmes complexes ». Thèse d'état, faculté des sciences, Université Mohamed 1er d'Oujda, Maroc, 1993.

[SBS92] M. SBAÏ, M. BARBOUCHA, et M. STAROSWIECKI. « Génération des règles à partir des données incohérentes ». Dans *ICEA'92 First International Conference on Electronic and Automatic Control*, Tizi Ouzou, 3-5 mai 1992.

[Seb96] M. SEBBAN. « *Modèles théoriques en reconnaissance de formes et architecture hybride pour machine perceptive* ». Thèse de doctorat, Université Lyon I, 1996.

[Sha48] C. E. SHANNON. « A Mathematical Theory of Communication ». *Bell System Technical Journal*, 1948.

[Sha49a] C. E. SHANNON. « Communication in the Presence of Noise ». *IRE*, 37:10, 1949.

[Sha49b] C. E. SHANNON. *The Mathematical Theory of Communications.* The University of Illinois Press, Urbana, Il, 1949.

[Sil68] David C. SILLS. William of Ockham. Dans *International Encyclopedia of the Social Sciences*, pages 269–270. Macmillan Company & The Free Press, New-York, 1968.

[Sle74] D. SLEPIAN. *Key Papers in the Development of Information Theory.* IEEE Press, New-York, 1974.

[Sta84] M. STAROSWIECKI. « Analyse structurale des systèmes complexes ». *Rairo Automatique*, 18(2), 1984.

[Sza62] G. SZASZ. *Introduction to Lattice Theory.* Academic Press, New-York and London, 1962.

[Tor82] V. M. TORO CORDOBA. « *Contribution à l'analyse structurale des systèmes complexes à l'aide de l'entropie et ses généralisations* ». Thèse de doctorat, Univ. des Sciences et Techniques de Lille Flandres Artois, Lille I, mars 1982.

[Utg95] Paul E. UTGOFF. « Decision Tree Induction Based on Efficient Tree Restructuring ». Rapport Technique, Department of Computer Science, University of Massachusetts, 17 mars 1995.

[Val84] L. G. VALIANT. « A Theory of the Learnable ». *Comm. ACM*, pages 1134–1142, 1984.

[Wal77] B. WALLISER. *Systèmes et modèles.* Editions du Seuil, 1977.

[Wyn81] A. D. WYNER. « Fundamental Limits in Information Theory ». Dans *Proceedings of IEEE*, volume 69, février 1981.

[ZS98] D. A. ZIGHED et M. SEBBAN. « Sélection et validation statistique de variables et de prototypes ». *Revue Électronique sur l'Apprentissage par les Données*, 2(1):1–21, février 1998.