République Algérienne Démocratique et Populaire.
Ministère de l'Enseignement Supérieur
Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF
FACULTE DE GENIE ELECTRIQUE
DEPARTEMENT D'ELECTRONIQUE

THESE

en vue de l'obtention du diplôme de **DOCTORAT ES SCIENCE**

SPECIALITE : Electronique

OPTION : Traitement du signal

Présentée par

SENOUSSI Hafida

Sujet du Mémoire de Thèse

Sélection de Données pour l'Apprentissage des Réseaux de Neurones, Arbres de Décision et les k-Plus Proches Voisins : Application en Diagnostic de Pannes

Soutenue le 27 avril 2015 devant le jury :

Mr Z. AHMED FOUATIH	Professeur	USTO	Président
Mr A.H.BOUDINAR	Maître de conférences A	USTO	Rapporteur
Mr N. ZERHOUNI	Professeur	FEMTO-ST (Besançon)	Co- Rapporteur
Mme F. HENDEL	Professeur	USTO	Examinateur
Mr M. A. CHIKH	Professeur	Université de Tlemcen	Examinateur
Mr M. ZERIKAT	Professeur	ENP Oran	Examinateur

A ma Mère, qui est partenaire à part entière de ma réussite, de par son amour, son optimisme sans faille, ces longues années de sacrifices pour m'aider à avancer dans la vie et aller au delà de mes capacités. Merci Maman pour les valeurs nobles, l'éducation et ton soutien permanent et indéfectible.

A la mémoire de mon Père,

A mes chers frères :
Mustapha, Taj Eddine, Fouad, Jamel, Azzedine,
Sid Ahmed et Mohammed,

A ma chère Sœur Nassima

A la mémoire d'un grand « Homme » qui m'a tant appris.

« Abandonné à sa solitude, l'homme se sent assailli d'un sentiment de vide cosmique. C'est sa façon de remplir ce vide qui déterminera le type de sa culture et de sa civilisation, c'est-à-dire tous les caractères internes et externes de sa vocation historique. Il y a essentiellement deux manières de le faire: regarder à ses pieds, vers la terre, ou lever les yeux vers le ciel. L'un peuplera sa solitude de choses. Son regard dominateur veut posséder. L'autre peuplera sa solitude d'idées. Son regard interrogateur est en quête de vérité. Ainsi naissent deux types de culture : une culture d'empire, aux racines techniques et une culture de civilisation aux racines éthique et métaphysiques. » Malek Bennabi. *Problème des idées dans la société musulmane*.

Remerciements

J'adresse toute ma reconnaissance à Monsieurs Boudinar Ahmed Hamida et Monsieur Denai Mouloud qui ont accepté de diriger mes travaux de thèse ainsi que Monsieur Zerhouni Noureddine, qui m'a accueilli et dirigé ce travail au sein de l'équipe Systèmes de Maintenance Intelligents (SyMI) du département AS2M à Besançon (France). Je les remercie pour leur soutien, et la confiance qu'ils ont placé en moi.

J'exprime une incommensurable gratitude à Madame Brigitte Chebel-Morello, qui a co-dirigé ce travail au sein de l'équipe Systèmes de Maintenance Intelligents. Sans elle, cette véritable amitié et engagement moral, l'achèvement de ce travail n'aurait jamais vu le jour. Je la remercie sincèrement pour sa présence, disponibilité, patience et sourire, pour avoir toujours su m'orienter, me redynamiser tout au long de ces années.

Je remercie Monsieur Ahmed Fouatih Zoubir du département d'Electronique de l'USTOMB d'avoir accepté de juger ce travail et de présider le jury.

Je remercie également Madame Hendel Fatiha du département d'Electronique de l'USTOMB, Monsieur Zerikat Mokhtar de l'école supérieur ENP d'Oran ainsi que Monsieur Chikh Mohammed Amine de l'Université de Tlemcen d'avoir accepté d'examiner ce mémoire.

Merci tout particulièrement à ma précieuse amie et collègue Melle Zakuiya Khiyat qui m'a accueilli au Laboratoire de Génie Électrique d'Oran (LGEO), son soutien sans faille, ces conseils, son aide morale si précieuse tout au long de mon parcours a été pour moi cette étincelle qui a permis l'aboutissement et la concrétisation de ce travail.

Je remercie sincèrement mes amis si précieux : Hadjira, Mohammed, Rachida, Faten, Tama, Imène, Mohammed et Mustapha pour leur encouragement, disponibilité et soutien permanent.

J'aimerai aussi remercier Monsieur Flazi Samir pour m'avoir accueilli au sein de son laboratoire (LGEO), sans oublier bien sûr les enseignants du département d'électronique de l'USTOMB qui ont contribué de par leur enseignement, conseil, attention, et encouragement à ma formation.

Je tiens aussi à remercier toutes les personnes de l'équipe COSMI à Besançon ainsi que le laboratoire (LGEO) à Oran qui ont contribué à ce que ce travail se fasse dans une ambiance très chaleureuse. Je pense en particulier à Otilia, Karim, Mez, Ali, Eric, Mohammed, Aicha, Hbib et les autres.

Avant propos

Cette thèse s'inscrit dans le cadre d'un co-encadrement entre le département d'Electronique de l'Université des Sciences et de la Technologie d'Oran (Algérie) et le département AS2M de Besançon (France). Cette thèse a par ailleurs bénéficiée d'une bourse de recherche de 18 mois dans le cadre du programme de bourse nationale (PBN) au sein de l'équipe Systèmes de Maintenance Intelligents (SyMI) à Besançon (France).

L'idée de ce projet de thèse est venue du constat qu'en diagnostic de défaillances, les outils de prétraitement de données tel que la sélection de données sont très peu utilisés, notamment les algorithmes de sélection de variables qui traitent la corrélation forte entre variables.

La recherche que nous avons menée se veut être une contribution devant permettre de mettre en relief l'efficacité des algorithmes de sélection de variables contextuels à améliorer les performances des systèmes de diagnostic et de détection.

Table des matières

Liste des figures

Liste des tableaux

Notations et abréviations

BS : Stratégie de recherche arrière (Backward Selection).

C4.5 : Algorithme d'arbre de décision.

Chi2 : Algorithme de discrétisation de variable.

CFS : Algorithme de sélection de variable (*Fast Feature Selection*).

ConsistancySubsetEval(GA) : Algorithme utilisant le critère de consistance avec une stratégie de recherche par Algorithme Génétique.

DC : Dicriminating capacity.

DCG : Dicriminating capacity gain.

ECD : Extraction de Connaissance à Partir de Données.

FCBF : Algorithme de sélection de variables (Fast Correlation based Feature Selection algorithm)

FS : Stratégie de recherche (Foward Selection)

INTERACT : Algorithme de sélection de variables (Interacting based Feature Selection algorithm).

IBk ou KNN : algorithme des K plus proches voisins.

GainRatio : Algorithme de sélection de variables utilisant le gain d'information comme critère de sélection de variable avec une stratégie de recherche de type FS.

mRMR : Algorithme de sélection de variables (*minimum Redondancy Maximun Relevance*).

MDLM : Algorithme de discrétisation de variable.

MLP : Perceptron Multicouches (*Multi Layer Perceptron : MLP*).

Relief : Algorithme de sélection de variables.

STRASS : *STrong Relevant Algorithm for Subset Selection* (*Algorithme de sélection de variables à pertinence forte*).

WrapperGA(C4.5) : Méthode enveloppe de sélection de variables utilisant la précision de C4.5 et une stratégie de recherche par Algorithme Génétique.

WrapperGA(IB_k) : Méthode enveloppe de sélection de variables utilisant la précision des KNN (IB_k) et une stratégie de recherche par Algorithme Génétique.

Introduction générale

La maintenance industrielle prend une place de plus en plus grandissante dans l'entreprise. La fiabilité des systèmes de production, l'évolution vers une meilleure qualité, l'automatisation de plus en plus grande des processus techniques amènent les services de maintenance à une position centrale dans l'entreprise de production. Le développement actuel des Technologies de l'Information et de la Communication, facilite la mise en place d'une maintenance intelligente. En effet grâce à l'évolution technologique, l'intelligence se trouve de plus en plus distribuée et se rapproche des niveaux bas (opérationnels) des systèmes automatisés. En effet, L'opération de diagnostic menée par l'expert est souvent très complexe et demande des connaissances ainsi qu'un raisonnement, généralement difficiles à formaliser. Mais avec les outils de *fouille de données* (en anglais : data mining), on peut formaliser ces informations contenues dans l'historique de fonctionnement, qui représentent alors la base d'apprentissage supervisé pour l'algorithme d'induction. Si l'on considère, les différents paramètres mesurés sur le procédé comme variables d'entrée de l'algorithme de fouille de données. On cherchera par conséquence à associer un mode de fonctionnement à ces variables d'entrée. Les variables de sortie sont alors des variables catégorielles où chaque catégorie représente un mode de fonctionnement. Sachant que la détection peut être vue comme une classification à deux classes : fonctionnement normal et fonctionnement anormal du procédé. Par conséquent, le diagnostic peut être vu comme une extension de la détection pour le cas où le nombre de classes est supérieur à 2. Ainsi, dans le cas d'un fonctionnement anormal détecté, on peut s'intéresser à savoir quelle type de faute (ou défaut) s'est produite.

Ce travail s'inscrit dans le cadre très général du processus *d'Extraction de Connaissance à partir des Données* (ECD), plus particulièrement les méthodes de sélection de variables avec une application au problème particulier du diagnostic industriel. L'ECD a pour origine d'une part la taille volumineuse des données et d'autre part les problèmes rencontrés par les outils de fouille de données. Nous avons focalisé notre recherche sur les techniques issues

de l'apprentissage supervisé et les technique de sélections de variables comme outil de prétraitement de données afin d'en extraire l'information utile et pertinente. Dans ce contexte le problème qui émerge se pose en termes de réduction de la dimension des données qui concerne le nombre de variables descriptives caractérisant chacune des observations.

La réduction de dimension peut être réalisée par quatre familles de méthodes : l'Extraction, la Construction, la Discrétisation et la Sélection de variables.

- L'extraction a pour objet de rechercher un petit ensemble de nouvelles variables décrivant les données de façon équivalente à l'ensemble initial de variables.
- La construction de variables génère, par processus d'agrégation un ensemble de variables de taille inférieur à l'ensemble initial.
- La discrétisation des variables réduit la dimension des données par un traitement sur les valeurs prises par les variables.
- La sélection de variables ou d'objets (Blum et al., 1997), (Guyon et al., 2003) connue aussi dans la communauté de fouille de données sous le nom de *filtrage de données*, fait l'objet de notre étude. Tout particulièrement la sélection d'un sous ensemble de variables pertinentes.

Les méthodes de filtrage de variables ont émergé suite à l'incapacité des algorithmes d'apprentissage inductif, en particulier les arbres de décision, à détecter les interactions entre variables. Outre leur capacité à traiter les variables corrélées qui déterminent les classes, elles permettent de détecter les variables redondantes ou inutiles. Ce sont donc des méthodes efficaces de prétraitement qui permettent non seulement d'accélérer le processus d'apprentissage, mais également d'en améliorer les performances en termes de précision et de généralisation.

L'étape de filtrage de données dans le contexte de la sélection de variables, a intéressé beaucoup d'auteurs (Bobrowski, 1988), (Almuallim, et al., 1991), (Battiti, 1994), (Blum, et al., 1997), (Yu, et al., 2004), (Dash, et al., 2000), (Hall, 2000), (Yu, et al., 2004), (Liu, et al., 2005), (Zhao, et al., 2007), (Karegowda, et al., 2010), (Chandrashekar, et al., 2014).

Cette étape a pour objectif de privilégier la qualité à la quantité en sélectionnant un sous ensemble de variables pertinentes pour expliquer la variable classe ou variable but. Toutefois aucun de ces algorithmes ne détectent les variables partiellement redondantes. C'est justement l'objet de notre étude. Notre travail s'articule autour de quatre chapitres.

Le premier chapitre de ce travail inscrit les démarches d'un processus d'Extraction de Connaissances à partir de Données (ECD) aussi connu sous le terme anglais *KDD* (*Knowledge Discovry in Databases*). Le processus présente plusieurs étapes aussi importantes les unes que les autres, à savoir la collecte de données, l'étape de prétraitement, la fouille de données et le post-traitement. Deux nous concernant directement : le prétraitement et la fouille de données. Maints travaux soulignent la diminution des performances des algorithmes de fouille de données quand beaucoup de variables descriptives ou d'exemples sont traitées (Almuallim, et al., 1991), (Kira, et al., 1992), (Langley, et al., 1997). En effet, trop de variables descriptives diminuent la précision d'un classificateur et certaines même parasitent le traitement. Il faut noter toutefois que toutes les variables n'ont pas la même importance dans l'élaboration du classificateur. Nous présenterons, les notions de bases en fouille de données, les définitions des différentes variables pouvant être rencontrées dans des ensembles d'apprentissage traitant des données déterministes, non bruitées et modélisées par des variables nominales ou catégoriques. Ces notions étant posées, nous décrirons les algorithmes de discrétisation de variables, en raison de leur importance pour le traitement d'ensembles de données pouvant comporter des variables continues ainsi que les méthodes de classification supervisée que nous avons utilisé afin de valider nos travaux sur le filtrage de données. Dans la littérature, parmi les algorithmes d'induction ayant la préférence des chercheurs du domaine, nous avons étudié les arbres de décision, les k plus proches voisins et les réseaux de neurones multicouches.

Le deuxième chapitre est dédié au diagnostic des défaillances, dans ce chapitre nous allons présenter les techniques les plus courantes en diagnostic d'équipements industriels ainsi que les différentes définitions dans la littérature associée à ce domaine, mettant en emphase l'intérêt de notre méthodologie. Nous allons plus particulièrement nous intéresser

aux méthodes issues de l'ECD et considérer le diagnostic dans le cadre de l'apprentissage supervisé.

Nous effectuerons, dans le chapitre 3, un état de l'art des algorithmes contextuels de sélection de variables (*Feature selection*), avant de présenter par la suite, notre algorithme de sélection de variable noté STRASS, un algorithme contextuel de sélection de variables qui permet de prendre en compte d'une manière plus fine que les autres méthodes, le type de variables. Nous présenterons aussi les critères contextuels de sélection de variables proposés, que l'on resituera au préalable par rapport aux différents critères utilisés dans le cadre de la sélection de variables. En effet, la plus part des travaux en statistiques font l'hypothèse dans bien des cas erronée de l'indépendance entre les variables décrivant l'ensemble d'apprentissage. Nous relaxons cette hypothèse forte, et nous recherchons algorithmiquement à l'aide de 2 critères discriminants les interactions entre variables descriptives. Ces deux critères ont été construits à partir du *pouvoir discriminant* (Vignes, et al., 1992), (Michaut, 1999) qui est un critère contextuel mais qui travaille sur des paires de concepts, ce qui rend ce dernier peu adapté en sélection de données et notamment pour les bases à grandes dimensions. Nous allons donc commencer par présenter les critères contextuels sous leurs formes par paires afin de pouvoir par la suite les transformer sous une forme contingentielle. En effet, cette dernière est plus adaptée en fouille de données en raison de la combinatoire de ses calculs. Les nouveaux critères sont ensuite implantés dans un algorithme de filtrage, avec une recherche bidirectionnelle. L'algorithme que nous avons noté STRASS (*STrong Relevant Algorithm for Subset Selection*) (Senoussi, et al., 2008), mettra en évidence la dépendance partielle entre variables et permettra ainsi d'affiner la sélection d'un sous ensemble minimum de variables pertinentes ainsi que de générer une catégorisation de variables très intéressante.

Le dernier chapitre est consacré à la mise en œuvre des techniques étudiées. Nous allons :

- Dans un premier temps, comparer l'algorithme de sélection de variables STRASS par rapport aux algorithmes les plus utilisés que l'on a identifié comme

les meilleurs en ce qui concerne la prise en compte de différents types de variables. Afin de mettre en évidence l'efficacité de notre algorithme à traiter des données partiellement corrélées, nous l'avons testé sur des données artificielles et réelles connues pour leurs variables fortement corrélées, nous avons aussi jugé bon de voir l'impact de notre méthodologie sur les performances en classification de différents algorithmes d'induction tel que les arbres de décision, les k plus proches voisins et les réseaux de neurones multicouches.

- Dans un deuxième temps, nous allons appliquer notre méthodologie en diagnostic. Pour cela, nous allons d'abord exposer l'exemple d'un procédé chimique : le Tennessee Eastman Process (TEP) qui a constitué notre principale base de travail pour la partie applicative. Notre intérêt s'est porté sur le processus TEP, car ce dernier est reconnu pour ces données qui présentent de grandes interactions (corrélations) (Jockenhövel, et al., 2003), (Chiang, et al., 2004), (Verron, et al., 2008), (Nashalji, et al., 2009). Nous avons donc jugé bon de traiter ces données par notre algorithme. A cela succède alors la phase de sélection de données pertinentes qui sont ensuite utilisées comme données d'entrées pour les algorithmes d'apprentissage (Senoussi, et al., 2011(a)), (Senoussi, et al., 2012). Les résultats de classification obtenus sont ensuite comparés par rapport à d'autres travaux ayant traité le processus TEP en diagnostic de défauts. Nous discuterons les avantages et inconvénients de chaque méthode en termes de performance prédictive. Dans la dernière partie du chapitre, nous allons proposer cette fois d'utiliser la catégorisation de variables que permet d'obtenir l'algorithme STRASS lors de son déroulement afin de concevoir un système de détection de défaut plus fiable (Senoussi, et al., 2011(b)).

Chapitre 1

Cadre de la sélection de variables & ECD

Sommaire

1.1 Introduction

La création de vastes bases de données dans presque tous les domaines de l'activité humaine a généré une demande pressante d'outils permettant de transformer les données en connaissance orientée but. Afin de satisfaire ces besoins, les chercheurs exploitent de nouvelles méthodes d'extraction automatique de connaissances. De nombreux travaux se sont intéressés depuis bien longtemps à rechercher la structure sous-jacente des données notamment à partir de méthodes comme les statistiques, la reconnaissance de formes ou l'intelligence artificielle. Tous ces axes de recherche peuvent être fédérés sous le nom d'Extraction de Connaissance à partir de Données (ECD).

Les algorithmes de filtrage tiennent une place importante dans l'ECD. Ces derniers permettent non seulement, d'appliquer des algorithmes de fouille de données mais aussi dans certains cas d'améliorer les résultats induits.

Nous présentons dans ce chapitre le processus complet d'ECD. Les techniques et outils que nous utilisons sont : La discrétisation des données comme outil de prétraitement de données, la sélection de variables et l'apprentissage supervisé. L'accent sera mis sur les notions de bases pour la sélection de variables ainsi que les techniques de classification que nous avons utilisées dans le cadre de nos travaux de recherche.

1.2 Extraction de connaissances à partir de données

L'Extraction de connaissances à partir de données (ECD) ou Knowledge Discovery in Databases (KDD) et un processus d'identification de structure inconnues, quelles que soient leur formes, en vue d'en extraire de la connaissance (Fayyad, 1996). C'est un processus qui a émergé de l'analyse des données, de l'apprentissage automatique (intelligence artificielle) et des bases de données ; il est intégré dans le schéma organisationnel de l'entreprise. Les données proviennent d'entrepôts construits exprès pour leur exploitation. L'ECD présente plusieurs étapes aussi importantes les unes que les autres, à savoir :

1. La collecte (acquisition) de données et accès aux données, stockées sous une forme structurée (base de données, fichiers tabulaires) ou non-structurée (texte, image, etc.).
2. La préparation des données en vue du traitement (data warehousing), supprimer le bruit et réaliser une transformation par projection ou réduction de ces données suivant l'application étudiée.
3. Utilisation de techniques de fouille de données (data mining) : issues de la statistique ou de l'apprentissage automatique : La fouille de données se départage en deux groupes distincts. Le premier est le partitionnement (segmentation, clustering) issue de l'apprentissage non-supervisé, le second englobe les méthodes d'apprentissage supervisé (classification).
4. Evaluer et valider les connaissances extraites.
5. Déploiement des connaissances en vue d'une utilisation effective.

Le déroulement d'un projet n'est pas linéaire. On peut constater dans l'étape de validation (post-traitement), que les performances obtenues ne sont pas suffisantes ou que les utilisateurs du domaine jugent l'information inexploitable, il s'agira alors de choisir une autre méthode de fouille, ou de remettre en cause les codages, ou de chercher à enrichir les données.

L'étape de fouille de données (*data mining*) intègre à la fois le choix de la modélisation adéquate et de la méthode à utiliser ainsi que son application à la recherche de structure sous-jacentes des données et à la création de modèles explicatifs et/ou prédictifs. Le modèle obtenu doit être rapide à créer, rapide à utiliser, compréhensible pour l'utilisateur, donnant de bonnes performances qui ne se dégradent pas dans le temps (cas des modèles évolutifs). Il va de soi qu'aucun modèle n'aura toutes ces qualités. Il n'existe pas de meilleure méthode de fouille. Il faudra faire des compromis selon les besoins dégagés et les caractéristiques connues des outils (Morello, et al., 2001), (Arsselin & Kettaf, 2005), (Cornuéjols,et al., 2011).

1.3 Notions de base pour le filtrage de données

Trop de variables descriptives diminuent la précision d'un classifieur et certaines même parasitent le traitement. La prise en compte de leur nature est donc essentielle pour l'obtention de bons concepts. Dans cette section nous décrivons les types et la nature des variables que nous sommes susceptibles de rencontrer. Les objets de l'ensemble d'apprentissage sont décrits par un ensemble de variables (descripteurs, attributs) que l'on peut répertorier suivant leur type.

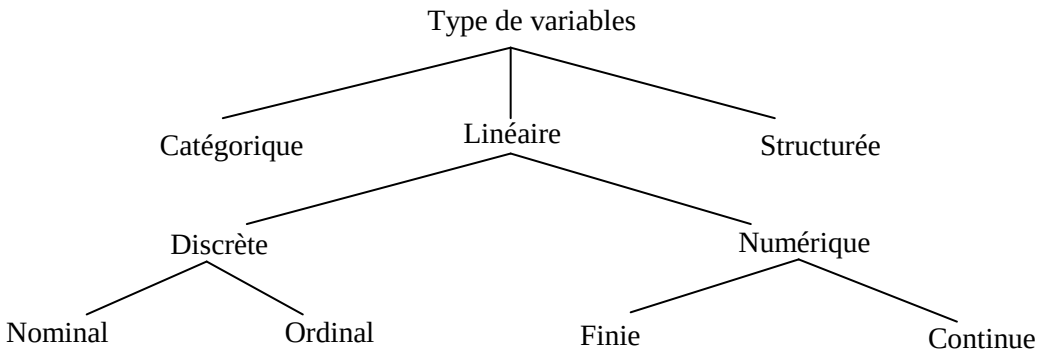

Figure 1-1 **:** *Hiérarchie des types de variables*

1.3.1 Type des variables

A. Les variables nominales ou catégoriques

Leur domaine de définition est composé de symboles ou de noms indépendants les uns des autres. Ainsi, aucune structure n'est supposée relier les valeurs du domaine.

B. Les variables linéaires

L'ensemble des valeurs possibles est composé de valeurs totalement ordonnées. Parmi ces ensembles, on distingue plusieurs catégories :

- Les valeurs nominales ordonnées : par exemple, le descripteur « *officier de l'armée de terre »* a pour domaine [lieutenant, capitaine, commandant, lieutenant colonel, général].
- Les valeurs ordinales : par exemples, le descripteur « *note obtenue* » a pour domaine [1,2,3,…,19,20].
- Les variables continues prennent leurs valeurs dans le domaine des réels ce qui implique que le nombre de valeurs possibles est infini.

C. Les variables structurées

Ces variables permettent une structuration des données. Une telle variable est une combinaison des variables précédentes. Son domaine de définition et composé d'une hiérarchie de généralisation ayant une structure d'arbre. Par exemple, dans l'ensemble des valeurs du descripteur « lieu » : « ALGERIE », serait un nœud parent des nœuds « Oran », « Alger », « Annaba », etc. La figure (1-1) présente une hiérarchie des types de variables précédemment décrites.

Les algorithmes utilisés dans le cadre de l'ECD présentent des aptitudes plus ou moins fortes à traiter tel ou tel type de variables. Par contre les algorithmes de sélection de variables que nous proposons dans le cadre de ce travail traitent des données symboliques,

il est donc nécessaire de transformer les données continues (données réelles) en données discrétisées (voir section 4).

1.3.2 Nature des variables

D'après leur nature toutes les variables n'ont pas la même importance dans l'élaboration du classifieur. Il est par conséquent très utile de pouvoir identifier dans l'ensemble d'apprentissage traité, les variables susceptibles de discriminer des concepts. Nous passerons en revue différentes définitions et propriétés décrivant les variables en prenant appui sur deux notions de base, la *corrélation* et la *pertinence*.

1.3.2.1 La corrélation

Liu (Liu, et al., 2005) formalise la corrélation comme suit: Mesures de dépendance sont aussi connus comme des mesures de corrélation ou de similarité. Ils mesurent la capacité à prédire la valeur d'une variable à partir de la valeur d'une autre variable.

Suivant la nature des données on trouve différentes définitions:

A. Corrélations entre variables de différentes natures

Définition 1

Nous regardons la façon dont une variable est associée à la classe. Une variable X est préférée à une autre variable Y si l'association de la variable X avec la classe C est plus élevée que l'association entre Y et C (Liu, et al., 2005).

B. Corrélation entre variables

Hall (Hall, 1998) étudie la corrélation entre variables, pour faire la différence entre la corrélation entre variables descriptives elles-mêmes et entre variables descriptives et la variable but. Il utilise la notion d'inter-corrélation et il reformule la pertinence d'un ensemble de variables comme suit :

Définition 2 (données non étiquetées)

L'association entre deux variables aléatoires mesure la similarité entre les deux.

Définition 3 (données étiquetées)

Un bon sous-ensemble de variables est celui qui contient des variables fortement corrélées avec (prédictives de) la classe, mais non corrélées les unes des autres.

Définition 4

Une variable est utile si elle est corrélée avec ou prédictive de la classe, sinon elle n'est pas pertinente (Kohavi, et al., 1997).

1.3.2.2 La Pertinence

Nous avons trouvé différentes définitions de la pertinence dans les travaux de Liu (Liu, & Motoda, 1998), Kohavi (Kohavi & John, 1997) et Blum et Langley (Blum, et al., 1997) que nous avons résumé comme suit:

Définition 5

Une variable est considérée pertinente si une fois retirée du sous ensemble de variables la mesure de l'ensemble des variables restantes sera détériorée, cette mesure peut être soit la précision, la consistance, l'information, la distance ou bien la dépendance (Liu, et al., 1998).

Définition 6

Une variable y_k est pertinente si elle est conditionnellement dépendante de la variable but (Kohavi, et al., 1997).

Définition 7 (Pertinence forte)

Une variable y_k est fortement pertinente par rapport à un échantillon d'exemples N s'il existe des exemples A et B appartenant à N qui ne diffèrent que par leur assignement à y_k et sont de classes différentes (Blum, et al., 1997).

Définition 8

Une variable y_k est fortement pertinente si les performances d'un classifieur de Bayes sont détériorés à cause de la suppression de y_k seulement (Kohavi, et al., 1997) .

Définition 9 (Pertinence faible)

Une variable y_k est faiblement pertinente par rapport à un échantillon d'exemples N (ou bien à un concept C et à une distribution D), si il est possible de supprimer un sous ensemble de variables tel que y_k devient fortement pertinente (Blum, et al., 1997).

Définition 10

Une variable y_k est faiblement pertinente si elle n'est pas fortement pertinente et si il existe un sous ensemble de variables, S', tel que les performances d'un classifieur de Bayes avec S' sont inférieures à ceux avec $S' \bigcup \{y_k\}$ (Kohavi, et al., 1997).

1.3.2.3 La Redondance

Deux variables sont dites redondantes quand elles jouent le même rôle, elles discriminent les objets d'une manière identique (Hall, 1998).

1.3.2.4 Sous ensemble optimal de variables

Définition 11

Soit un algorithme d'induction I, une base de données étiquetées N représentée par les variables $y_1, y_2, \ldots y_n$. S_{opt} est un sous ensemble optimal de variables tel que la précision du classifieur $C=I(N)$ est maximale (Kohavi, et al., 1997).

La pertinence et la corrélation de variables sont des notions importantes pour élaborer des critères pour le filtrage de données. Toutefois ces définitions ne prennent pas en compte le cas ou une variable est corrélée par morceaux à une ou à un ensemble de variables. C'est à dire que la combinaison de variables permet la description d'un concept cible, cette description peut être partielle (sur un ensemble de la base de données). Notre contribution

majeure concerne la sélection de variables pertinentes à l'aide d'un algorithme particulièrement efficace pour le traitement des données fortement corrélées et la détection des variables redondantes.

1.4 Prétraitement : discrétisation des attributs

Les bases de données réelles sont généralement constituées de variables de différentes natures (continues, nominales, ordinales). Cependant, un nombre assez important d'algorithmes d'apprentissage ainsi que la plupart des algorithmes de sélection de variables ont été conçu pour traiter les attributs discrets «symboliques». Les algorithmes de sélection de variables que nous avons utilisés dans nos travaux nécessitent des attributs discrets ou donnent de meilleurs résultats quand ces derniers sont discrétisés. Pour pouvoir appliquer les algorithmes existants à ces données continues, il faut donc passer par la discrétisation (Kerber, 1992), (Liu, & Motoda, 1998), (Lereno, 2000), (Graja, 2008). La discrétisation présente plusieurs avantages, entre autre elle permet de diminuer le temps d'exécution des algorithmes et d'éliminer certaines variables non pertinentes ce qui est le cas de l'algorithme chi2 (Liu, et al., 1995).

Les travaux portant sur la discrétisation peuvent être classés selon trois axes :

1. Les méthodes supervisées ou non supervisées
    a. supervisées : on tient compte de la classe d'appartenance de chacun des objets ;
    b. non supervisées (aveugle) : on ne tient compte que de la similarité des objets sans se préoccuper de leur classe d'appartenance respective.

2. Les méthodes locales ou globales
    a. locales : elles définissent localement les bornes et réalisent la discrétisation pendant la phase d'apprentissage en même temps avec l'utilisation de la variable.
    b. globales : la discrétisation est réalisée en prétraitement.

3. Les méthodes de discrétisation statiques et dynamiques
    a. statiques : la discrétisation a lieu sur chaque variable indépendamment des autres.
    b. dynamiques : les variables sont appréhendées ensemble afin de tenir compte des éventuelles interactions.

Etant donné notre intérêt pour les algorithmes d'apprentissage supervisés et les méthodes filtres de sélection de variables, nous focalisons notre attention sur les algorithmes de discrétisation statique, globale et supervisée. Dans cette section, nous commençons par présenter le formalisme lié au problème de discrétisation, ensuite nous détaillerons les différentes étapes de ce processus. Après, nous étudions les principaux algorithmes que nous avons utilisés dans le cadre de ce travail.

1.4.1 Formalisme et notation

Soit l'attribut continu A à valeurs réelles et défini sur un domaine D_A. Pour tout exemple ω issu d'un échantillon d'apprentissage noté Ω, $A(\omega)$ désigne la valeur prise par cet exemple pour l'attribut A.

($A(\omega) \in R$) et $Y(\omega)$ désigne la classe de l'exemple. Si l'exemple appartient à la classe $y_i(i = 1,..., L)$, alors nous pouvons écrire $Y(\omega) = y_i$.

Discrétiser l'attribut A revient à découper D_A en K intervalles I_i, $1 \le i \le K$ tels que :

$$\bigcup_{i=1}^{K} I_i = \mathrm{D}_A \tag{1.1}$$

$$\bigcap_{i=1}^{K} I_i = 0 \qquad \forall \quad 1 \le i \le K \tag{1.2}$$

Supposons que l'espace de définition I de la variable continue A est représenté par l'intervalle $[d_{min}, d_{max}]$. On veut découper I en P sous intervalles.

$$I_1 = [d_{\min}, d_1[, ..., I_j = [d_{j-1}, d_j[, ..., I_K = [d_{K-1}, d_{\max}[\quad (1.3)$$

Il s'agit de trouver une suite finie strictement croissante de points de discrétisation d_j ($j=1...K$). Une fois les points de discrétisation trouvés, l'attribut A est remplacé par un attribut A^* qui prend ses valeurs dans l'ensemble $\{1, ..., K\}$ de la manière suivante :

$$A^*(\omega) = \begin{cases} 1 & \text{si} \quad A(\omega) < d_{p-1} \\ i & \text{si} \quad d_{i-1} \le A(\omega) < d_i \\ P & \text{si} \quad A(\omega) \ge d_{p-1} \end{cases} \quad (1.4)$$

Cela revient à transformer un vecteur de données initialement numérique en un vecteur disjonctif. Les intervalles trouvés sont remplacés par des noms symboliques.

Nous présentons au paragraphe suivant les algorithmes de discrétisation que nous avons étudié.

1.4.2 Algorithme GhiMerge

La méthode de ChiMerge (Kerber, 1992) est la méthode la plus connue de fusion d'intervalles. Le critère statistique χ^2 est utilisé pour décider s'il faut fusionner deux intervalles adjacents.

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \quad (1.5)$$

Avec :

k = nombre de classes

m = 2 (les deux intervalles à comparer).

A_{ij} = le nombre d'exemples du $i^{ème}$ intervalle appartenant à la $j^{ème}$ classe.

$E_{ij} = \frac{R_i - C_j}{N}$, avec : R_i = le nombre d'exemples d'apprentissage du $i^{ème}$ intervalle et

C_j = le nombre d'exemples appartenant à la $j^{ème}$ classe.

χ^2 (α, *L-1*) est la valeur lue dans la table du Chi 2 au risque α et à (*L-1*) degré de liberté.

L'étape d'initialisation consiste à trier les exemples d'apprentissage en ordre croissant de valeurs de l'attribut à discrétiser. Ensuite, construire autant d'intervalles que de nombre d'exemples (chaque exemple est mis dans un intervalle). Le processus de fusionnement des intervalles continu jusqu'au moment où tous les intervalles adjacents auront une valeur de χ^2 excédant le seuil χ^2_{seuil} désiré (les intervalles adjacents sont alors significativement différents par le test d'indépendance). La valeur de χ^2_{seuil} est choisie en fonction du niveau d'indépendance désiré, elle est lue dans la table du Chi 2 (pour un risque donné et un degré de liberté fixé). Pendant chaque itération du processus de fusion la valeur χ^2 de chaque paire d'intervalles adjacents est calculée. La paire d'intervalles ayant la valeur minimale de χ^2 est fusionnée.

1.4.3 Algorithme Chi2

Dans l'algorithme ChiMerge la valeur de α ainsi que celle du seuil χ^2 (α, *L-1*) doit être spécifiée pour chaque variable. Ce choix n'est pas si évident, une valeur trop petite ou trop grande de α a un impact négatif sur la discrétisation. Liu, et al. (Liu, et al., 1995) proposent un nouvel algorithme, Chi2, où α est automatiquement déterminé à partir des données.

Soit *D* l'ensemble des données formées par les attributs à discrétiser et leurs instances. L'algorithme Chi2 se devise en deux phases (Graja, 2008) :

Phase 1 :

Au début, α est initialisé à une valeur assez grande, par exemple 0.5, ensuite pour chaque attribut de *D*, les instances sont ordonnées et les points de discrétisation initiaux calculés. Après le χ^2 est calculé pour chaque paire d'intervalles adjacents afin de

déterminer la plus petite valeur χ^2_{min}. Si cette dernière est inférieure au seuil χ^2 (*α, L-1*) les deux intervalles doivent êtres fusionnés.

Une fois toutes les paires traitées, α doit êtres diminué, et le processus (calcul de χ, fusionnement, diminution de α) est répété jusqu'à ce que le taux d'inconsistance des données excède un seuil prédéfini δ.

Phase 2 :

C'est une étape de finition. Partant du dernier seuil α_0 de la phase 1, les variables sont traitées une à une. Cette étape servira à arrêter le processus de la phase 1 alors que certains attributs peuvent être encore traités sans influer sur le taux d'inconsistance.

1.4.4 Algorithme MDLM

L'algorithme MDLM est un algorithme de discrétisation de variables continues conçu par Fayyad et Irani (Fayyad & Irani, 1993). Les auteurs proposent d'utiliser une mesure d'entropie et démontrent qu'un point de discrétisation maximisant cette mesure d'entropie est un point frontière.

Soit A l'attribut à discrétiser et N un ensemble d'instances caractérisées par A. Par définition F est un point frontière si :

$$\exists\, \omega_1, \omega_2 \in N \text{ tel que : } \begin{cases} Y(\omega_1) \neq Y(\omega_2) \\ A(\omega_1) < F < A(\omega_2) \end{cases} \tag{1.6}$$

$$\exists\, \omega_3 \in N \text{ tel que : } A(\omega_1) < A(\omega_3) < A(\omega_2)$$

Soit d un point de coupure. Nous notons N_1 le sous ensemble d'instances de N dont la valeur excède d et $N_2=N- N_1$. Par définition, l'entropie des classes induite par d notée $I(A,d\,;N)$ est :

$$I(A, d\,; N) = \frac{|N_1|}{|N|} I(N_1) + \frac{|N_2|}{|N|} I(N_2) \tag{1.7}$$

L'algorithme MDLM est un algorithme récursif qui se base sur la maximisation du gain d'information pour trouver les points de discrétisation. L'Algorithme cherche à chaque fois la meilleure partition binaire maximisant le gain d'information.

1.5 Méthodes de classification

Le but de notre travail est dans un premier temps de combiner les méthodes de sélection de variables avec des classifieurs (algorithmes de classification) afin d'améliorer les performances de classification de ces derniers et dans un deuxième temps, proposer un processus le plus fiable possible pour le diagnostic de défauts, nous commençons par donner quelques notions de base liées au problème d'induction et plus généralement à l'apprentissage automatique (*Machine Learning*) (Arsselin, et al., 2005), (Russell, et al., 2010), (Cornuéjols, et al., 2011). Ensuite, nous présentons les classifieurs utilisés dans nos travaux avec leurs avantages et inconvénients, des indications pour mieux les choisir.

Les algorithmes d'induction sont des méthodes ou les classes de la population initiale sont connues. Ces méthodes sont les plus couramment utilisées en fouille de données et sont identifiées comme procédés de classification. Dans l'apprentissage inductif (l'apprentissage supervisé ou de concepts à partir d'exemples), les observations caractérisent des objets étiquetés (pré-classés) par un expert en une ou plusieurs classes (concepts) représentées par les valeurs (modalités) : $y_{but} = \{m_1^{but}, ..., m_l^{but}, ..., m_p^{but}\}$ de la variable à expliquer (variable exogène) y_{but}. Les objets sont représentés par des enregistrements (ou descriptions) qui sont constitués d'un ensemble de champs ou attributs (variable endogène) $y = \{y_1, ..., y_r\}$, prenant leurs valeurs dans un domaine. L'hypothèse induite (la classification) peut être vue comme une règle de reconnaissance de concept. On a alors : $m_u^{but} = f(y_1, ..., y_r)$.

Si un objet vérifie cette règle, alors il représente le concept donné. « Si les données fournies à une méthode d'apprentissage sont des exemples classés par une source de connaissance indépendante - un expert ou un modèle de simulation - il s'agit

d'apprentissage à partir d'exemples » (Kodratoff, et al., 1991(a)). La dénomination anglaise pour ce type d'apprentissage et *classification.*

Supposons que l'on dispose d'une population convenablement étiquetée. Le processus d'obtention d'un classifieur peut être généralement résumé en deux étapes principales. L'une des méthodes les plus répandues consiste à diviser l'ensemble d'apprentissage en deux sous-ensembles : un ensemble d'entraînement $\mathbf{\Omega_1}$ (training set) et un ensemble test $\mathbf{\Omega_2}$ (test set). Le premier de ces ensembles est utilisé afin d'induire un classifieur et le second sert à la validation du classifieur obtenu.

1.5.1 Apprentissage croisé

Typiquement, l'évolution de l'erreur, pour l'apprentissage et le test, en fonction des itérations d'apprentissage, suit les courbes de la figure (1-2), l'erreur diminue toujours, alors que sur la base de test elle passe par un minimum. Si l'apprentissage se prolonge au delà, les performances en test diminuent, c'est ce qu'on appel le sur-apprentissage (*overfitting*), une solution pour trouver la solution optimale est d'effectuer la procédure d'apprentissage par validation croisée.

La technique divise les données disponibles en trois sous-ensembles. Le premier sous-ensemble est l'ensemble d'apprentissage, qui est employé pour construire le modèle du classifieur. Le deuxième sous-ensemble est l'ensemble de validation qui est prise en compte pendant le processus de d'apprentissage. L'erreur diminuera normalement pendant la phase initiale de l'apprentissage. Cependant, quand le classifieur entre dans la phase du sur-apprentissage, l'erreur sur l'ensemble de validation commencera typiquement à monter. Quand l'erreur de validation augmente pour un nombre indiqué d'itérations, l'apprentissage est arrêté, le modèle est retourné, le troisième sous ensemble d'essai n'est pas employé pendant l'apprentissage, mais il est employé pour évaluer le classifieur ainsi conçu.

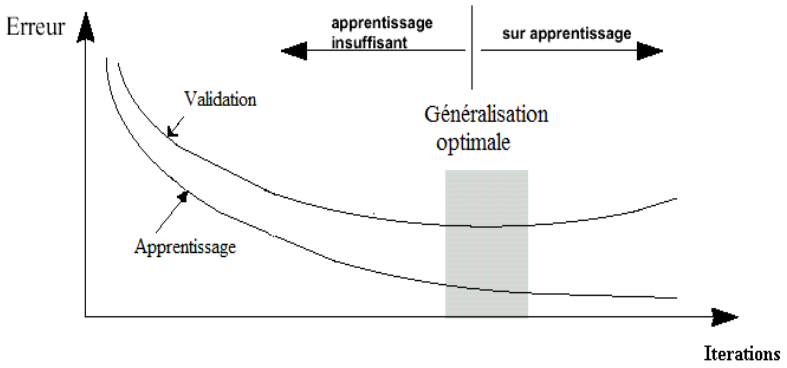

Figure 1-2 : *Evolution de l'erreur sur les bases d'apprentissage et validation au cours de l'apprentissage et de la généralisation.*

1.5.2 Le choix d'une méthode de classification

La performance est un aspect du comportement d'un classifieur qui peut être mesuré quantitativement, c'est-à-dire en utilisant un indicateur mesurable donné. C'est pourquoi les objectifs du système d'apprentissage doivent être clairement définis au préalable. D'une manière générale on souhaitera que le classifieur obtenu offre de bonnes performances en termes de :

Généralisation : le classifieur « résume » l'ensemble des données disponibles sous une structure donnée (arbre de décision, réseaux de neurones, etc.) et il doit être ainsi capable de prédire à quelle classe appartient un objet qu'il ne connaît pas. La généralisation exprime donc la capacité prédictive du classifieur.

Compréhensibilité : il est admis qu'un humain éprouve des difficultés à déceler les structures sous-jacentes d'un ensemble de données, même si l'ensemble est relativement simple. Toutefois, une représentation structurelle des données, permet une meilleure compréhension car cette représentation des données est plus concise que la population initiale. En conséquence plus une représentation est simple et plus elle sera potentiellement intelligible et réutilisable comme c'est le cas par exemple des arbres de décision.

La performance d'un calssifieur peut ainsi être mesurée par sa précision prédictive mais aussi par sa vitesse d'apprentissage, ou encore sa compréhensibilité. Deux paramètres important conditionnent donc la bonne performance d'un classifieur : les données disponibles ainsi que l'algorithme d'apprentissage utilisé.

1.5.3 L'analyse discriminante

L'analyse discriminante (classifieur de Bayes) est une technique statistique de classification se basant sur la règle de Bayes (John, et al., 1995), (Arsselin, et al., 2005), (Cornuéjols, et al., 2011). Elle repose sur l'estimation de densité de probabilité, les données sont supposées générées par un processus inconnu. Étant donné un échantillon observé Ω composé de n objets ou individus ou instances $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$, c_i représente les différentes classes formant l'espace des exemples. Chaque élément de Ω est caractérisé par un ensemble de r variables ou attributs $X = \{x_1, x_2, \ldots, x_r\}$. De ce fait, les méthodes statistiques ne doivent s'appuyer que sur les coordonnées des individus dans l'espace de description afin de déterminer dans celui-ci plusieurs domaines qui correspondent aux classes. Si nous notons $P(\omega)$ la densité de probabilité d'existence du prototype ω, et $P(\omega|c_i)$ la densité de l'instance ω lorsqu'on sait qu'il appartient à la classe c_i, on a :

$$P(\omega) = \sum P(\omega|c_i) P(c_i) \tag{1.8}$$

La règle de Bayes s'énonce de la manière suivante :

Choisir la classe c_i maximisant la densité de probabilité $P(c_i|\omega)$

avec:

$$P(c_i|\omega) = \frac{P(\omega|c_i) P(c_i)}{P(\omega)} \quad \textit{loi de Bayes} \tag{1.9}$$

Dans le cas général, la loi de Bayes se présente sous la forme suivante :

$$P\left(c_i \left| \omega_1, ... \omega_l \right.\right) = \frac{P\left(\omega \left| c_i \right.\right) P\left(c_i\right)}{\sum_{i=1}^{m} P\left(\omega \left| c_i \right.\right) P\left(c_i\right)} \tag{1.10}$$

où:

l : représente le nombre d'instances à classer

m : représente le nombre de classes

La règle de bayes nous donne la densité de probabilité à postériori à partir de la probabilité à priori sur les classes et de la densité de probabilité conditionnelle de ω_i sachant que ω_i appartient à cette même classe. La solution optimale de mettre une observation ω dans la classe c_i est obtenue si et seulement si :

$$\forall j \text{ , } P(c_i \left| \omega \right.) > P(c_j \left| \omega \right.) \tag{1.11}$$

C'est-à-dire que la classe c_i est la plus probable à contenir l'exemple ω.

On remplaçant $P\left(c_i \left| \omega \right.\right)$ par son expression dans l'équation (1.9), on aura:

$$P\left(\omega \left| c_i \right.\right) P\left(c_i\right) > P\left(\omega \left| c_j \right.\right) P\left(c_j\right) \tag{1.12}$$

En pratique, seul le numérateur est pris en considération, puisque le dénominateur ne dépend pas de la variable de concept *C* et les valeurs des variables descriptives x_i sont données. Le dénominateur est donc constant.

La règle de décision de Bayes consiste donc à choisir d'affecter l'individu à la classe dont la probabilité à postériori (qui a été calculée par la formule de Bayes) est la plus grande. La décision ainsi prise minimise le risque de l'erreur de classification. Ce qui est l'objectif de toute méthode de classification.

La formule de Bayes permet de déterminer les probabilités d'appartenance à postériori si les densités de probabilité et les probabilités à priori sont connues. La phase d'apprentissage consiste à estimer tous les paramètres du modèle (probabilités *à priori* des

classes et lois de probabilités associées aux différentes variables descriptives). Pour ce faire il existe deux groupes de méthode :

- Les méthodes paramétriques qui font une hypothèse sur la forme analytique de la distribution de probabilité.
- Les méthodes non paramétriques qui ne font aucune hypothèse sur la forme de la distribution de probabilité.

L'approche paramétrique comme c'est le cas par exemple du classifieur Naive Bayes (John, et al., 1995) nécessite de faire des hypothèses sur la nature de la loi de probabilité en supposant que les variables sont indépendantes. Cependant, pour des problèmes réels complexes, il est impossible de connaître le modèle des données. Dans ce cas on a recours aux méthodes non-paramétriques telles que les k-plus proches voisins, les arbres de décision et les réseaux de neurones. C'est la raison pour laquelle, nous avons opté pour le choix de ces méthodes dans nos travaux de thèse.

1.5.4 Les *k* plus proches voisins

Les k plus proches voisins plus connus en anglais sous le nom *K-Nearest Neighbor* (*K-NN*) (Aha, et al., 1991), (Russell, et al., 2010) est une méthode d'apprentissage non paramétrique qui ne nécessite pas de construction de modèle, C'est l'échantillon d'apprentissage, associé à une fonction de distance et d'une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle.

Pour prédire la classe d'un exemple donné ω, l'algorithme cherche les *k* plus proches voisins de ce nouveau cas et prédit la réponse la plus fréquente de ces *k* plus proches voisins. Le principe de décision consiste tout simplement donc à calculer la distance de la exemple inconnu ω à tous les échantillons fournis. L'exemple est alors affecté à la classe majoritairement représentée parmi ces *k* échantillons. La méthode utilise deux paramètres : le nombre *k* et la fonction de similarité pour comparer le nouvel exemple aux exemples déjà classés.

1.5.4.1 Apprentissage et classification

Soit Ω l'ensemble de données d'apprentissage : $\Omega = \{(X, C)\}$

Avec $C \in \{1,...,m\}$ représente la classe de l'exemple i et $X = (x_1,....,x_r)$ les variables descriptives.

Il y a plusieurs choix possibles pour les distances, tel que la distance euclidienne définie comme suit :

$$d(\omega_i, \omega_j) = \sqrt{\sum_{k=1}^{r} \left(X_k(\omega_i) - X_k(\omega_j) \right)^2} \tag{1.13}$$

ou bien la distance de Minkowski :

$$d(\omega_i, \omega_j) = \left(\sum_{k=1}^{r} \left| X_k(\omega_i) - X_k(\omega_j) \right|^q \right)^q \tag{1.14}$$

Quand *k=1*, l'exemple ω_i est attribué à la classe de l'exemple qui lui est le plus proche Si *k*> 1, ce sont tous les *k* plus proches observations qui sont prises en considération dans la prise de décision. Ainsi la décision est en faveur de la classe majoritairement représentée par les *k* voisins. Soit k_i le nombre d'observations issues du groupe des plus proches voisins appartenant à la classe *i*.

$$\sum_{i=1}^{m} k_i = k \tag{1.15}$$

Ainsi une nouvelle observation est prédite dans la classe *C* avec :

$$C = \max(k_i) \tag{1.16}$$

Il n'existe pas de distance universellement optimale, cependant une bonne connaissance du problème traité peut guider le choix de cette distance.

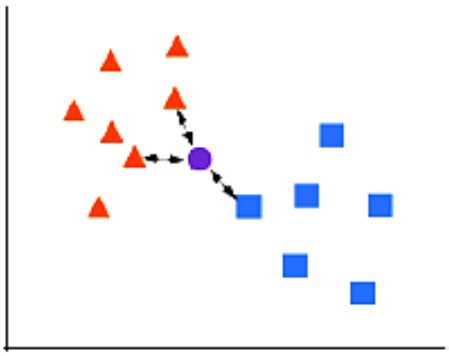

Figure 1-3 : *Décision selon les 3 plus proches voisins*

Le paramètre *k* doit être déterminé par l'utilisateur. L'exemple de la figure (1-4) illustre l'influence de *k* sur les frontières de décision. En effet une grande valeur de *k* réduit l'effet du bruit sur la classification et donc le risque de sur-apprentissage, par contre une faible valeur de *k* permet de mieux visualiser les frontières entre classes et donc éviter un 'oversmoothing' ou sur-lissage.

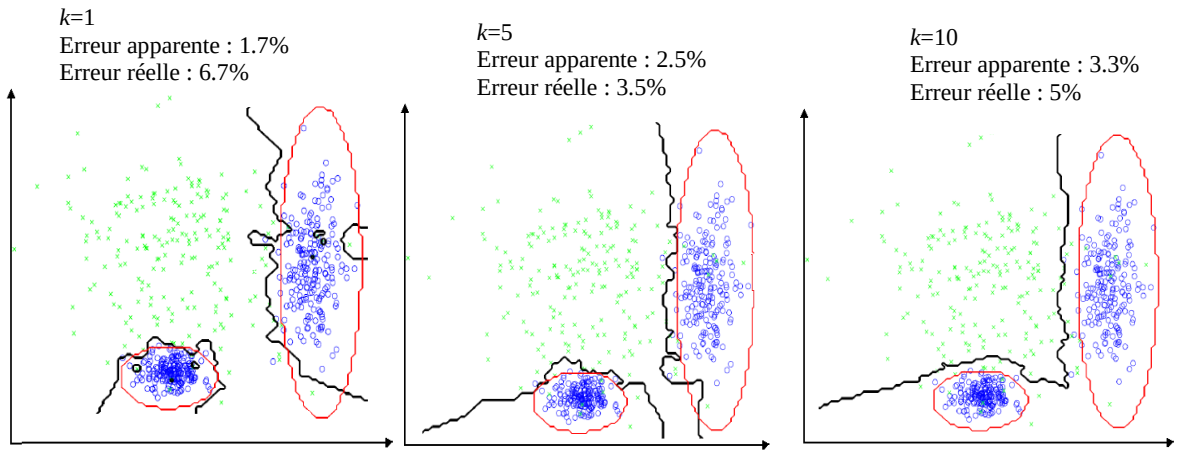

Figure 1-4 : *Influence de k sur les frontières de décision*

Un bon choix pour *k* peut être réalisé grâce à des techniques heuristiques, par exemple, la validation-croisée. La valeur de *k* qui minimise l'erreur de classification totale c'est à dire aussi bien l'erreur apparente (calculée sur l'ensemble de données test) que l'erreur réelle (calculée sur l'ensemble de données de validation) est choisie.

Algorithme KNN

Début

Paramètre : Le nombre *k* de voisins

Données : Un échantillon de n exemples d'apprentissage $\Omega = (\omega_1, ..., \omega_n)$

La classe d'un exemple ω est Y(ω), $Y=\{C_1, C_2, \ldots C_m\}$

Entrée : un enregistrement *X*

Pour chaque exemple ω **faire**

Calculer la distance d(X,ω) ;

fin

KNN= les *k* plus proches voisins de *X* qui minimise la distance *d* ;

Pour chaque $\{\omega \in KNN\}$ **faire**

Calculer les scores des classes ;

fin

Attribuer *Y*(*X*) à la classe ayant le plus grand score;

Sortie : la classe de *X* est $Y(X)=C_j$;

fin

1.5.4.2 Algorithme général

L'algorithme des *k* plus proches voisins est une méthode d'apprentissage à base d'exemples. Il ne comporte pas de phase d'apprentissage en tant que telle. Les données faisant partie de l'ensemble d'entraînement sont seulement emmagasinés. Lorsqu'un nouvel exemple à classer arrive, il est comparé aux exemples (prototypes) d'entraînement à l'aide d'une mesure de similarité. Bien que la phase d'apprentissage est inexistante puisque les données sont stockées telles quelles en mémoire, la classification d'un nouveau cas est par

contre coûteuse puisqu'il faut comparer ce cas à tous les exemples déjà classés. En plus un *KNN* de base utilise toutes les variables descriptives d'un exemple pour calculer la similarité avec un nouvel exemple à classer. Hors dans des bases de données à grandes dimensions ce qui est le cas en data mining, les variables discriminantes sont considérées au même titre que les autres (les moins discriminantes). Pour remédier à ce problème les variables sont pondérées suivant leur pertinence. Afin de choisir ces poids, la méthode de validation croisée peut êtres utilisée (Mathieu-Dupas, 2010). Les performances de la méthode dépendent du nombre de voisins, du choix de la distance, et du mode de combinaison des réponses des voisins.

1.5.5 Les réseaux de neurones

Les Réseaux de Neurones Artificiels ou formels (RNA) sont des modèles mathématiques imitant la structure et les fonctions des réseaux de neurones biologique. Ce sont des outils très utilisés pour la classification, l'estimation, la prédiction et la segmentation. Ils sont aussi capable de trouver la même solution que celle fournie par la formule de Bayes, sans condition particulière (Zurada, 1992), (Herault, 1994), (Antoine Cornuéjols, 2011). En effet on va montrer que la sortie d'un réseau de neurones est une estimation des probabilités *a postériori* d'appartenance aux classes. Nous nous limitons dans ce chapitre aux réseaux de neurones dédiés aux tâches d'estimation et classification que sont les *Perceptrons Multicouches* (*PMC*) *ou Multilayer Perceptron* (*MLP*). Dans cette section de présentation des réseaux de neurones formels, nous commençons par donner quelques définitions relatives aux réseaux de neurones à couches, ensuite, nous présenterons l'algorithme d'apprentissage qui sert à entrainer ce type de réseau : l'algorithme de rétro-propagation du gradient.

1.5.5.1 Le neurone biologique

Un neurone est une cellule constituée principalement de trois parties (Figure 1-5) : ce sont les dendrites, le soma et l'axone. Les dendrites collectent les signaux venant d'autres cellules au niveau de points de contact avec les autres neurones appelés synapses.

L'information est ensuite acheminée vers le corps cellulaire ou soma qui recueille et concentre l'ensemble des informations reçues par les dendrites. Les réseaux de neurones artificiels sont des architectures artificielles inspirées à partir d'un tel fonctionnement.

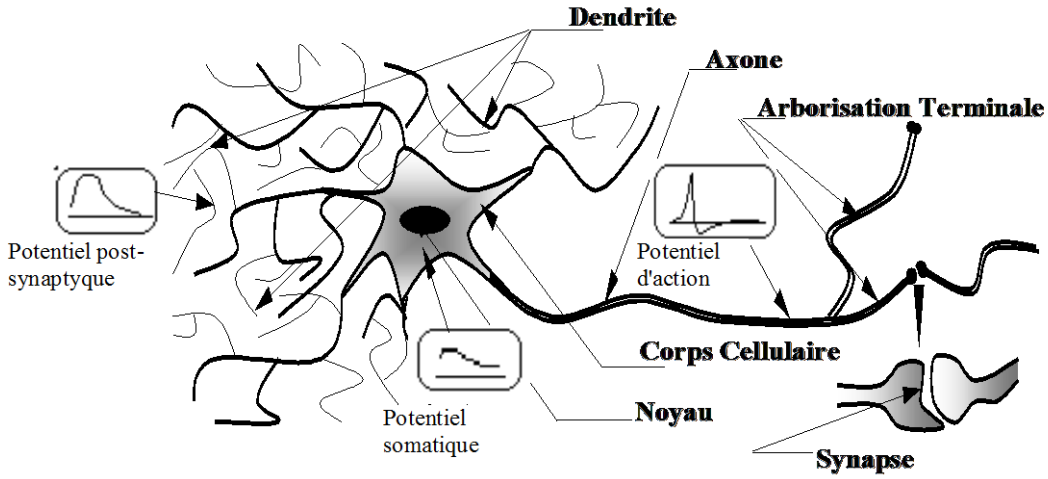

Figure 1-5 : *Neurone typique de vertébré*

1.5.5.2 Le neurone formel

Dés l'établissement du fonctionnement réel d'un neurone biologique, plusieurs modèles ont été proposés, dont le but principal est de refléter ce fonctionnement. Le plus important est celui de MC Culloch et Pitts établit en 1943.

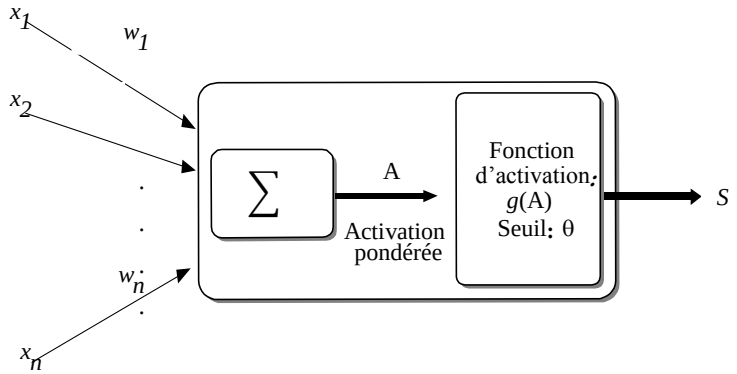

Figure 1-6 : *Neurone modélisation générale*

Dans ce modèle, chaque unité de traitement (ou neurone formel) calcule une somme pondérée des signaux fournis par les autres unités auxquelles elle est connectée et ce résultat est comparé à une valeur de seuil. Si le seuil est franchi, le neurone est activé et cet état d'activation est propagé à travers le réseau pour les étapes de calcul ultérieures. Le fonctionnement du neurone formel est modélisé par la fonction suivante :

$$S = g\left(\sum_{i=1}^{n} \omega_i x_i + \theta\right) \tag{1.17}$$

Où ω_i est appelé le poids de la connexion *i*, le neurone est donc constitué de deux modules successifs (Figure 1-6) : une transformation linéaire (le produit scalaire) qui multiplie chaque valeur d'entrée par la composante des poids correspondante à laquelle est ajouté un terme constant, le biais. Ensuite, on applique une transformation *g* non linéaire en général à ce potentiel. La combinaison linéaire est appelée potentiel ou entrée totale et la transformation non linéaire *g* la fonction d'activation.

Le biais Θ est le seuil interne du neurone, il peut êtres envisagé comme le coefficient de pondération n°0 de l'entrée du neurone et la sortie devient : $S = g\left(\sum_{i=0}^{n} \omega_i x_i\right)$.

La nature de ces entrées peut être binaires (0,1) ou (-1,1) ou bien réelles, la fonction d'entrée totale $\left(\sum_{i=1}^{n} \omega_i x_i\right)$ définit le prétraitement effectué sur les entrées, la fonction d'activation définit l'état interne du neurone en fonction de son entrée totale, elle est souvent choisie de façon à avoir une sortie comprise entre 0 et 1 et peut être:

- Une fonction binaire à seuil (fonction signe ou fonction de Heaviside).
- Une fonction linéaire à seuil ou multi-seuil.
- Une fonction sigmoïde.
- Une fonction stochastique.
- Toute autre fonction croissante et impaire.

Un premier choix possible pour la fonction d'activation est la fonction de Heaviside (définie par g(A) =1 si A>0 et g(A)=0 sinon). Cette fonction n'est pas dérivable et les réseaux de neurones qui sont entraînés par des algorithmes de gradient utilisent plus souvent des approximations dérivables de cette fonction. Parmi elles, la fonction la plus utilisée est la fonction sigmoïde (ou tangente hyperbolique).

1.5.5.3 Architecture des réseaux de neurones

Il existe deux structures de réseau de neurones, en fonction du graphe de leurs connexions :

- Les réseaux bouclés (dynamique) :

  Ces réseaux sont utilisés pour la modélisation dynamique de processus non linéaires et pour leur commande. Le graphe des connexions est cyclique : la sortie de chaque neurone est reliée à l'entrée d'un ou plusieurs neurones en aval. Ces réseaux sont décrits par un système d'équations aux différences. Ils sont le siège de contre réactions en fonction du temps. Ils sont particulièrement adaptés pour construire des réseaux de type de Hopfield ou de Boltzman avec des procédures d'apprentissage sans professeur (non-supervisé).

- Les réseaux non bouclés (ou statique) :

  Dans ces réseaux, le flux d'information circule des entrées vers les sorties sans retour en arrière. Si l'on représente le réseau comme un graphe dont les nœuds sont les neurones et les arêtes les connexions entre ceux-ci. Il existe deux types de réseaux de neurones : les réseaux complètement connectés (Chaque neurone du réseau est connecté à tous les autres neurones ainsi qu'à lui-même) et les réseaux à couches (Figure 1-7). Le réseau de neurones *Perceptron multicouches* est un cas particulier de ce dernier type. C'est la structure que nous avons utilisée dans le cadre de la résolution des problèmes de classification présenté dans ce mémoire et que nous allons détailler dans ce qui suit.

Les réseaux non bouclés sont souvent appelés « réseaux statiques », le temps ne joue aucun rôle fonctionnel : lorsque les entrées sont constantes, les sorties le sont aussi et le temps nécessaire pour le calcul d'une fonction donnée est négligeable et peut êtres considérer comme instantané.

1.5.5.4 Le Perceptron multicouches

Les neurones sont organisés en couches où tous les neurones d'une couche sont connectés aux neurones de la couche suivante, la première couche s'appelle couche d'entrée ou rétine : aucun calcul n'est effectué, une cellule d'entrée ne fait que copier son entrée vers sa sortie. Les entrées correspondent aux attributs (ou leurs codages) du problème considéré, la dernière couche s'appelle couche de sortie et les couches intermédiaires s'appellent couches cachées. Tous les neurones d'une couche sont les entrées de chaque neurone de la couche suivante seulement : autrement dit, il n'y a pas de retour arrière et on passe d'une couche à la suivante. Les calculs sont effectués des entrées vers la ou les sorties, le neurone élémentaire est, en règle générale, celui considéré dans le paragraphe précédent. Les sorties de ces neurones correspondent aux valeurs à estimer (ou leurs codages) pour le problème considéré. Les restrictions introduites dans ce type de réseaux sont requises pour l'établissement de l'algorithme d'apprentissage que nous allons détailler par la suite. Ce type de réseaux est souvent appelé "multi-layer perceptron" (MLP) (Herault, 1994), (Senoussi, 2001).

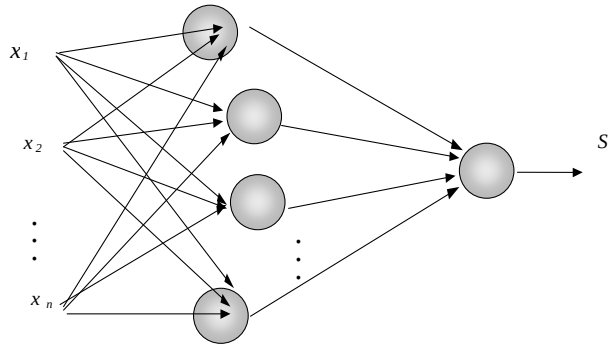

Figure 1-7 : *Un perceptron multicouche : une couche d'entrée de n cellules, 1 couche cachée et 1 couche de sortie à 1 neurone.*

1.5.5.5 L'algorithme d'apprentissage de rétro-propagation du gradient

Depuis les premiers travaux sur le neurone formel il a fallu attendre le début des années 80 pour qu'un algorithme d'apprentissage efficace (par rapport à la résolution des problèmes non linéaires) apparaisse : *l'algorithme de rétro-propagation du gradient* (Werbos, 1990). Les neurones utilisés sont de type sommateur à fonction d'activation continue et dérivable. En général la fonction d'activation retenue est la même pour tous les neurones d'une couche, voire pour tous les neurones du réseau. La fonction d'activation choisie est souvent une fonction sigmoïde. Ce choix qui satisfait aux exigences de mise en œuvre de l'algorithme se justifie en outre par les propriétés mathématiques qu'il confère au réseau de neurones. Cet algorithme est une généralisation de la règle de Widrow-Hoff pour un réseau multi-couche.

A un vecteur d'entrée *x*, on associe un vecteur de sortie *S*. Si les poids ω_{ij} ont des valeurs quelconques (ce qui est le cas initialement et avant la fin de l'apprentissage), le vecteur de sortie *S* observé est à priori différent de la sortie désirée *d*. On peut associer à cette différence une fonction de coût *E*.

L'apprentissage est un ensemble de règles qui déterminent les valeurs des poids ou connexions pour que le réseau accomplisse une certaine tâche. La règle d'apprentissage de Widrow-Hoff est basée sur l'idée de réduire progressivement la différence entre la sortie obtenue et la sortie désirée. En faisant varier les poids, cette règle est donnée par:

$$\Delta\omega_{ij}\left(t+\Delta t\right)=\lambda\left(d_i-s_i\right)x_j \tag{1.18}$$

λ : Pas d'adaptation ou vitesse d'apprentissage.

L'objectif de l'algorithme rétro-propagation du gradient de est de minimiser une fonction de coût. L'équation suivante exprime cette fonction de coût à partir de l'erreur quadratique, pour un couple entrée-sortie, avec d_i la sortie désirée pour le neurone d'indice *i* et S_i la sortie obtenue par le réseau.

$$E = \sum_i \left(d_i - s_i\right)^2 \tag{1.19}$$

L'application de la règle de la mise à jour des poids donne:

$$\omega_{ij} = \omega_{ij} - \frac{\delta E}{\delta \omega_{ij}} \tag{1.20}$$

L'approche la plus utilisée pour la minimisation de la fonction E est basée sur la méthode du gradient. On commence l'entraînement par un choix aléatoire des vecteurs initiaux du poids. L'apprentissage comporte une première phase de calcul dans le sens direct (de l'entrée vers la sortie du réseau de neurones) où chaque neurone effectue la somme pondérée de ses entrées et applique ensuite la fonction d'activation g. L'équation (1.21) correspond à ce calcul avec A_i l'entrée totale du neurone i, x_j l'état du neurone de la couche précédente et ω_{ij} le poids de la connexion entre les deux neurones.

$$s_i = g(A_i) \qquad \text{où} \qquad A_i = \sum_{j=0}^{n} \omega_{ij} x_j \tag{1.21}$$

Cette phase, dite de *propagation*, permet de calculer la sortie du réseau en fonction de l'entrée.

L'algorithme de rétro-propagation consiste à effectuer une descente de gradient sur le critère *E*. Le gradient de *E* est calculé pour tous les poids de la manière suivante :

$$\frac{\delta E}{\delta \omega_{ij}} = \frac{\delta E}{\delta A_i} \frac{\delta A_i}{\delta \omega_{ij}} = \frac{\delta E}{\delta A_i} x_j \tag{1.22}$$

Le gradient sera ensuite noté C_i :

$$C_i = -\frac{\delta E}{\delta A_i} \tag{1.23}$$

La modification des poids est obtenue suivant l'équation :

$$\omega_{ij}^{t+1} = \omega_{ij}^{t} + \alpha C_i s_j \tag{1.24}$$

où α est la vitesse d'apprentissage : un petit nombre positif qui représente le pas de déplacement en direction du minimum le plus proche.

On distingue alors deux cas, suivant que le neurone d'indice *i* est un neurone de sortie ou non. Dans le cas de la couche de sortie, le gradient attaché aux cellules de sortie est alors obtenu par l'équation suivante:

$$C_i = -\frac{\delta E}{\delta A_i} = -\frac{\delta}{\delta A_i}\left(\sum_k \left(d_k - s_k\right)^2\right) = 2\left(d_i - s_i\right) g'\left(A_i\right) \tag{1.25}$$

Pour les neurones des couches cachées, l'ordre de calcul des gradients est l'inverse de celui utilisé pour la mise à jour des états dans le réseau. Il s'effectue de la couche de sortie vers l'entrée; on parle alors de rétro-propagation. L'expression du gradient est obtenue comme indiqué dans l'équation :

$$C_i = -\frac{\delta E}{\delta A_i} = -\sum_{k=0}^{n} \frac{\delta E}{\delta A_k} \frac{\delta A_k}{\delta A_i} = \sum_{k=0}^{n} C_k \frac{\delta A_k}{\delta A_i} = \sum_{k=0}^{n} C_k \frac{\delta A_k}{\delta s_i} \frac{\delta s_i}{\delta A_i} \tag{1.26}$$

ou bien :

$$C_i = g'(A_i) \sum_{k=0}^{n} \omega_{kj} C_k \tag{1.27}$$

avec C_k le gradient du neurone *k* de la couche suivante dans le sens de la propagation.

Dans le cas de l'algorithme à gradient total, les exemples de la base d'apprentissage sont présentés successivement au réseau, on présente le premier vecteur d'entrée, une fois on a la sortie du réseau, l'erreur correspondante et le gradient de l'erreur par rapport à tous les poids sont calculés. Les poids sont alors ajustés. On refait la même procédure pour tous les exemples d'apprentissage. Ce processus est répété jusqu'à ce que les sorties du réseau soient suffisamment proches des sorties désirées. Les gradients accumulés au fur et à mesure de la modification des poids n'interviennent qu'après présentation de tous les exemples. Par contre dans le cas des réseaux de neurones à gradient stochastique la modification des poids est effectuée pour chaque exemple présenté. L'algorithme de rétro-propagation se résume finalement aux étapes suivantes:

Algorithme de rétro-propagation du gradient

1. Choix de la taille du réseau, initialisation des poids et des seuils (biais) des neurones aléatoirement
2. Choisir les vecteurs d'entrées (instances) et de sorties désirées, correspondants dans la base d'apprentissage
3. E=0; (accumulateur de l'erreur)
4. **pour** chaque exemple (instance) **faire**
    5. propagation : calculer les sorties de chaque neurones *i* de chaque couche

$$s_i = g(A_i) \qquad \text{où} \qquad A_i = \sum_{j=1}^{n} \omega_{ij} x_j$$

    6. rétro-propagation pour chaque neurone *i* de chaque couche

        Calculer l'erreur de sortie en utilisant l'expression

$$C_i = 2g'(A_i)(d_i - s_i)$$

        Calculer l'erreur dans les couches en utilisant l'expression

$$C_i = g'(A_i)\sum_{k=0}^{n} \omega_{kj} C_k$$

    7. mise à jour des poids pour tous les couples (*i*,*j*)

$$\omega_{ij} = \omega_{ij} + \alpha C_i s_j$$

    8. Accumulation de l'erreur

$$E = E + \sum_i (d_i - s_i)^2$$

9. **Si** la condition sur l'erreur ou sur le nombre d'itérations est atteinte **alors** fin **sinon** aller à l'étape 4.

1.5.5.6 Architecture et paramètres d'apprentissage

Il n'existe pas de méthode générale permettant de fixer une architecture de réseau pour un problème donné. Néanmoins, Hornik (Hornik, 1991) a démontré qu'un Perceptron

multicouches avec une seule couche cachée doté d'un nombre suffisant de neurones, peut approximer n'importe quelle fonction avec la précision souhaitée. Cette étude justifie notre choix pour une architecture de réseaux de neurones à une seule couche cachée. En pratique on choisit une architecture (dans notre cas le nombre de neurones cachés) et il reste à trouver (par apprentissage) les paramètres convenables.

Parmi les paramètres qui influencent les résultats de l'apprentissage ainsi que la prédiction des réseaux de neurones multicouches entraînés par l'algorithme de rétro-propagation du gradient (Senoussi, 2001) :

Valeurs initiales des poids :

En pratique on choisit aléatoirement les valeurs initiales des poids entre -A1 et +A2, le choix des bornes dépendent de la dynamique des signaux d'entrée. L'idée de base est qu'il faut éviter de travailler dans la zone saturée de la fonction d'activation (Sigmoïde), cela conduit à une valeur très faible, donc on choisit aléatoirement les valeurs initiales des poids, entre –1 et +1.

Choix du pas d'adaptation :

Le pas d'adaptation α (vitesse d'apprentissage) doit être choisi avec soin si l'on veut une vitesse de convergence adéquate, sans toute fois entraîner une instabilité dans le processus d'apprentissage. La solution consiste à faire varier α en tenant compte des variations de l'erreur au cours de l'apprentissage de la manière suivante :

Posons α_{dec} et α_{inc} deux coefficient qui permettent l'adaptation de la vitesse d'apprentissage tel que :

$$\alpha_{inc} > 1 \qquad et \quad \alpha_{dec} < 1$$

- Quand l'erreur augmente, il faut réduire la vitesse d'apprentissage caractérisée par $\alpha\alpha_{dec}$ et la mise à jour des poids se fait par l'équation :

$$\omega_{ij}^{t+1} = \omega_{ij}^{t} + \alpha\alpha_{dec} C_i s_j$$

- Quand l'erreur diminue, il faut augmenter la vitesse d'apprentissage caractérisée par $\alpha\alpha_{inc}$ et la mise à jour des poids se fait par l'équation :

$$\omega_{ij}^{t+1} = \omega_{ij}^{t} + \alpha\alpha_{inc} C_i s_j$$

Test d'arrêt

La convergence de l'algorithme vers une solution optimale ne peut être assurée en fixant dés le départ un nombre d'itérations. En pratique, il faut choisir un critère de convergence. On cherche à arrêter l'algorithme si l'erreur *E* est minimale, c'est -à-dire si le gradient de l'erreur est nul. Il existe trois tests possibles :

1. Le module du gradient est proche de zéro, dans ce cas le réseau risque de prendre le premier minimum trouvé et non pas le minimum adéquat qui est la solution désirée (Figure 1-8).

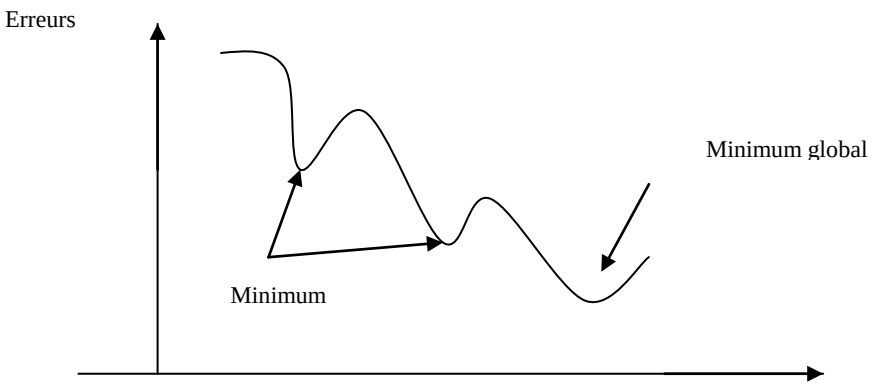

Figure 1-8 : *Evolution de l'erreur quadratique moyenne sur les bases d'apprentissage et de validation au cours de l'apprentissage*.

2. *E* est inférieur à un seuil E_{min}

3. Apprentissage croisé : Si l'apprentissage se prolonge au-delà d'un certain seuil, les performances en test diminuent, c'est ce qu'on appel le sur-apprentissage

(overfitting), une solution pour trouver la solution optimale est d'effectuer la procédure d'apprentissage par validation croisée.

Le moment

Afin d'éviter les minimums locaux, qui ne sons pas désirés, on ajoute un paramètre « le moment » dans la règle de la mise à jour des poids qui devient :

$$\omega_{ij}^{t+1} = \omega_{ij}^{t} + m\left(\omega_{ij}^{t} + \omega_{ij}^{t-1}\right) + (1-m)\alpha C_i s_j$$

L'introduction du paramètre *m* à deux effets :

- Le premier est d'amortir la variation des poids en introduisant un effet de mémoire des variations passées. Le second est de donner aux réseaux la possibilité d'améliorer la solution trouvée, autrement dit permettre aux réseaux de chercher un minimum de la fonction coût plus adéquat.

Les réseaux de neurones sont des approximateurs parcimonieux universels, ils sont par conséquent d'excellents outils pour la résolution des problèmes de modélisation et de classification, ils traitent facilement les données réelles et les algorithmes sont robustes au bruit. Cependant, ces propriétés d'approximateurs, ont une portée limitée en raison de l'indétermination qui subsiste sur le nombre de couches et le nombre de neurones par couche. Cette indétermination est d'autant plus grave qu'une sur-paramétrisation entraîne le phénomène de sur-apprentissage. L'idée est de concevoir des réseaux à architecture évolutive, en ajustant pendant l'apprentissage la taille du réseau afin que sa complexité soit adaptée au problème à résoudre. L'idée de l'élagage de réseau (Herault, 1994), connue aussi dans les graphes et les arbres, s'avère très efficace, quoique parfois coûteuse en temps de calcul. Il existe deux approches : La première est l'application de ces principes aux perceptrons multicouches. La seconde repose sur un autre modèle de neurone, le neurone à noyau ou à Fonctions Radiales de Bases (RBF) (Antoine Cornuéjols, 2011), (Lereno, 2000).

1.5.6 Les arbres de décision

Un arbre de décision (Quilan, 1986), (Quinlan, 1983) est la représentation graphique d'une procédure de classification. Il permet de modéliser simplement, graphiquement et rapidement un phénomène mesuré plus ou moins complexe. Pour certains domaines d'application, il est essentiel de produire des procédures de classification compréhensibles par l'utilisateur. C'est en particulier le cas pour l'aide au diagnostic médical où le médecin doit pouvoir interpréter les raisons du diagnostic.

La figure (1-9) illustre un exemple simple d'arbre de décision qui est présenté dans l'ouvrage de Quinlan (Quinlan, 1983). Ici on cherche à classer une population d'individus en deux classes par rapport à un jeu {jouer, ne pas jouer} à partir des prévisions météorologiques (Tableau 1-1).

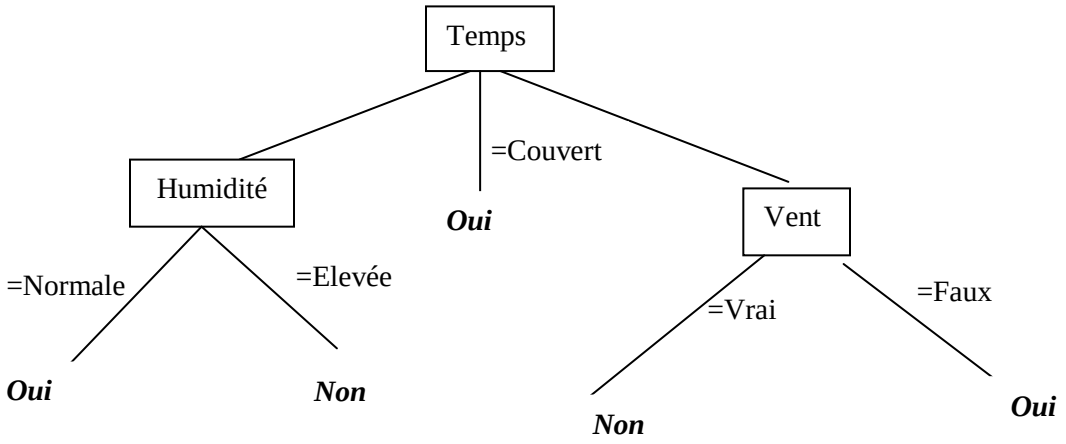

Figure 1-9 : *Exemple d'arbre de décision sur les données "weather"*

Un arbre de décision est constitué d'un ensemble de règles permettant de segmenter (diviser, partitionner) un ensemble de données en groupes homogènes. Chaque règle associe une conjonction de tests sur les variables descriptives. Le premier sommet est appelé la « racine » de l'arbre, les variables suivantes qui correspondent aux nœuds non terminaux (tests sur les attributs) sont des variables de segmentation, chaque branche (arête, arc) correspond à une modalité de la variable (réponse à un test) considérée à ce niveau de

l'arbre. Lorsque les tests sont binaires, l'une des branches correspond à une réponse positive au test et l'autre branche à une réponse négative, les feuilles représentent les classes. Ce processus est réitéré sur chaque nœud de l'arbre, les nœuds qui ne sont pas purs sont segmentés jusqu'à l'obtention de feuilles pures. Un exemple ne peut être situé dans deux feuilles différentes de l'arbre.

Tableau 1-1. Données "weather" (Quinlan, 1993)

Temps	Température	Humidité	Vent	Tennis ?
Ensoleillé	Chaude	Élevée	FAUX	Non
Ensoleillé	Chaude	Élevée	VRAI	Non
Couvert	Chaude	Élevée	FAUX	Oui
Pluvieux	Modérée	Élevée	FAUX	Oui
Pluvieux	Fraîche	Normale	FAUX	Oui
Pluvieux	Fraîche	Normale	VRAI	Non
Couvert	Fraîche	Normale	VRAI	Oui
Ensoleillé	Modérée	Élevée	FAUX	Non
Ensoleillé	Fraîche	Normale	FAUX	Oui
Pluvieux	Modérée	Normale	FAUX	Oui
Ensoleillé	Modérée	Normale	VRAI	Oui

Temps {ensoleillé, couvert, pluvieux}
Température {chaud, modéré, frais}
Humidité {élevée, normale}
Vent {VRAI, FAUX}

Pour classer un exemple, il suffit de descendre dans l'arbre selon les réponses aux différents tests pour l'exemple considéré. La classe attribuée est alors celle par défaut associée à la feuille qui correspond à la description. La procédure de classification associée est compréhensible par tout utilisateur, les attributs apparaissant dans l'arbre sont tous présélectionnés suivant leur pertinence par rapport au problème de classification considéré suivant des mesures de pertinence que l'on va présenter par la suite. On peut également remarquer qu'un arbre de décision a une traduction immédiate en règles de décision.

1.5.6.1 Apprentissage des arbres de décision

Étant donné un échantillon observé Ω composé de *n* objets ou individus ou instances $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$, un ensemble de classes $Y_{but} = \{m_1^{but}, ..., m_m^{but}\}$ et un arbre de décision *T*. Chaque élément de Ω est caractérisé par un ensemble de *r* variables ou attributs : $L = \{y_1, ..., y_k, \ldots, y_r\}$, $O_k = \{m_1^k, ..., m_v^k, \ldots, m_{m_k}^k\}$ est un ensemble de m_k modalités (valeurs) de la variable y_k où m_v^k est la modalité *v* de la variable y_k.

A chaque position (nœud) *p* de *T* correspond un sous-ensemble de l'échantillon qui est l'ensemble des exemples qui satisfont les tests de la racine jusqu'à cette position. La plupart des algorithmes procèdent de façon descendante, en partant de l'échantillon des données d'apprentissage toutes classes confondues. La construction de l'arbre se poursuit tant que, les exemples issus du partitionnement n'appartiennent pas à la même classe ou bien qu'il reste au moins un attribut à tester. A la fin, on obtient un arbre de nœuds (tests) dont les feuilles correspondent à des ensembles d'exemples aussi « pures » (idéalement appartenant tous à la même classe) que possible.

Au cours de la construction de l'arbre, le test mis en place à chaque nœud est basé sur l'examen de la meilleure façon de séparer en classes distinctes le sous-ensemble des exemples d'apprentissage considérés.

Pour chaque variable candidate, l'ensemble des observations est partitionné et afin d'évaluer la pertinence de la variable dans la segmentation, une mesure de qualité est calculée ; la variable retenue sera alors celle qui optimise cette mesure. Les méthodes diffèrent selon la mesure utilisée (Lereno, 2000).

1.5.6.2 Critère de sélection d'un attribut pour la segmentation

La théorie de l'information permet de déterminer l'homogénéité entre deux distributions de probabilités en utilisant la mesure de l'*information mutuelle,* ou *entropie croisée* issue de l*'entropie de Shannon*. Cette dernière quantifie l'information apportée par un événement

sur un système (Shanon, 1948). Si l'on considère la variable aléatoire y_k et O_y l'ensemble des modalités de y_k. La quantité d'information apportée par cette variable est définie par :

$$H\left(y_k\right) = -\sum_{v=1}^{m_k} P\left(y_k \left| m_v^k \right.\right) \log_2 P\left(y_k \left| m_v^k \right.\right) \tag{1.28}$$

$$H\left(y_k \left| y_{but} \right.\right) = -\sum_{v=1}^{m_k} \sum_{u=1}^{m} P\left(y_k \left| m_v^k, y_{but} \right| m_u^{but}\right) \log_2 P\left(y_k \left| m_v^k, y_{but} \right| m_u^{but}\right) \tag{1.29}$$

Le gain d'information utilisé par (Quinlan, 1979) appliqué dans l'algorithme d'arbre de décision ID3 mesure l'interaction entre une variable descriptive et une classe dans la formation d'arbre de décision. Il est donné par :

$$H\left(y_k, y_{but}\right) = \sum_{v=1}^{m_k} \sum_{u=1}^{m} P\left(y_k \left| m_v^k, y_{but} \right| m_u^{but}\right) \log_2 \frac{P\left(y_k \left| m_v^k, y_{but} \right| m_u^{but}\right)}{P\left(y_k \left| m_v^k \right.\right) P\left(y_{but} \left| m_u^{but} \right.\right)} \tag{1.30}$$

$$H\left(y_k, y_{but}\right) = H\left(y_k\right) - H\left(y_k \left| y_{but} \right.\right) \tag{1.31}$$

Pour construire un nœud dans un arbre, il suffit de chercher parmi les *r* attributs celui qui possède la plus grande corrélation avec la répartition en classe, autrement dit celui qui a la meilleure information mutuelle ou bien celui qui donne le meilleur gain d'information.

$$Gain\left(y_k\right) = H_{y_{but}} - H_{y_{but}|y_k} \tag{1.32}$$

L'algorithme générique peut s'écrire :

Algorithme d'apprentissage d'arbre de décision

Données : un échantillon Ω de m enregistrements étiquetés
Initialisation : arbre vide ; nœud courant : racine ; échantillon courant : Ω
répéter
décider si le nœud courant est terminal
si le nœud courant est terminal **alors**
étiqueter le nœud courant par une feuille
sinon
sélectionner un test et créer le sous-arbre
finsi
nœud courant : un nœud non encore étudié
échantillon courant : échantillon atteignant le nœud courant
Jusqu'à production d'un arbre de décision
sortie : arbre de décision

La construction des arbres de décision nécessite :

1. Un choix adéquat de la variable de segmentation sur un sommet ou un nœud. Par exemple, pourquoi avons-nous choisi la variable « Temps » à la racine de l'arbre ? Il nous faut donc une mesure qui permet d'évaluer les descripteurs et ainsi de sélectionner le meilleur parmi les candidats à la segmentation sur un sommet.

2. Critères d'arrêt de l'algorithme d'apprentissage :

   - prédéfinir un seuil de la proportion d'exemples d'une classe dans un nœud afin d'éviter le sur-apprentissage;
   - fixer un seuil d'entropie en dessous duquel on refuse d'éclater un sommet;
   - arriver à un nœud pur.

3. La règle de décision optimale lorsqu'une feuille n'est pas pure : généralement, on étiquette le nœud courant par la classe majoritaire. Cependant, pour certains problèmes, il se peut que les erreurs de classification d'une classe aient des conséquences différentes. Dans ce cas, il est possible de définir des coûts de mauvaise classification et la classe choisie le sera en fonction des coûts attribués.

1.5.6.3 Elaguer l'arbre de décision obtenu

Il est possible de poursuivre la croissance de l'arbre jusqu'à obtention d'un arbre d'erreur nulle si il n y a pas d'inconsistance dans les données (exemples ayant la même description mais des classes différentes), ou d'un arbre ayant une erreur sur l'ensemble d'apprentissage la plus petite possible. Cependant, l'objectif d'une procédure de classification est de bien classer des exemples non encore rencontrés, on parle de pouvoir de généralisation. Si l'algorithme fournit en sortie un arbre très grand qui classe bien l'échantillon d'apprentissage, on se trouve confronté au problème de sur-apprentissage : on a appris *par cœur* l'ensemble d'apprentissage, mais on n'est pas capable de généraliser. L'objectif de la phase d'élagage est d'obtenir un arbre plus petit (on élague des branches, c'est-à-dire que l'on détruit des sous-arbres) dans le but d'obtenir un arbre ayant un meilleur pouvoir de généralisation (même si on fait augmenter l'erreur sur l'ensemble d'apprentissage) (Rakotomalala, 2005).

Afin d'obtenir une meilleure généralisation et ainsi contrebalancer le sur-apprentissage, la façon la plus efficace est généralement d'utiliser une partie A de l'ensemble d'apprentissage pour construire un arbre T_{max} dont toutes les feuilles sont aussi pures que possible ; ensuite élaguer l'arbre avec un ensemble de validation V de données. Le reste des données soit l'ensemble T test, servira à évaluer le risque (erreur réelle) de l'arbre construit.

La très grande majorité des méthodes recensées à ce jour respectent ce schéma, il est alors facile de les positionner les unes par rapport aux autres (Lereno, 2000), (Michaut, 1999). Nous préférons dans nos travaux mettre l'accent sur un algorithme didacticiel très largement utilisé : l'algorithme C4.5 développé par Quinlan (Quinlan, 1983). C4.5 permet la génération d'un arbre de décision, élagué ou non. Cette méthode a pour but global de

générer un arbre de décision petit et simple capable de classifier les nouvelles instances. L'algorithme C4.5 se base sur la mesure de l'entropie dans l'échantillon d'apprentissage. L'algorithme travaille sur des données symboliques que ce soient des variables catégorielles ou numériques discrètes. Les variables continues doivent être discrétisées avant la mise en œuvre de l'algorithme pour préserver l'efficacité de l'apprentissage et la pertinence du modèle produit. Nous décrivons son fonctionnement de manière précise en Annexe A.

Un arbre de décision est facile à interpréter car il est la représentation graphique d'un ensemble de règles. Si la taille de l'arbre est importante, il est difficile d'appréhender l'arbre dans sa globalité. Cependant, les outils actuels permettent une navigation aisée dans l'arbre comme parcourir une branche, développer un nœud, élaguer une branche (Witten & Frank, 2000). L'arbre ne contient que les attributs jugés utiles pour la classification. L'algorithme peut donc être utilisé comme prétraitement qui permet de sélectionner l'ensemble des variables pertinentes pour ensuite appliquer une autre méthode (Lereno, 2000), (Sugumaran, et al., 2007). L'attribution d'une classe à un exemple à l'aide d'un arbre de décision est un processus très efficace (parcours d'un chemin dans un arbre). L'inconvénient majeur des arbres de décision est leur sensibilité au nombre de classes. Les performances tendent à se dégrader lorsque le nombre de classes devient trop important.

1.6 ECD en surveillance et diagnostic

Pour la mise en place d'un système de surveillance par les outils d'ECD, l'expert est censé connaître les modes de bon fonctionnement et certains modes de défaillances. Une grande partie des modes de bon fonctionnement est généralement fournie par les données du constructeur de l'équipement. Par contre, les informations concernant les modes de défaillance peuvent provenir de deux origines différentes : soit fournies par le constructeur ou par le bureau des études (provenance de haut), soit collectées en cours de fonctionnement de l'équipement (provenance de bas). Ces connaissances sont emmagasinées dans un historique de fonctionnement (base de données). Celui-ci contient les différentes relations de "causes à effets" des situations de dysfonctionnement d'un équipement.

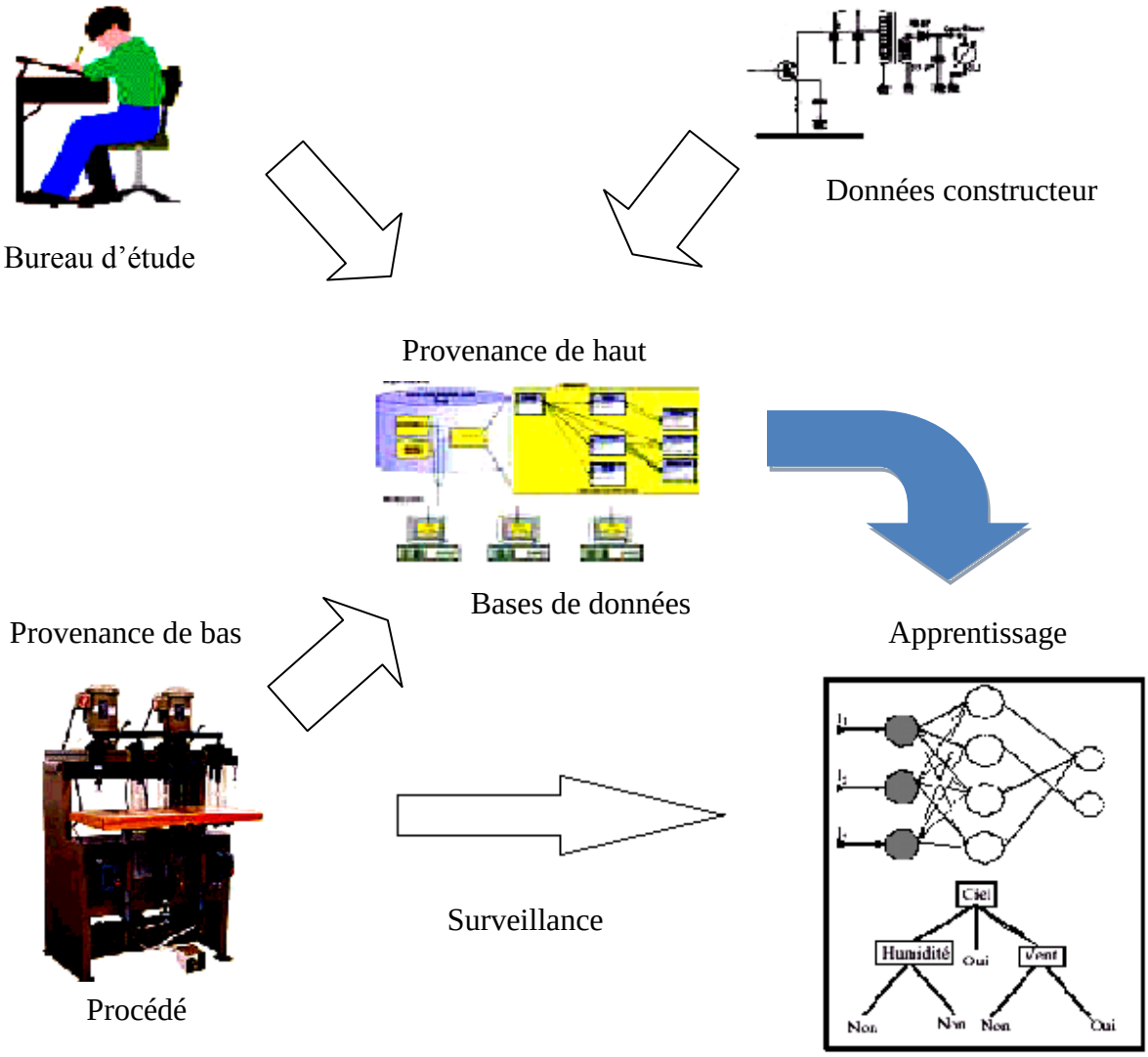

Figure 1-10 : *Sources des informations d'apprentissage*

L'opération de diagnostic menée par l'expert est souvent très complexe et demande des connaissances ainsi qu'un raisonnement, généralement difficiles à formaliser. Cependant, avec les outils de fouille de données on peut formaliser ces informations contenues dans l'historique de fonctionnement, qui représentent alors la base d'apprentissage pour

l'algorithme de fouille de données. Les variables d'entrée peuvent être constituées par les différents paramètres mesurés sur le procédé. On cherche alors, à associer un mode de fonctionnement (fonctionnement nominal, dégradé, défaut, …) à ces variables d'entrée. Les variables de sortie sont des variables catégorielles où chaque catégorie représente un mode de fonctionnement. Les outils de fouille de données permettent de définir la relation entrées-sorties qui représentent dans ce cas directement l'opération de diagnostic. Le plus important et que la réussite de cette application est tributaire de la qualité des informations contenues dans l'historique de fonctionnement.

1.7 Conclusion

Nous avons abordé dans ce chapitre le processus ECD. Au sein de ce processus, nous avons d'abord présenté les aspects généraux de la sélection de variables puis les techniques d'apprentissage que nous avons utilisées dans le cadre de nos travaux à savoir, les k-plus proches voisins, les réseaux de neurones et les arbres de décision.

Comme il a été mentionné tout au long de ce chapitre, l'un des éléments à ne pas négliger durant la phase d'apprentissage est la pertinence des données disponible en entrée. C'est pourquoi nous avons mis au point un algorithme de filtrage de variables. Il sera intéressant de voir l'impact de la sélection de données sur les performances des algorithmes d'apprentissage utilisés.

Chapitre 2
Diagnostic industriel

Sommaire

2.1 Introduction

Le diagnostic industriel dont la vocation première est de détecter, identifier et localiser les défaillances des systèmes industriels, joue un rôle primordial pour contribuer, par détection rapide ou précoce, à faire gagner des points de disponibilité et de productivité des capitaux investis dans l'outil de production par la réduction des coûts directs (relatifs aux diverses pièces de rechange, main d'œuvre, etc.) et indirects (engendré par un arrêt de production) de la maintenance des équipements de production.

Les méthodes de diagnostic peuvent être divisées en deux grandes catégories : les méthodes qui se basent sur l'existence d'un modèle formel de l'équipement à surveiller (méthode internes), et les méthodes qui se basent uniquement sur l'analyse des variables (données ou paramètres) de surveillance ainsi que sur les connaissances à priori des experts humains (méthode externes).

Les méthodes qui se basent sur une modélisation de l'équipement sont dépendantes d'une modélisation physique de l'équipement. Le modèle servira de référence pour un fonctionnement nominal et tout écart par rapport au point de fonctionnement nominal sera synonyme de défaillance. Cependant, quand aucun modèle n'est disponible, on peut alors mettre en œuvre des techniques externes qui utilisent une base de données issue de la collecte de mesures sur le processus étudié. La seule connaissance repose alors sur l'expertise humaine confortée par un solide retour d'expérience basé sur l'étude de cas réels (Zwingelstein, 1995), (Dubuisson, 1990), (Dubuisson, et al., 2001).

2.2 Notions de base en diagnostic industriel

Afin d'introduire le lecteur à nos travaux de recherche, nous allons commencer par présenter des généralités sur le domaine de la maintenance et de son évolution ainsi que les techniques les plus courantes en diagnostic d'équipements industriels, pour focaliser par la suite nos propos sur la partie « intelligente » de la maintenance, notamment celle qui utilise les techniques d'ECD et d'Intelligence Artificielle (IA) et qui constitue le cœur de nos travaux.

2.2.1 Définition des termes de base utilisés en diagnostic

Les organismes de normalisation internationaux AFNOR et CEI (AFNOR, 2001) ont défini avec précision les vocabulaires à utiliser dans les différents secteurs industriels.

Dégradation

Une dégradation représente une perte de performances d'une des fonctions assurées par un équipement.

Une défaillance

L'altération ou la cessation de l'aptitude d'un ensemble à accomplir sa ou ses fonction(s) requise(s) avec les performances définies dans les spécifications techniques.

Panne

Une panne est l'inaptitude d'une entité (composant ou système) à assurer une fonction requise.

Un défaut

Un défaut est une anomalie de comportement au sein du système. Ce concept est important dans les opérations de surveillance pour la conduite et la maintenance des processus industriels.

Tout écart entre la caractéristique observée et la caractéristique de référence est considéré comme étant un défaut. Il est donc clair qu'une défaillance conduit à un défaut. Mais un défaut n'induit pas nécessairement une défaillance. En Effet, le dispositif peut conserver son aptitude à accomplir sa tâche principale si les défauts n'ont pas d'impacts sur cette tâche. L'art du diagnostic consiste à détecter de façon précoce un défaut avant qu'il ne conduise à un état de défaillance donc de panne.

Surveillance

La surveillance est un dispositif passif, informationnel qui analyse l'état du système et fournit des indicateurs. La surveillance consiste notamment à détecter et classer les

défaillances en observant l'évolution du système puis à les diagnostiquer en localisant les éléments défaillants et en identifiant les causes premières. La surveillance se compose donc de deux fonctions principales qui sont la **détection** et le **diagnostic**.

La détection

C'est l'étape qui décide si le système est soumis à un défaut ou pas.

La localisation

Cette étape permet de localiser le défaut et donc de déterminer quelle partie du système est affectée par l'anomalie.

Le diagnostic

Le diagnostic se décompose en deux fonctions la localisation et l'identification. La localisation permet de déterminer le sous-ensemble fonctionnel défaillant tandis que l'identification de la cause consiste à déterminer les causes qui ont mené à une situation anormale.

La maintenance

La maintenance est l'ensemble de toutes les actions techniques, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise.

2.2.2 Historique et évolution de la maintenance

L'apparition du terme "maintenance" dans l'industrie a eu lieu vers 1950 aux USA, il se superpose progressivement à l'entretien.

Années 60 : Maintenance réactive / corrective

Intervient après la détection et la localisation d'un défaut.

Années 70 : Maintenance préventive (préventive systématique)

Effectuée dans l'intention de réduire la probabilité de défaillance d'un bien ou la dégradation d'un service rendu. C'est une intervention de maintenance prévue,

préparée et programmée avant la date probable d'apparition d'une défaillance. La maintenance systématique : effectuée selon un échéancier établi suivant le temps ou le nombre d'unités d'usage (Héng, 2005).

Années 80: Maintenance prédictive (prévisionnelle, préventive conditionnelle)

Basée sur la surveillance en continu de l'évolution du système, afin de prévenir un dysfonctionnement avant qu'il n'arrive sans prendre en compte la loi de dégradation. La décision d'intervention préventive est prise lorsqu'il y a un défaut imminent survient, ou approche d'un seuil de dégradation prédéterminé.

Années 90 : Maintenance proactive

Implique la surveillance et la vérification continues des causes primaires de défaillance du système surveillé.

Années 2000 : La Télémaintenance et E-maintenance

Le développement actuel des technologies de l'information et de la communication et la distribution de l'intelligence aux niveaux les plus bas, permet de passer de la maintenance classique à la maintenance à distance et en temps réel la « Télémaintenance » et la « E-maintenance » (Zemouri, 2003).

- *La Télémaintenance* est un concept de récupération de données à distance ou de prise de contrôle et de décision à distance.
- *La e-Maintenance* est un concept lié au principe de *web-services,* elle intègre le principe de base de la Télémaintenance en lui associant une dimension forte, constituée par la *coopération* et le *partage des connaissances* au niveau des informations mais aussi des hommes, des services (ingénierie, exploitation, maintenance, sûreté, achats, comptabilité, …) et des sociétés (client / fournisseur, inter fournisseurs, inter clients, …) (Racoceanu, 2006).

2.3 Organisation générale de la procédure de diagnostic

La sélection de la méthode de diagnostic la plus appropriée à un système industriel donné ne peut se faire qu'après un recensement des besoins et de connaissances disponibles sur ce dernier.

Ce paragraphe a pour objectif de structurer la démarche pour retenir la méthode techniquement et économiquement la plus efficace. La procédure de diagnostic de défaillances et de dégradations susceptibles d'affecter les différentes entités d'un processus industriel s'articule autour des étapes suivantes :

- L'extraction des informations nécessaires à la mise en forme des caractéristiques associées aux fonctionnements normaux et anormaux, à partir de moyens de mesures appropriées ou d'observations réalisées lors des rondes par les personnels de surveillance.
- L'élaboration des caractéristiques et signatures associées à des symptômes révélateurs de défaillances et de dégradations en vue de la détection d'un dysfonctionnement.
- La détection d'un dysfonctionnement par comparaison avec des signatures associées à des états de fonctionnements normaux et anormaux et la définition d'indicateurs de confiance dans la détection.
- La mise en œuvre d'une méthode de diagnostic de la défaillance ou de la dégradation à partir de l'utilisation des connaissances sur les relations de cause à effet.
- La prise de décision en fonction des conséquences futures des défaillances et des dégradations. Cette prise de décision peut conduire à un arrêt de l'installation si les conséquences de la défaillance sont importantes pour la sécurité des personnes et des biens ou à une reconfiguration du fonctionnement du procédé pour éviter une perte de production en attendant le prochain arrêt de production le plus proche.

2.4 Classification des méthodes du diagnostic industriel

Les méthodes de diagnostic se classifient en deux grandes familles : Les méthodes avec modèle (internes) et les méthodes sans modèle (externes).

2.4.1 Méthodes de diagnostic avec modèle (internes)

Ces méthodes se basent sur l'existence d'un modèle formel de l'équipement et utilisent généralement des techniques de *l'Automatique*. Elles impliquent une connaissance approfondie du fonctionnement du système sous la forme de modèle mathématique, les méthodes de diagnostic internes se regroupent en deux grandes familles :

2.4.1.1 Méthodes de diagnostic de défaillances par modélisations fonctionnelles et matérielles

Cette catégorie de méthodes comprend des outils industriels comme l'AMDEC (Analyse des Modes de Défaillance, de leur Effet et de leur Criticité) et l'Arbre De Défaillances ADD. Elles sont basées sur la décomposition d'un système en éléments matériels ou fonctionnels et la représentation graphique de la structure fonctionnelle du système par exemple à l'aide de l'arbre fonctionnel. Ces méthodes nécessitent l'identification à priori des défauts et/ou dysfonctionnements ainsi que leurs relations éventuelles. Or ce recensement, requiert une longue expérience dont la durée d'acquisition peut excéder le cycle de vie du système (Dubuisson, et al., 2001).

2.4.1.2 Méthodes de diagnostic à base de modèle physique

Les méthodes de diagnostic par modélisation physique consistent à comparer les grandeurs déduites d'un *modèle représentatif du bon comportement* des différentes entités du processus avec les mesures directement observées sur le processus industriel. Tout écart est alors synonyme d'un dysfonctionnement, c'est-à-dire la présence d'un ou plusieurs défauts. Ce type de raisonnement par l'absurde, ou un défaut est par définition n'importe quoi d'autre que le comportement attendu et qui n'est pas nécessairement recensé à priori,

s'avère une méthode de raisonnement très puissante, cette dernière pallie aux limites des approches traditionnelles. Cependant, ces modèles sont difficiles à mettre en œuvre pour les équipements complexes, et quand ces modèles existent, leur réponse est souvent entachée d'incertitudes de modélisation ce qui pose des problèmes dans la démarche de diagnostic pour la prise de décision. Ces incertitudes sont dues au fait qu'on ne peut pas prendre en considération tous les paramètres physiques d'un équipement ainsi que les bruits de mesures. Le terme « modèles » ici est employé par opposition aux connaissances issues d'un raisonnement plus profond en IA (Dubuisson, et al., 2001).

2.4.2 Méthodes de diagnostic sans modèle (externes)

Nombreuses sont les applications industrielles dont le modèle est difficile, voire impossible à obtenir suite à une complexité accrue ou à de nombreuses reconfigurations intervenants durant le processus de production. Pour ce type d'applications industrielles, les seules méthodes de diagnostic opérationnelles sont celles sans modèle (Dubuisson, et al., 2001). Celles-ci représentent les outils statistiques de *Traitement du Signal et* les techniques de *l'intelligence Artificielle* et servent comme outil de base pour l'aide à la décision dans ce cas.

1. **Les outils statistiques de *Traitement du Signal*** sont généralement qualifiés d'outils de traitement de bas niveau, parce qu'ils sont en contact direct avec le signal capteur, en effectuant des tests sur les signaux d'acquisition, ces tests ne sont capables d'assurer que la fonction détection de défaillances en générant des alarmes brutes, sans aucune information concernant leur signification (Zemouri, 2003).

2. **Les techniques de *l'Intelligence Artificielle* (IA)** dites de haut niveau sont plutôt orientées vers la communication avec l'expert. Leur réponse est plus élaborée que celle des techniques de bas niveau et elles sont capables de détecter, interpréter (association à un mode) et diagnostiquer les défaillances. l'intelligence artificielle permet de pallier la complexité des systèmes à diagnostiquer. Elle est

> relativement bien adaptée aux problèmes du diagnostic par sa capacité à traiter de grandes quantités d'informations, des données aussi bien numériques que symboliques, des données dépendant du contexte ainsi que les données incomplètes (Dubuisson, et al., 2001).

Dans nos recherches, nous abordons le problème de diagnostic de défaillances en vue de la maintenance avec un aspect utilisant les techniques de *l'Intelligence Artificielle* et plus précisément les techniques d'ECD qui utilisent les techniques de l'IA et de reconnaissance de formes.

2.4.2.1 Techniques de l'intelligence artificielle

Globalement, ces méthodes sont classées en trois groupes (Racoceanu, 2006) : les méthodes à base de modèles comportementaux, les méthodes de reconnaissance de formes et les méthodes à base de modèles explicatifs.

2.4.2.2 Les méthodes à base de modèles comportementaux

Les méthodes à base de modèles comportementaux ou bien modèles de dysfonctionnement se caractérisent par la possibilité de simuler le comportement du système notamment ces modes de dysfonctionnement. Le modèle ici aide à prédire le comportement du système observé. Un problème est relevé, lorsque ce qui est prédit par le modèle et ce qui est observé est incompatible. Ce type de connaissances, peuvent non seulement aider à détecter les défauts, mais aussi à identifier la panne responsable du dysfonctionnement. Les chercheurs s'intéressent particulièrement aux systèmes à événements discrets en utilisant des outils tels que les réseaux de Pétri et les automates d'états finis (Racoceanu, 2006).

2.4.2.3 Les méthodes à base de modèles explicatifs

Ces méthodes cherchent les causes qui expliquent les symptômes par une représentation causale des liens entre les défaillances, leurs causes et leurs effets observables tels que les réseaux Bayésiens.

2.4.2.4 Les méthodes de reconnaissance de formes

Les formes représentent le vecteur d'entrée composé par les différentes données de l'équipement (données mesurables et qualifiables) et les classes correspondent aux différents modes de fonctionnement (Dubuisson, 1990). On distingue les méthodes symboliques (la procédure de classification produite peut être écrite sous forme de règles) et les méthodes non symboliques ou adaptatives (la procédure de classification produite est de type boîte noire). Dans ces méthodes, on retrouve principalement, les réseaux neuronaux, la logique floue, les réseaux neuro-flous, les systèmes expert et le raisonnement à partir de cas.

2.4.2.4.1 Système expert

Un système expert, est un outil informatique d'intelligence artificielle, conçu pour simuler le savoir-faire d'un spécialiste, dans un domaine précis et bien délimité, grâce à l'exploitation d'un certain nombre de connaissances fournies explicitement par des experts du domaine. Il est composé de deux parties : une *base de connaissances* représentant à la fois du savoir faire et l'expertise nécessaires pour résoudre un problème, elle-même composée d'une base de règles qui modélise la connaissance du domaine considéré et d'une base de faits et d'un *moteur d'inférences* capable de raisonner à partir des informations contenues dans la base de connaissance, de faire des déductions, etc.

2.4.2.4.2 Outils statistiques de reconnaissance de formes

Les méthodes statistiques se décomposent en méthodes paramétrique et non paramétrique. Le cas Bayésien correspond à des critères de décision construits à partir des lois de probabilité. Le cas paramétrique correspond à une connaissance a priori sur les lois de probabilité dont il faut estimer les paramètres. La variété des méthodes viendra de la diversité des hypothèses possibles. Cependant ces méthodes supposent une connaissance *à priori* de tous les états de fonctionnement et ne prennent pas en compte l'évolution du système. Les méthodes non paramétriques (sans hypothèse à priori sur les distributions de probabilité) ont été également proposées en statistiques, la plupart de ces méthodes sont

issues de l'intelligence artificielle. On distingue les méthodes symboliques où la procédure de classification produite peut être écrite sous forme de règles (tel que les arbres de décision) et des méthodes non symboliques dites de type « boîte noire » tel que les réseaux de neurones.

2.4.2.4.3 Raisonnement à partir de cas

Le raisonnement à partir de cas (RàPC), Case Based Reasoning (CBR) en anglais est une approche récente pour résoudre et apprendre des problèmes et leurs solutions.

Le RàPC est une technique pour résoudre des problèmes basés sur l'expérience, et donc relativement bien adapté aux problèmes de diagnostic pour lesquels la notion d'expérience est relativement importante. Il correspond à la résolution d'un nouveau problème en se remémorant une situation précédente similaire. Le principe de fonctionnement de la méthode consiste à stocker les cas (expériences) précédents dans une mémoire afin de résoudre un nouveau problème en réutilisant cette expérience dans le contexte de la nouvelle situation ou en l'adaptant selon les différences (Riverol & Carosi, 2008), (Haouchine, M, K. 2009). Le RàPC manipule une base de connaissance qui contient deux parties : *la connaissance générale,* souvent représentée par une base de règles et qui peut intervenir dans toutes les phases du RàPC et *la mémoire*, qui contient les cas qui représentent l'expérience du système.

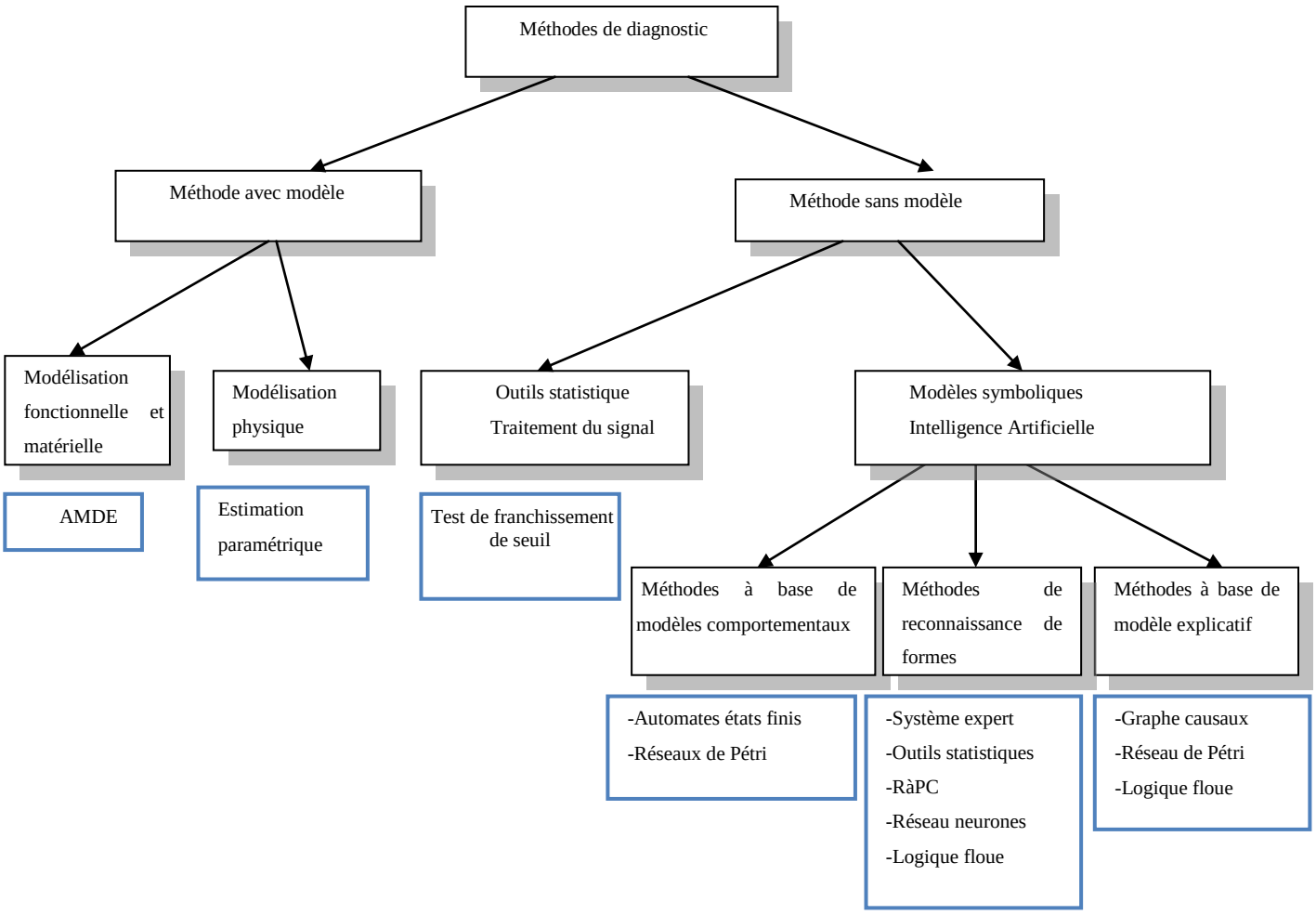

Figure 2-1 : *Classification globale des méthodes de diagnostic* (Racoceanu, 2006)

2.5 Position et apport de notre étude

Le rôle d'un système de diagnostic est d'identifier le défaut le plus probable qui a engendré l'apparition d'un symptôme. Dans le cadre de ce travail nous allons considérer un système de diagnostic comme un classificateur de données où les classes correspondent aux différents défauts.

La réussite de cette méthodologie est principalement tributaire de la qualité des informations contenues dans l'historique de fonctionnement. L'expert est censé connaître les modes de bon fonctionnement et certains modes de défaillances. Une grande partie des

modes de bon fonctionnement est généralement fournie par les données du constructeur de l'équipement. Par contre, les informations concernant les modes de défaillance peuvent provenir soit par le constructeur soit collectées en cours de fonctionnement de l'équipement. Ces informations sont emmagasinées dans un historique de fonctionnement dans des entrepôts de données (bases de données).

2.5.1 Etat de l'art sur la sélection de variables en diagnostic

Les méthodes élaborées en diagnostic se basent généralement sur les outils de traitement de signal ou bien sur des modèles physiques. Les techniques issues de l'apprentissage automatique ont aussi largement été utilisées. Nous allons dans ce qui suit citer brièvement quelques travaux en sélection de variables pour la détection de défauts dans ce contexte.

Dans (Paljak, et al., 2009), les auteurs ont sélectionné un ensemble compact mais suffisant de variables de contrôle comme paramètres d'estimation pour le contrôle prédictif et adaptatif destiné au diagnostic et la maintenance dans une large infrastructure de surveillance et de supervision. Pour cela ils ont utilisé l'algorithme de sélection de variable mRMR (*minimum Redundancy Maximum Relevance*) en combinaison avec une approche d'approximation linéaire, pour la sélection d'un ensemble réduit de caractéristiques le plus représentatif du système dans le cadre de l'instrumentation dédiée à la supervision.

Dans les travaux de Yang, et al. (Yang, et al., 2008), une large étude en diagnostic de défauts de roulements (*roller bearing*) dans les machines asynchrones est présentée utilisant les SVM (*Support Vector Machine*) machine à support de vecteur en combinaison avec d'autres méthodes de sélection de données.

Jack, et al. (Jack, et al., 2002) utilisent les algorithmes génétique comme outil pour la sélection d'un sous ensemble optimal de variables descriptives, ces dernières sont par la suite introduites comme entrées de deux algorithme d'induction : les réseaux de neurones et les SVM.

Travaillant sur les moteurs à induction Casimira et al. (Casimira, et al., 2006) ont fait une extraction de variables à partir d'une analyse temps fréquence sur le courant et la tension statorique afin d'obtenir un ensemble réduit de variables temps fréquence, cet ensemble est ensuite analysé avec un algorithme de sélection de variables de type séquentiel arrière (*sequential backward*) afin de sélectionner les variables temps fréquence les plus pertinentes. Les résultats obtenus en présélectionnant les variables ont nettement amélioré les performances de classification de l'algorithme d'induction utilisé : les k-plus proche voisin.

Les travaux de Sugumaran, et al. (Sugumaran, et al., 2007) concernent le diagnostic de défauts de roulement dans la machine tournante. Pour cela, ils analysent des signaux de vibration obtenus à partir d'un capteur piézoélectrique, ses signaux caractérisent différents modes de fonctionnement (pas de défaut, défaut de bague intérieure, défaut de bague extérieure et défaut de bague intérieure et extérieure). Dans un premier temps, un ensemble de 11 variables temps fréquence est extrait à partir d'une transformation temps fréquence de ses signaux. L'algorithme d'arbre de décision C4.5 est utilisé comme algorithme de sélection de variable afin de sélectionner 4 variables considérées les plus pertinentes. Dans un deuxième temps, l'algorithme d'induction *Proximal Support Vector Machine* (PSVM) est appliqué afin de classer les défauts.

Nous pouvons aussi citer les travaux de Torkola, et al. (Torkola, et al., 2004) spécialement dédiés à la sélection de capteurs car nous allons proposer dans le chapitre 4, la conception d'un système de détections de défaillances à partir d'une sélection de capteurs. Afin de concevoir un système qui permet des manœuvres automatiques d'une voiture, les auteurs (Torkola, et al., 2004) appliquent une sélection de variables qui a pour principal objectif d'identifier les capteurs qui sont nécessaires pour la classification de 12 manœuvres (tourner à gauche, à droite, en route,…). Des données capteurs tel que, accélération, vitesse, frein, etc ; sont collectées en utilisant un simulateur de conduite. Ensuite, un total de 138 variables est extrait à partir de ces données. Pour la sélection de variables, les auteurs combinent l'algorithme CFS (Hall, 1998) et les arbres de décision afin de réduire

l'ensemble de variables obtenu, sélectionner les plus pertinentes par rapport au concept étudié en plus de la sélection des meilleurs capteurs.

2.5.2 Méthodologie proposée pour le diagnostic

Comme nous avons vu dans ce chapitre, dans la plupart des modélisations des systèmes industriels, des incertitudes persistent entre le comportement du système réel et l'évolution du modèle. Ces incertitudes sont dues, d'un côté, aux manques de connaissances exhaustives sur le fonctionnement de l'équipement et, d'un autre côté, le modèle ne prend en compte qu'une partie des paramètres qui influent sur le système. Par ailleurs, dans certains cas, ce modèle est quasiment impossible à obtenir.

Prenant acte de ces diverses considérations nous proposons une démarche basée sur le développement des techniques *d'Extraction de Connaissance à partir des Données* pour le diagnostic de défauts, étape fondamentale de la fonction maintenance.

En effet, l'exploitation et l'exploration des bases de données pour en dériver de *la connaissance,* nous permet de proposer rapidement une bonne solution ou bien de converger plus rapidement vers une solution satisfaisante. Dans le cadre de ce travail, nous allons essayer de combiner les méthodes de sélection de variables avec des méthodes d'apprentissage supervisée afin d'améliorer les performances de classification de ces derniers et ainsi appliquer la combinaison de ces techniques en diagnostic.

2.5.2.1 Prétraitement des données

Pour le développement de tels outils plusieurs challenges doivent être relevés, principalement la dimensionnalité des données issues des entrepôts de données et leur hétérogénéité (symbolique et numérique). Dans ce travail, nous proposons quelques approches pour surmonter de manière appropriée de tels challenges. La première approche aborde le problème d'hétérogénéité par un algorithme de discrétisation des attributs continus. La deuxième approche permet de réduire la haute dimensionnalité des données en utilisant les algorithmes de sélection de variables, cette dernière se trouve au cœur de nos

travaux de recherches grâce à l'élaboration de l'algorithme de sélection de variables STRASS.

2.5.2.2 Méthodes d'induction

La description du contexte du concept (la classification) à fournir est réalisée initialement à partir d'une base d'apprentissage, les variables d'entrée peuvent être constituées par les différents paramètres mesurés sur le procédé. On cherche alors à associer un mode de fonctionnement (fonctionnement nominal, dégradé, …) à ces variables d'entrée. Les variables de sortie sont alors des variables classes où chaque classe représente un mode de fonctionnement.

En apprentissage supervisé, il existe plusieurs approches ayant la préférence des chercheurs du domaine et que nous avons jugé bon d'utiliser afin de valider nos travaux en sélection de variables; tel que : les réseaux de neurones, les arbres de décision et les et les k plus proches voisins. Le chapitre 1 situe notre étude par rapport à ces différentes alternatives.

2.6 Conclusion

Ce chapitre a eu pour objectif d'introduire les différents concepts relatifs au diagnostic et de présenter les différentes méthodes proposées dans ce domaine. Nous pouvons trouver dans la littérature plusieurs classifications de méthodes, nous avons distingué, essentiellement, entre les méthodes internes basées sur l'existence de modèles mathématiques et les méthodes externes basées sur les outils d'intelligence artificielle et les techniques de reconnaissance de formes. Nous avons aussi présenté la procédure de détection et identification des défauts (diagnostic) comme une procédure de classification. Cette procédure est englobée dans un processus plus grand qui est l'ECD. En effet, ces outils peuvent fournir une solution intéressante pour des problématiques de surveillance et de diagnostic industriel, car leurs utilisations ne nécessitent pas l'existence d'une modélisation formelle du système.

Chapitre 3
Algorithmes contextuels de sélection de variables

Sommaire

3.1 Introduction

Chaque observation est caractérisée par un ensemble de variables. Ces variables ne sont pas toute informatives. En effet, certaines d'entre elles peuvent être peu significatives, corrélées ou non pertinentes pour la tâche de fouille de données. La sélection des paramètres pertinents présente un intérêt majeur qui permet non seulement de réduire le volume de l'information à traiter et par conséquent de réduire le temps de calcul et la complexité des algorithmes de classification mais aussi d'améliorer les performances de généralisation de ces derniers. L'étape de filtrage de données indispensable dans ces conditions, a intéressé beaucoup d'auteurs (Fayyad, et al., 1990), (Almuallim, et al., 1991), (Almuallim, et al., 1994), (Blum, et al., 1997), (Dash, et al., 1997), (Hall, 2000), (Dash, et al., 2000), (Guyon, et al., 2003), (Zhao, et al., 2007), (Karegowda, et al., 2010), (Chandrashekar, et al., 2014). Cette étape a pour objectif de privilégier la qualité à la quantité en sélectionnant un sous ensemble de variables pertinentes pour expliquer la variable but ou variable classe.

Nous commençons donc, par donner les caractéristiques générales des algorithmes de sélection de variables. Ensuite, on décrira les algorithmes contextuels qui prennent en compte d'une manière plus fine que les autres méthodes le type de variables. En effet la

plus part des travaux en statistiques font l'hypothèse dans bien des cas erronées de l'indépendance entre les variables décrivant l'ensemble d'apprentissage.

3.2 Processus de sélection de variables

(Liu, et al., 1998) et (Liu, et al., 2005) ont comparé les différents travaux en sélection de données suivant quatre points :

- *La stratégie de recherche :*
    - **sélection avant FS** (*foward selection*) qui consiste, partant d'un ensemble vide, à ajouter un à un les attributs jusqu'à ce que tous les attributs aient été ajoutés. A chaque étape FS choisit l'attribut qui, ajouté aux attributs déjà sélectionnés, produit le meilleur sous-ensemble intermédiaire de variables selon un certain critère.
    - **élimination arrière BE** (*backward selection*), à l'inverse de FS, la constitution de l'ensemble d'attributs est réalisée par élimination des variables à partir de l'ensemble complet des attributs jusqu'à ce que l'ensemble ne contienne plus d'attributs.
    - ou **bidirectionnel**, qui est une combinaison des 2 stratégies précédentes.

- *La génération de la recherche* : la stratégie de recherche d'uns sous ensemble optimal de variables peut êtres soit complète, heuristique ou aléatoire.

- *Le critère d'évaluation :* c'est le point le plus important dans une procédure de sélection de variables, il permet d'évaluer l'aptitude du sous ensemble trouvé à la discrimination inter classes. Deux grandes méthodes de sélection sont associées à ce critère : Filtre et enveloppe.

    - Les méthodes filtres travaillent indépendamment de l'algorithme de fouille de données (la sélection se fait avant apprentissage).

- Les méthodes enveloppes qui utilisent la précision de l'algorithme d'induction comme critère d'évaluation. On retrouve aussi une nouvelle famille d'algorithmes de sélection de variables issue des deux familles : les méthodes hybrides.

- *Critère d'arrêt :* c'est un critère qui permet de stopper la recherche, ce dernier peut dépendre du critère d'évaluation ou être prédéfini par l'utilisateur.

Le premier type de méthode : les méthodes filtres sont indépendantes du processus d'induction et font donc l'objet de notre choix par rapport aux méthodes enveloppes. Ces dernières, basées sur des critères de précision présente un temps de calcul trop important.

3.2.1 Critères d'évaluation

Les critères d'évaluation peuvent êtres divisés en 5 groupes : les mesures d'informations, les mesures de distances, les mesures d'indépendance ou de dépendance (corrélation) et les mesures de précision.

3.2.1.1 Information

Le gain d'information (Shanon, 1948) est un critère largement utilisé en fouille de donnée, il est défini comme étant la différence entre l'incertitude à priori et l'incertitude à postériori. La sélection d'une variable est alors vue par rapport à la quantité d'information apporté au concept étudié. Si l'on considère deux variables aléatoires *A* et *B*, le gain d'information de la variable *A* par rapport à la variable *B* peut êtres défini comme suit :

$$\begin{aligned} IG(A|B) &= H(A) + H(B) - H(A,B) \\ H(A) &= -\sum_{a \in A} P(a) \log_2 P(a) \\ H(B) &= -\sum_{b \in B} P(b) \log_2 P(b) \end{aligned} \tag{3.1}$$

3.2.1.2 Distance

Ces mesures sont également connues sous le nom de séparabilité ou discrimination, elles évaluent la séparabilité des classes en se basant sur les distributions de probabilités des classes. Ainsi, si l'on considère un problème à deux classes et deux variables descriptives y_k, y_l : la variable y_k est jugée plus pertinente que la variable y_l si y_k induit une différence plus grande entre les probabilités conditionnelles des deux classes que y_l.

3.2.1.3 Dépendance

Egalement connues sous le nom de mesures de corrélation ou d'association. Elles estiment le degré avec lequel une variable est associée à une autre variable, par exemple la corrélation d'une variable descriptive par rapport à une variable classe. La plus connue est le chi2 χ^2, fonction statistique très répondu mise au point par Pearson (Stigler, 2008).

3.2.1.4 Consistance

Ces mesures cherchent à déterminer le sous ensemble de variables le plus petit qui satisfait un certain degré d'inconsistance dans les données (Dash, et al., 2000). Le degré d'inconsistance est calculé de la manière suivante :

Deux instances sont considérées inconsistantes si elles sont identiques et qu'elles appartiennent à des classes différentes. Les groupes constitués de ces instances sont successivement considérés et pour chaque groupe, on calcule la quantité d'inconsistance qui est égale à :

$$Ns\text{-}Nr \tag{3.2}$$

Avec *Ns* est le nombre d'instances semblables,

Nr est le nombre d'instances de la classe la plus représentée dans ce groupe.

Le pourcentage d'inconsistance est la somme de toutes les quantités d'inconsistance divisée par le nombre total d'instances.

3.3 Algorithmes de sélection de variables contextuels

Lorsque l'on dépasse une certaine dimension de l'ensemble d'apprentissage, les méthodes enveloppes se révèlent inefficaces et cela est dû essentiellement au temps de calcul. C'est la raison pour laquelle nous consacrons notre étude aux méthodes filtres, en nous intéressons plus particulièrement aux algorithmes de génération heuristique, permettant ainsi d'éviter toute explosion combinatoire. On peut classer les algorithmes filtres de sélection de variables suivant leurs critères d'évaluation en en deux grandes catégories :

1. Les algorithmes myopes de sélection de variables qui estiment la qualité d'une variable hors du contexte des autres.

2. Les algorithmes contextuels de sélection de variables qui prennent en compte les interactions (corrélations) entre variables.

Certains algorithmes sont qualifiés de contextuels mais d'une manière abusive, car ils n'utilisent pas des critères contextuels, puisqu'une variable est estimée en dehors du contexte des autres variables. C'est l'algorithme dans lequel s'intègre ces critères qui les rendent contextuels comme c'est le cas de l'algorithme référence de la communauté de sélection de variables, l'algorithme Relief (Kononenko, 1994).

Nous avons constaté que les algorithmes basés sur des critères myopes ne détectent pas les corrélations contrairement à ceux utilisant les critères contextuels. Cependant les algorithmes d'induction trouvent des difficultés à travailler avec des données corrélées. Hors la plus part des algorithmes existants font partie de la première catégorie.

Parmi les algorithmes contextuel de sélection de variables nous pouvons citer : Relief (Kononenko, 1994), (Kononenko, et al., 1997), CFS (Hall, 2000), mRMR (Peng, et al., 2005), FCBF (Yu, et al., 2004) et INTERACT (Zhao, et al., 2007). Ces algorithmes ont été reconnus comme étant les plus performants selon ce point de vue. Notre contribution se situant à ce niveau, il nous semble opportun de décrire plus en détail ces algorithmes.

3.3.1 Relief

Un algorithme qui donne une liste ordonnée de toutes les variables suivant leur pondération par un critère de distance. Un échantillon d'instances à partir de l'ensemble d'apprentissage est choisi, le nombre d'instances et le seuil de sélection de variable sont fixés par l'utilisateur. Il est considéré comme contextuel par (Kononenko, 1994), car il traite de façon appropriée les attributs fortement dépendants de la variable but. La meilleure variable est celle qui discrimine les plus proches voisins appartenant à des classes différentes et qui comporte des valeurs similaires pour ceux de mêmes classes. Le nombre de variables retenues dépend des seuils retenus par l'utilisateur.

Relief traite chaque variable indépendamment des autres variables, ceci peut le rendre moins performant devant les variables partiellement ou totalement corrélées. En effet, Kira et al. (Kira, et al., 1992) page 133) ainsi que Kohavi et al. ((Kohavi, et al., 1997) page 7) ont souligné l'incapacité de ce dernier à détecter les variables redondantes et les variables à pertinence faible notamment dans les bases de données réelles. Cependant Relief est un algorithme de référence dans la communauté de fouille de données, la raison pour laquelle nous avons jugé bon de l'utiliser à titre comparatif avec notre algorithme de sélection de variables.

3.3.2 CFS

CFS (Hall, 1998), (Hall, 2000) utilise le mérite contextuel comme critère d'évaluation d'une variable, ce dernier est calculé à partir de la corrélation entre variables. C'est le rapport entre la moyenne des corrélations de l'ensemble de variables à sélectionner et la variable but sur la moyenne des corrélations entre variables prises par paires.

Le mérite utilisé dans l'algorithme CFS est caractérisé par l'heuristique suivante :

$$Merit(S) = \frac{r\,\overline{Cy_{but}y_k}}{\sqrt{r + r(r-1)\,\overline{Cy_k y_l}}} \tag{3.3}$$

S : sous ensemble de variables.

$\overline{Cy_{but}y_k}$: moyenne de la corrélation entre les variables de S et la variable but.

$\overline{Cy_k y_l}$: moyenne de l'intercorrélation entre variables, les variables sont prises par paires.

Ce critère permet de déterminer les variables pertinentes en considérant celles qui sont corrélées à la variable but et faiblement corrélées aux autres variables, une variable est redondante quand elle est fortement inter-corrélée aux autres variables. L'algorithme est performant tant que l'interaction entre variables n'est pas trop grande.

3.3.3 mRMR

mRMR (*minimum Redondancy Maximun Relevance*) (Peng, et al., 2005) traite aussi de la dépendance entre variables et la variable but et la dépendance entre paires de variables. Ils ont utilisé une mesure se basant sur le gain d'information pour le calcul de la dépendance entre variables variable but (pour maximiser la pertinence) et la dépendance entre paires de variables (pour minimiser la redondance). En considérant *S* le sous ensemble de variables à sélectionner avec *m* variables y_i et la variable but y_{but}, la mesure utilisée par l'algorithme mRMR est calculée de la manière suivante :

$$\max \phi(D,R), \quad \phi = D - R \tag{3.4}$$

Tel que :

$$\max D(S, y_{but}), \quad D = \frac{1}{|S|}\sum_{y_k \in S} I(y_k, y_{but}) \tag{3.5}$$

et

$$\min R(S), \qquad R = \frac{1}{|S|^2}\sum_{y_k, y_l \in S} I(y_k, y_l) \tag{3.6}$$

Avec *D* la moyenne des corrélations entre variables et variable but et *R* la moyenne des corrélations entres paires de variables.

I représente l'information mutuelle de deux variables *a* et *b*. Elle est définie en se basant sur leur distribution de probabilité conjointe *P(a,b)* et leur probabilités marginales respectives *P(a)* et *P(b)* de la manière suivante:

$$I(A|B) = \sum_{a \in A, b \in B} P(a,b) \log \frac{P(a,b)}{P(a)P(b)} \tag{3.7}$$

Le sous ensemble optimal est celui qui maximise la distance entre les deux gains d'information.

3.3.4 FCBF

FCBF (*Fast Correlation based Feature Selection algorithm*) (Yu, et al., 2004) est un algorithme spécialement conçu pour palier aux problèmes engendrés par la corrélation des données dans les grandes bases de données. Pour cela FCBF utilise un critère se basant sur l'information apportée par une variable par rapport aux autres variables ce qui permet de détecter les variables redondantes. Ce critère est l'incertitude symétrique (SU : *symmetrical uncertainty*) défini par l'équation suivante :

$$SU(Y, y_{class}) = 2\left[\frac{IG(Y \mid y_{class})}{H(Y) + H(y_{class})}\right] \tag{3.8}$$

Sachant que le gain d'information (IG) est la différence entre l'incertitude à priori et l'incertitude à postériori en incluent une variable descriptive, SU vient compenser le biais induit par les variables ayant plus de modalités lorsque le gain d'information est utilisé seul. L'algorithme comporte deux phases : (1) calcul de SU pour chaque variable, la sélection des variables les plus pertinentes suivant l'ordre de pertinence des variables en considérant un seuil prédéfini et (2) sélection des variables prédominantes (à pertinence forte).

3.3.5 INERACT

INTERACT (*Interacting based Feature Selection algorithm*) (Zhao, et al., 2007) combine deux mesures pour la sélection des variables, mesure d'information et d'inconsistance. L'algorithme est en deux parties : dans la première, les variables sont ordonnées en se basant sur leur gain d'information. Dans la seconde partie de l'algorithme, les variables sont évaluées selon un critère noté C-contribution. C-contribution représente le taux d'inconsistance engendré par la suppression d'une variable.

3.4 Comparaison des algorithmes de filtrage contextuels

Les points forts de ces algorithmes concernent leur efficacité à traiter des problèmes bruités, modales et continues. Par contre dans le cas de Relief l'utilisateur doit fournir un seuil de pertinence tout en choisissant le bon échantillon de départ, ce qui rend l'algorithme ouvert mais cela nécessite un minimum de tâtonnement pour fixer ces seuils. CFS et mRMR calculent l'inter-corrélation entre paires de variables, ceci permet de déterminer les redondances entres deux variables (équivalence) et non pas la redondance d'une variable par rapport à un ensemble de variables. Ce type de redondance que nous avons qualifié de redondance partielle ou corrélation par morceaux, ne peut être traité que si on compare la contribution d'une variable par rapport à un ensemble de variables. Pour résoudre ce problème, nous allons présenter dans la section suivante l'algorithme STRASS, un algorithme que nous avons conçu à partir de deux critères d'évaluation et qui mettra en évidence la dépendance partielle entre variables et permettra ainsi d'affiner la sélection d'un sous ensemble minimum de variables pertinentes.

3.5 STRASS : Algorithme de sélection de variables à Pertinence Forte

Les mesures contextuelles étant plus performantes que les mesures myopes, nous avons donc élaboré une mesure contextuelle. Le pouvoir discriminant original (Vignes, et al., 1992), (Michaut, 1999), (Michaut et al., 1999) est le seul critère à notre connaissance qui

est contextuel mais qui travaille sur des paires de concepts. Cependant l'utilisation de ces critères pose problème quand à l'application de l'algorithme de filtrage dans les grandes bases de données. En effet, en fouille de données, l'approche par paires a bien longtemps été délaissée en raison de la combinatoire de ses calculs et on lui a préférée l'approche contingencielle. Ces arguments nous ont permis d'exploiter l'approche contingentielle pour reformuler les critères contextuels sous une forme contingentielle. Cette partie nous concernant de près nous donnerons l'expression de chaque critère par paires et son équivalent en forme contingentielle après utilisation des formules de passage « paires-contingence » de Marchotorchino (Marcotorchino, 1984).

Travaillant dans un contexte supervisé, et sur des objets représentés en attribut valeur, nous développons les critères que nous avons utilisé ci après que nous nommerons *pouvoir discriminant*, et *gain du pouvoir discriminant*.

3.5.1 Le pouvoir discriminant

Le pouvoir discriminant original (Vignes, et al., 1992) est un critère contextuel mais qui travaille sur des paires de concepts. Nous donnons au préalable quelques définitions issues du formalisme de Diday (Kodratoff, et al., 1991) nécessaires à la présentation de ce critère.

Soit un événement élémentaire $e_k = [y_k = V_k]$ où $V_k \subset O_k$

Un objet assertion est une conjonction d'évènements élémentaires :

$$a = [y_1 = V_1] \wedge \ldots\ldots \wedge [y_p = V_p] \tag{3.9}$$

Soit A l'ensemble des assertions, soit N l'ensemble des couples d'assertions $N=A \times A$

Soit la fonction :

$$comp(V_{ik}, V_{jk}) = \begin{cases} 1 & si \quad V_{ik} \cap V_{jk} = \phi \\ 0 & \text{sinon} \end{cases} \tag{3.10}$$

où V_{ik} , V_{jk} sont les valeurs prises par la variable y_k dans l'assertion a_i respectivement a_j.

3.5.1.1 Le pouvoir discriminant d'un sous ensemble de variables

Le pouvoir discriminant d'un sous ensemble de variables $L = \{y_1, ...,y_l\}$ est égale au nombre de couples assertions discriminés par au moins une variable de L.

$$PD(L,N) = \sum_i \sum_j \max_{y_k \in L} \left(\left(comp\left(V_{ik}, V_{jk}\right) \right) \right) \tag{3.11}$$

3.5.1.2 Le pouvoir discriminant original d'une variable

Le pouvoir discriminant original d'une variable y_l sur un ensemble d'objets par rapport à un ensemble de variables L est égal au nombre de couples d'assertions discriminés par y_l et par aucune autre variable. Il est donné par l'équation suivante :

$$PD(y_l,L,N) = \sum_i \sum_j \max_{(a_i,a_j)\in K} \left(comp\left(V_{il}, V_{jl}\right) - \max_{y_k \in L} \left(comp\left(V_{ik}, V_{jk}\right) \right), 0 \right) \tag{3.12}$$

Les critères que nous proposons s'apparentent à ceux ci, et en revenant aux définitions de bases nous constatons que le pouvoir discriminant est un critère de pertinence. Il nous a servi de base pour la création d'un nouveau critère de sélection : **le pouvoir discriminant but** d'une variable qui prend en considération la pertinence d'une variable par rapport à la variable but (classe). Nous développerons par la suite nos critères que nous nommerons *pouvoir discriminant but*, et *gain du pouvoir discriminant but*.

Les opérateurs d'agrégation que nous avons privilégiés sont booléens et ont permis de simplifier les critères et de les exprimer sous forme contingentielle, grâce aux formules de passage de Marchotorchino (Marcotorchino, 1984).

3.5.2 Approche par paires

L'approche par paires consiste à comparer les partitions induites par deux variables y_k y_{class}, paires d'objets (exemples) à paires d'objets.

Soit Ω la population observée composés de *n* objets ou individus ou instances : $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$.

Chaque éléments de Ω est caractérisé par un ensemble de *r* variables tel que $y = \{y_i, i = 1...r\}$ et une variable but ou classe y_{class}.

O_k est un ensemble de m_k modalités (valeurs) de la variable (attribut) y_k.

$O_k = \{m_1^k, m_2^k, \ldots, m_{q^k}^k\}$ où m_v^k est la modalité *v* de la variable y_k.

$y_{class} \in O_{class}$ tel que $O_{class} = \{m_1^{class}, .m_2^{class}, ..., m_p^{class}\}$.

Le critère contextuel est construit à partir de fonctions booléennes définies pour chaque variable sur une paire d'objets donnée, et est agrégé sur toutes les variables de telle manière à obtenir une mesure de pertinence forte et une mesure de pertinence faible que l'on introduira ci dessous.

A chaque variable y_k on associe une fonction bouléenne φ_{ij}^k relative à chaque paire d'instances (ω_i, ω_j), $i \neq j$.

$$(\omega_i, \omega_j) \mapsto \varphi_{i,j}^k = \varphi^k(\omega_i, \omega_j) = \begin{cases} 1 \Leftrightarrow y_k(\omega_i) = y_k(\omega_j) \\ 0 \ \textit{ailleurs} \end{cases}, \quad i, j = 1, \ldots, n \tag{3.13}$$

Si l'on formule φ_{ij}^k avec les modalités m_v^k, *v=1..*q^k,

$$\varphi_{ij}^k = \sum_{v=1}^{q^k} \varphi_k(\omega_i | m_v^k) \varphi_k(\omega_j | m_v^k) \tag{3.14}$$

$$\varphi_k(\omega_i | m_v^k) = \begin{cases} 1 & si\ y_k(\omega_i) = m_v^k \\ 0 & ailleurs \end{cases} \tag{3.15}$$

Les critères proposés sont élaborés à partir de la définition de la pertinence forte et faible (définitions 7 et 9 chapitre 1).

3.5.2.1 Pertinence faible par rapport à la variable but

Si l'on considère la population Ω et la variable descriptive y_k, le **pouvoir discriminant but** (DC : ***discriminating capacity*)** de y_k dans Ω est défini comme étant le nombre de paires discriminées sur l'ensemble de l'échantillon Ω**.** Ce critère permet de mesurer la *pertinence faible (Weakly Relevant WR)* d'une variable.

$$sur\left(\Omega\times\Omega\right)\mapsto WR(y_k,\Omega)=\sum_{i=1}^{n}\sum_{j=1}^{n}\overline{\varphi_{i,j}^{k}}\,\overline{\varphi_{i,j}^{class}} \tag{3.16}$$

Démonstration:

Soit **WR**, une fonction booléenne qui représente la pertinence faible d'une variable, cette fonction est égale à 1 quand la variable est pertinente.

$$\begin{aligned}
&\left(\omega_i,\omega_j\right)\mapsto \varphi_{i,j}^{k}=\varphi_k\left(\omega_i,\omega_j\right)=0\Leftrightarrow y_k(\omega_i)\neq y_k(\omega_j)\Leftrightarrow \overline{\varphi_{i,j}^{k}}=1\\
&\left(\omega_i,\omega_j\right)\mapsto \varphi_{i,j}^{class}=\varphi_{class}\left(\omega_i,\omega_j\right)\Leftrightarrow y_{class}(\omega_i)\neq y_{class}(\omega_j)\Leftrightarrow \overline{\varphi_{i,j}^{class}}=1\\
&\left(\omega_i,\omega_j\right)\mapsto WR\left(y_k,\omega_i,\omega_j\right)=\overline{\varphi_{i,j}^{k}}\,\overline{\varphi_{i,j}^{class}}=1
\end{aligned} \tag{3.17}$$

L'agrégation de WR sur toutes les paires d'instances donne le pouvoir discriminant but noté **DC**[1] (*discriminating capacity*)**;** DC est le nombre de paires d'instances (objets) discriminées sur la population Ω.

Le **pouvoir discriminant but** (DC) d'un ensemble *L* de *m* variable (*L*= ($y_1,\ldots y_m$)) dans Ω est le nombre de paires d'objets discriminées sur l'ensemble de l'échantillon Ω par l'ensemble *L* de variables, il est donné par :

$$DC(L,\Omega^2)=\sum_{i=1}^{n}\sum_{j=1}^{n}\overline{\prod_{k=1}^{m}\varphi_{i,j}^{k}}\,\overline{\varphi_{i,j}^{class}} \tag{3.18}$$

[1]" Une variable y_k est à pertinence faible par rapport à un échantillon N de données si en éliminant un sous ensemble donné de variables la variable y_k devient fortement pertinente (Blum, et al., 1997) pages (248-249)".

3.5.2.2 Pertinence forte par rapport à la variable but

Une variable est fortement pertinente, si sa contribution à décrire la variable but est exclusive par rapport à un ensemble de variables sur tout l'échantillon de population Ω.

A. Critère contextuel: Pertinence forte par rapport à un ensemble de variables et la variable but

DC (L, Ω^2) est une mesure qui considère la pertinence d'un ensemble de variables. A partir de cette mesure, le critère contextuel noté **gain du pouvoir discriminant but** (**DCG :** *discriminating capacity gain*) est un critère qui mesure la p*ertinence forte* (*Strong relevance : SR*) d'une variable, il a été élaboré afin de comparer la contribution exclusive d'une variable y_k par rapport à un ensemble L de variables sur toutes les paires de l'échantillon de population Ω.

$$DCG(y_k, L, \Omega^2) = \sum_{i=1}^{n}\sum_{j=1}^{n} \overline{\varphi_{i,j}^{class}} \times \overline{\varphi_{i,j}^{k}} \times \prod_{l=1}^{m} \varphi_{i,j}^{l} \qquad (3.19)$$

Le critère DCG est une mesure de pertinence forte. Il est différent de zéro quand aucune variable ou combinaison de variables ne peut discriminer un concept mis à part la variable y_k objet de l'étude. Nous considérons ce critère comme une approximation possible de l'approximation Markov blanket[1]. L'approximation Markov blanket est un formalisme qui permet de considérer les k-degrés d'interactions et permet de déterminer les variables prédominantes.

B. Corrélation sur l'ensemble des paires de données

La combinaison sur l'ensemble des paires d'objets des différents morceaux concernant l'ensemble des variables permet de déterminer une corrélation totale sur L et ΩxΩ.

[1] Markov blanket approximation : Si l'on considère deux variables pertinentes y_i et y_j ($i \neq j$), une variable y_j forme une approximation de type Markov blanket pour yi si $SU_{j,c} \geq SU_{i,c}$ et $SU_{i,j} \geq SUi,c$. Tel que SU est le gain d'information symétrique ((Yu, et al., 2004) page 1213)

Si : $$DCG(y_l, L, \Omega^2) = 0 \quad (3.20)$$

alors y_l est corrélé à l'ensemble des variables $L=\{y_{l,\dots} y_m\}$, donc il y a corrélation du point de vue discrimination entre la variable y_l et les variables $y_{l,\dots} y_m$ sur toutes les paires de données.

C. Variable à pertinence forte par rapport à un ensemble L de variables

Soit la variable y_k et L un ensemble de variables, y_k est une variable à pertinence forte par rapport à l'ensemble L de variables si :

$$DCG(y_k, L, \Omega^2) \neq 0 \quad (3.21)$$

La variable y_k dans ce cas peut être considérée comme une «*variable prédominante* ».

« Une variable pertinente est *prédominante* si elle n'a aucune approximation de type Markov blanket dans l'ensemble de variable auquel elle appartient. Les variables prédominantes ne doivent pas êtres éliminées de l'ensemble de variables sélectionnées » (définition de (Yu, et al., 2004) page1208).

3.5.3 Variable redondante par rapport à un ensemble *L* de variables

Comme les variables corrélées (identiques ou non) par partie ou en totalité posent problèmes, nous pouvons les détecter et ne pas les prendre en compte grâce à la mesure DC de la manière suivante :

si : $$DC(y_k,\Omega^2)\text{-}DC(L\text{-}\{y_k\}, \Omega^2) = 0 \quad (3.22)$$

alors y_k est une variable non pertinente ou redondante par rapport au sous ensemble L de variables sur l'ensemble des données Ω.

Les critères que nous venons de présenter sont très efficaces pour le traitement des données avec de grandes interactions (corrélation). Toutefois, ces critères sous leur

représentation par paires nécessitent quelques adaptations pour pouvoir les appliquer sur les grandes bases de données. Ce qui fait l'objet de nos travaux, en effet :

1. Grâce aux travaux de Marchotorchino (Marcotorchino, 1984) sur les formules de passage « paires-contingence » nous avons pu reformuler les critères par paires en critères contingentiels et ainsi proposer l'algorithme STRASS (*Strong Relevant Algorithm for Subset Selection*) (Senoussi, et al., 2008), un algorithme pour la sélection d'un ensemble optimal de variables.

2. Nous avons ensuite prouvé la stabilité de notre algorithme de filtrage par rapport à des portions de données (voir étude de la stabilité chapitre 4). Ce qui nous a permis par la suite de proposer l'utilisation de STRASS dans le cas des grandes bases de données en ne considérant qu'un échantillon réduit de l'ensemble total des exemples pour la sélection de variables pertinentes.

3.5.4 Critères en forme contingentielle

Avant de donner les critères de selection de variables sous leurs forms contingentielle en utilisant les formules de passage de Marchotorchino (Marcotorchino, 1984), nous présentons cette approche à travers les tableaux de contingence.

3.5.4.1 Approche contingentielle

L'approche contingentielle a permis une prolifération de critères statistique. Elle se base sur l'utilisation des tableaux de contingence. Ce type de tableaux comptabilise le nombre d'objets n_{uv} possédant simultanément la modalité u de y_{class} et v de y_k. Il est obtenu à partir du tableau descriptif des données. Si l'on considère la fonction :

$$\varphi_k\left(\omega_i \middle| m_v^k\right) = \begin{cases} 1 & si\ y_k\left(\omega_i\right) = m_v^k \\ 0 & \text{sinon} \end{cases}$$

$n_v^k = \sum_{i=1}^{n} \varphi_k\left(\omega_i \middle| m_v^k\right)$, $v=1..q^k$: est le nombre d'instances ayant la modalité v de la variable y_k.

Tableau 3-1. Tableau de contingence des variables y_k et y_{class}

y_k / y_{class}	m_1^k	…	m_v^k	…	$m_{q^k}^k$	Total
m_1^{class}	n_{11}		n_{1v}		n_{1q^k}	$n_{1.}$
⋮						
m_u^{class}	n_{u1}		n_{uv}		n_{uq^k}	$n_{u.}$
⋮						
m_p^{class}	n_{p1}		…		n_{pq^k}	$n_{p.}$
Total	$n_{.1}$		$n_{.v}$		$n_{.q^k}$	$n_{..}$

On ne manipulera pas les tableaux correspondant aux paires d'objets et leurs variables $y_1, ...,y_k,\ldots, y_r$. On manipulera par contre les tableaux de contingence qui seront liés aux couples de variables $(y_1, y_{class}),\ldots, (y_k, y_{class}),\ldots, (y_r, y_{class})$. Le tableau (3-1) représente le tableau de contincence lié aux variables y_k et y_{class}.

n_{uv} représente le nombre d'instances ayant la modalité v de la variable y_k et la valeur m_u^{class} pour la variable classe y_{class}. Par conséquent, sont définis :

- le nombre d'instances appartenant à la classe u :

$$n_{u.} = \sum_{v=1}^{q^k} n_{uv} \tag{3.23}$$

- le nombre d'instances ayant la modalité m_v^k :

$$n_{.v.} = \sum_{u=1}^{p} n_{uv} \tag{3.24}$$

- le nombre total d'instances :

$$n = \sum_{u=1}^{p} \sum_{v=1}^{q^k} n_{uv} \tag{3.25}$$

La différence entre l'approche par paires et l'approche statistique est qu'on ne se situe plus dans l'espace des paires mais dans l'espace des objets (instances). L'approche contingentielle est moins coûteuse en encombrement mémoire que celle des comparaisons par paires. Il serait donc intéressant de pouvoir passer d'une approche à une autre. Nous présentons dans ce qui suit les formules de passage.

3.5.4.2 Les formules de passage de Marchotorchino

Si l'on considère $\varphi_{i.}^k = \sum_{j=1}^{n} \varphi_{ij}^k = n_{v.}$ le nombre d'instances ayant la même modalité de ω_i pour y_k et $\varphi_{.j}^k = \sum_{i=1}^{n} \varphi_{ij}^k = n_{.v'}$ le nombre d'instances ayant la même modalité de ω_j pour y_k alors $\varphi_{..}^k$ s'écrit :

$$\varphi_{..}^k = \sum_{v=1}^{q^k} n_v^2 = n_{.1}^2 + \ldots + n_{.v}^2 + \ldots + n_{.q^k}^2 \tag{3.26}$$

avec n_v^2 le nombre d'instances ayant la même modalité *v* de y_k.

Rappelons que :

$$\left(\omega_i, \omega_j\right) \mapsto \varphi_{i,j}^k = \varphi^k\left(\omega_i, \omega_j\right) = \begin{cases} 1 \Leftrightarrow y_k\left(\omega_i\right) = y_k\left(\omega_j\right) \\ 0 \text{ sinon} \end{cases} \quad , \quad i, j = 1, \ldots, n$$

y_k prend ses valeurs dans $O_k = \left\{m_1^k, m_2^k, \ldots, m_{q^k}^k\right\}$ et possède q^k modalités.

Sachant que l'on peut exprimer φ_{ij}^k en fonction des modalités, on a :

$$\varphi_{ij}^{k} = \sum_{v=1}^{q^k} \varphi_k\left(\omega_i \middle| m_v^k\right) \times \varphi_k\left(\omega_j \middle| m_v^k\right)$$

avec $\varphi_k\left(\omega_i \middle| m_v^k\right) = \begin{cases} 1 & \text{si } \omega_i \text{ possède la modalité } m_v^k,\ y_k\left(\omega_i\right) = m_v^k \\ 0 & \text{sinon} \end{cases}$

$\sum_{i=1}^{n} y_k\left(\omega_i \middle| m_v^k\right) = n_{.v}$: représente le nombre d'instances possédant la modalité *v*, pour tout $v=1..q^k$.

$\sum_{i=1}^{n} y_{class}\left(\omega_i \middle| m_u^{class}\right) = n_{u.}$: représente le nombre d'instances possédant la modalité *u*, pour tout $u=1...p$.

La formule mixte liant y_k et la variable classe y_{class} est donnée par :

$$\sum_{i=1}^{n} y_{class}\left(\omega_i \middle| m_u^{class}\right) \times y_k\left(\omega_i \middle| m_v^k\right) = n_{uv}$$

Marchotorchino (Marcotorchino, 1984) linéarise les carrés de contingence et démontre que :

$$\sum_{u=1}^{p}\sum_{v=1}^{q^k} {n_{uv}}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \varphi_{ij}^{class} \times \varphi_{ij}^{k} \tag{3.27}$$

$$\sum_{u=1}^{p} {n_{u.}}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \varphi_{ij}^{class} = \varphi_{..}^{class} \tag{3.28}$$

$$\sum_{1}^{q^k} {n_{.v}}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \varphi_{ij}^{k} = \varphi_{..}^{k} \tag{3.29}$$

3.5.4.3 Transformation des critères en forme contingentielle

En utilisant ces formules de passage le pouvoir discriminant but de y_k est reformulé de la marinière suivante:

$$DC^c(y_k) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\overline{\varphi_{ij}^k} \times \overline{\varphi_{ij}^{class}} = \frac{1}{2}\left[n^2 - \sum_{v=1}^{q^k} n_v^2 - \sum_{u=1}^{p} n_u^2 + \sum_{u=1}^{p}\sum_{v=1}^{q^k} n_{uv}^2 \right] \quad (3.30)$$

Démonstration

Rappelons que :

$$\varphi_{ij}^k = \sum_{v=1}^{q^k} \varphi_k\left(\omega_i \middle| m_v^k\right) \times \varphi_k\left(\omega_j \middle| m_v^k\right)$$

$$\varphi_{ij}^{class} = \sum_{u=1}^{p} \varphi_{class}\left(\omega_i \middle| m_u^{class}\right) \times \varphi_{class}\left(\omega_j \middle| m_u^{class}\right)$$

$$\overline{\varphi_{ij}^k} = 1 - \varphi_{ij}^k \text{ , } \overline{\varphi_{ij}^{class}} = 1 - \varphi_{ij}^{class}$$

Le pouvoir discriminant but d'une variable est alors égal à :

$$DC^c\left(y_k\right) = \sum_{i=1}^{n}\sum_{j=1}^{n}(1-\varphi_{ij}^k)(1-\varphi_{ij}^{class}) =$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(1-\sum_{v=1}^{q^k}\varphi_k\left(\omega_i \middle| m_v^k\right)\times\varphi_k\left(\omega_j \middle| m_v^k\right)\right)\left(1-\sum_{u=1}^{p}\varphi_k\left(\omega_i \middle| m_u^{class}\right)\times\varphi_k\left(\omega_j \middle| m_u^{class}\right)\right)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n}1 - \sum_{u=1}^{p}\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_k\left(\omega_i \middle| m_u^{class}\right)\times\varphi_k\left(\omega_j \middle| m_u^{class}\right) - \sum_{v=1}^{q^k}\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_k\left(\omega_i \middle| m_v^k\right)\times\varphi_k\left(\omega_j \middle| m_v^k\right)$$

$$\sum_{u=1}^{p}\sum_{v=1}^{q^k}\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_k\left(\omega_i \middle| m_v^k\right)\times\varphi_k\left(\omega_j \middle| m_v^k\right)\times\varphi_k\left(\omega_i \middle| m_u^{class}\right)\times\varphi_k\left(\omega_j \middle| m_u^{class}\right)$$

Sachant que :

$$\sum_{i=1}^{n}\sum_{j=1}^{n}1 = n^2$$

Les calculs peuvent êtres simplifiés de la manière suivante :

$$\sum_{u=1}^{p}\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_k\left(\omega_i\left|m_u^{class}\right.\right)\times\varphi_k\left(\omega_j\left|m_u^{class}\right.\right)=\sum_{u=1}^{p}\sum_{i=1}^{n}\varphi_k\left(\omega_i\left|m_u^{class}\right.\right)\times\sum_{j=1}^{n}\varphi_k\left(\omega_j\left|m_u^{class}\right.\right)=\sum_{u=1}^{p}\sum_{i=1}^{n}\varphi_k\left(\omega_i\left|m_u^{class}\right.\right)\times n_u$$

$$=\sum_{u=1}^{p}n_u\times n_u=\sum_{u=1}^{p}n_u^2$$

$$\sum_{v=1}^{q^k}\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_k\left(\omega_i\left|m_v^k\right.\right)\times\varphi_k\left(\omega_j\left|m_v^k\right.\right)=\sum_{v=1}^{q^k}\sum_{i=1}^{n}\varphi_k\left(\omega_i\left|m_v^k\right.\right)\times\sum_{j=1}^{n}\varphi_k\left(\omega_j\left|m_v^k\right.\right)=\sum_{v=1}^{q^k}\sum_{i=1}^{n}\varphi_k\left(\omega_i\left|m_v^k\right.\right)\times n_v$$

$$=\sum_{v=1}^{q^k}n_v\times n_v=\sum_{v=1}^{q^k}n_v^2$$

$$\sum_{u=1}^{p}\sum_{v=1}^{q^k}\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_k\left(\omega_i\left|m_v^k\right.\right)\times\varphi_k\left(\omega_j\left|m_v^k\right.\right)\times\varphi_k\left(\omega_i\left|m_u^{class}\right.\right)\times\varphi_k\left(\omega_j\left|m_u^{class}\right.\right)$$

$$=\sum_{u=1}^{p}\sum_{v=1}^{q^k}\sum_{i=1}^{n}\varphi_k\left(\omega_i\left|m_v^k\right.\right)\times\varphi_k\left(\omega_i\left|m_u^{class}\right.\right)\times\sum_{j=1}^{n}\varphi_k\left(\omega_j\left|m_v^k\right.\right)\times\varphi_k\left(\omega_j\left|m_u^{class}\right.\right)$$

$$=\sum_{u=1}^{p}\sum_{v=1}^{q^k}\sum_{i=1}^{n}\varphi_k\left(\omega_i\left|m_v^k\right.\right)\times\varphi_k\left(\omega_i\left|m_u^{class}\right.\right)\times n_{uv}$$

$$=\sum_{u=1}^{p}\sum_{v=1}^{q^k}n_{uv}\times n_{uv}=n_{uv}^2$$

Soit en final le pouvoir discriminant but sous forme contingentielle d'un ensemble *L* de variable $DC^c(L)$:

$$DC^c(L)=\frac{1}{2}\left[n^2-\sum_{m=1}^{qx}n_{x_m}^2-\sum_{u=1}^{p}n_u^2+\sum_{u=1}^{p}\sum_{m=1}^{qx}n_{ux_m}^2\right] \tag{3.31}$$

Le gain du pouvoir discriminant but sous forme contingentielle d'une variable donné y_k est ainsi reformulé de la manière suivante:

$$DCG^c\left(y_k\right)=DCG^c\left(y_k,S-y_k\right)=DC^c\left(S\right)-DC^c\left(S-y_k\right)$$

$$DCG^c(y_k)=\sum_{i=1}^{n}\sum_{i=1}^{n}\overline{\prod_{k=1}^{S}\varphi_{i,j}^k}\;\overline{\varphi_{i,j}^{class}}-\sum_{i=1}^{n}\sum_{i=1}^{n}\overline{\prod_{k'=1,k'\neq k}^{S}\varphi_{i,j}^k}\;\overline{\varphi_{i,j}^{class}}$$

$$DCG^c(y_k) = \frac{1}{2}\left[n^2 - \sum_{m=1}^{qx} n_{x_m}^2 - \sum_{u=1}^{p} n_u^2 + \sum_{u=1}^{p}\sum_{m=1}^{qx} n_{ux_m}^2 \right] - \frac{1}{2}\left[n^2 - \sum_{m',m'\neq m}^{qx'} n_{x'_m}^2 - \sum_{u=1}^{p} n_u^2 + \sum_{u=1}^{p}\sum_{m=1,m'\neq m}^{qx'} n_{ux'_m}^2 \right] \quad (3.32)$$

n_{x_m} est le nombre d'instances ayant la modalité x_m de la partition collective représentée par le sous ensemble de variables *S*.

$n_{x'_m}$ est le nombre d'instances ayant la modalité x'_m de la partition collective représentée par le sous ensemble de variables *S*-y_k.

x_m, x'_m : sont les modalités collectives, correspondant à l'agrégation des variables dans le sous ensemble *S* et *S*-y_k respectivement.

$$n_{ux'}^2 = \sum_{i=1}^{n} \varphi_{class}(\omega_i|u)\varphi_L(\omega_i|x_m) = \sum_{i=1}^{n} \varphi_{class}(\omega_i|u) \prod_{y_k \in L} \varphi_k(\omega_i|m_v^k) \quad (3.33)$$

$n_{ux'_m}$ est le nombre d'instances ayant la modalité *u* de la variable classe (m_u^{class}) et la modalité x'_m de la partition collective.

Le DCG^c, de par sa définition est un critère de sélection de variables fortement pertinentes. En effet, le DCG^c d'une variable y_k est non nul lorsqu'il n'existe aucune variable ou combinaison de variables descriptives à l'exception de y_k pouvant substituer à y_k pour la détermination de la variable endogène (classe). Il est nul dans le cas contraire. Le DCG^c est par conséquence une mesure contextuelle capable de détecter aussi bien la pertinence d'une variable par rapport à un ensemble de variables, mais aussi bien la non pertinence d'une variable par rapport à un ensemble de variables lorsque le sous ensemble et la variable sont dépendants. Il garantit ainsi l'élimination des attributs totalement inutiles en regard de la variable endogène.

3.5.5 Algorithme de sélection de variables STRASS

Les critères que nous avons élaboré sous forme contingentielle ont été implémentés dans un algorithme gourment de sélection de variables (Pudil, et al., 1994) que nous avons noté STRASS (Senoussi, et al., 2008), (Senoussi, et al., 2011(a)), (Senoussi, et al., 2011(b)), (Senoussi, et al., 2012). L'algorithme STRASS, nous permet de détecter un ensemble *optimal* de variables pertinentes. L'algorithme effectue une recherche en mode *génération séquentielle bidirectionnelle*, c'est à dire qu'un noyau de variables est composé en partant d'un ensemble vide *S* qui est construit petit à petit jusqu'à l'obtention d'un ensemble ayant le même degré de pertinence que l'ensemble de départ. A chaque ajout de variable tout l'ensemble est remis en question au fur et à mesure de la composition du noyau.

Soit S_o l'ensemble total de variables et S_f l'ensemble des variables sélectionnées. La stratégie de recherche adoptée n'est pas une génération complète mais une génération heuristique. En effet, la combinatoire de la génération complète est incompatible avec la gestion d'une très grande masse de données et donc avec le processus de l'ECD. Le critère d'arrêt est défini par l'obtention d'un pourcentage ρ sur le pouvoir discriminant but total des variables sélectionnées. ρ est un seuil donné par l'utilisateur représentant une perte sur le pouvoir discriminant but total des variables sélectionnées S_f par rapport au pouvoir discriminant but de la population entière de variables S_o. Afin de ne pas tomber dans un problème combinatoire dû à toutes les paires d'objets existantes dans un problème, nous avons utilisé les critères DC^c et DCG^c sous forme contingentielle.

La complexité de l'algorithme est $\Theta(r, n)$, où r est le nombre de variables descriptives (cardinal de S_0) et n est le nombre d'instances à discriminer (cardinal de Ω). L'algorithme se décompose en trois étapes suivant l'initialisation :

Algorithme STRASS

S_0={y1,y2,....yr} ; // ensemble de variables à traiter
S_f =Ø ; // ensemble de variables sélectionnées
DC_{Tot}=DC(S) ; // le pouvoir discriminant de l'ensemble total des variables
ρ // un seuil représentant une perte sur DC_{Tot}

Pour chaque variable y_k de S_0 **faire** // *Sélection des variables prédominantes*

Si $DCG^c(y_k, S_0 - \{y_k\}) \neq 0$
alors $S_f = S_f + \{y_k\}$;
$S_0 = S_0 - \{y_k\}$;
finSi
fin

Tant que $DC^c(S_f) < DC_{Tot}$ **faire** // *Sélection des variables à pertinence faible*

DC_{max}=0 ;

Pour chaque variable y_k de S_0 **faire**

Si $DC^c(S_f + \{y_k\}) > DC_{max}$

alors $DC_{max} = DC^c(S_f + \{y_k\})$;
$y_{kmax} = y_k$;
$S_f = S_f + \{y_{kmax}\}$;
$S_0 = S_0 - \{y_{kmax}\}$;
finSi
fin

Pour chaque variable *yk* de S_0 **faire** // *Suppression des variables redondantes*

Si $DC^c(y_k) - DC(S_f - \{y_k\}) >= \rho * DCtot$

alors S= $S_f - \{y_k\}$;
finSi
fin

Retourner S_f;

1° Etape : Sélection des variables à pertinence forte « les variables prédominantes »

Cette phase permet la sélection des variables incontournables ou essentielles, car elles sont les seules à discriminer les classes. Aucune autre variable ou ensembles de variables ne peut remplacer le rôle de ces variables.

2° Etape : Sélection des variables restantes

On recherche les variables qui ont le pouvoir discriminant le plus grand, ce sont des variables à pertinence faible c'est-à-dire que l'association de ces variables avec le sous ensemble prés-sélectionné permet d'augmenter le pouvoir discriminant des variables.

3° Etape : Suppression des variables redondantes

On supprime les variables devenues redondantes par rapport à l'ensemble des variables prés-sélectionnées lors de l'ajout d'une nouvelle variable. Cette étape garantit la non-redondance (corrélation) totale ou par morceaux des variables de l'ensemble sélectionné.

L'arrêt de l'algorithme est défini soit par l'obtention du même pouvoir discriminant que l'ensemble totales des variable S_0, soit en considérant une perte ρ sur le pouvoir discriminant total des variables sélectionnées S_f.

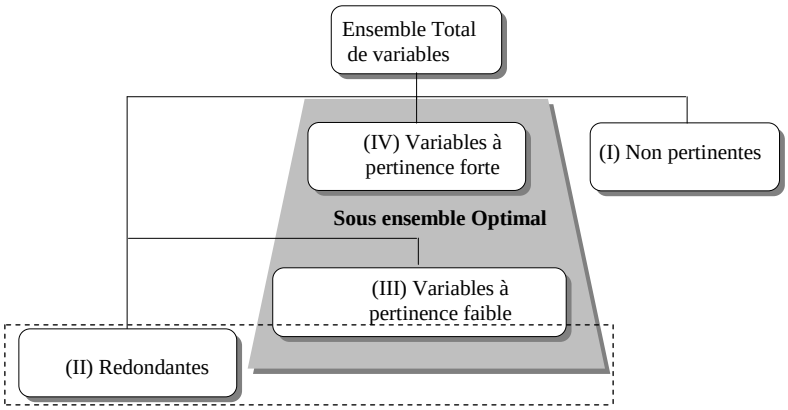

Figure 3-1 : *Catégorisation des variables* (Yu, et al., 2004)

3.5.6 Catégorisation des variables

D'après Yu et Lui (Yu, et al., 2004) un ensemble entier de variables peut être conceptuellement divisé en quatre groupes (Figure 3-1) : variables non pertinentes (I),

variables redondantes (II), variable à pertinence faible (III), et les variables à pertinence forte (IV). Le sous ensemble *optimal* de variables quand à lui est constitué de toutes les variables du groupe III et IV.

STRASS sélectionne un ensemble *optimal* de variables, ce dernier est constitué des variables à pertinence forte, les variables à pertinence faible et aucune variable redondante ou partiellement redondante. En plus STRASS retourne un sous ensemble de variables ordonnées suivant leur pouvoir discriminant et donne la catégorisation de variable équivalent à celle défini par Yu et Lui (Yu, et al., 2004).

3.6 Conclusion

Les travaux récents en sélection de variables visent à traiter les grandes bases de données et donc de trouver des algorithmes de filtrage adapté aux grandes dimensions. Les grandes bases de données posent le problème des grandes interactions entre les variables. Dans notre étude nous avons essayé de répertorier les algorithmes de filtrage qui ont été spécialement conçu afin de résoudre ces problèmes (la dimension et la corrélation) tel que CFS, mRMR, FCBF et INTERACT. Ces algorithmes ont été reconnus comme étant les plus performants selon ce point de vue. Toutefois, aucun de ces algorithmes ne détecte les variables partiellement redondantes. Problème auquel on s'est attelé dans le cadre de nos travaux de thèse et que nous allons essayer de résoudre avec l'algorithme STRASS, un algorithme que nous avons conçu à partir de critères d'évaluation contextuels. Les critères contextuels permettent de détecter aussi bien les variables à pertinence fortes que la contribution d'un ensemble de variables (pertinence faible) en examinant l'interaction d'un ensemble de variables, ce qui nous a aussi permis de détecter la redondance contextuelle ou partielle.

Chapitre 4

Evaluation de l'algorithme de filtrage

Sommaire

4.1 Introduction

Afin de pouvoir résoudre efficacement un problème, il est important de savoir quelle méthode de sélection de variables utiliser et dans quelles conditions. Pour cela il existe deux types d'évaluation : *directe* et *indirecte*. L'évaluation directe est la plus simple et la plus objective. Cependant, elle nécessite la connaissance à priori des variables pertinentes. La deuxième approche, qualifiée d'évaluation indirecte, estime la performance de la méthode de sélection à travers les résultats obtenus sur l'algorithme d'induction, l'idée étant de voir l'impact d'un sous ensemble sélectionné sur les performances de l'algorithme d'apprentissage (Senoussi, et al., 2008).

Dans le chapitre 2, nous avons montré les problèmes liés au diagnostic ainsi que les solutions préconisées dans le monde industriel. Dans ce contexte et afin de valider notre méthode en diagnostic de défauts, nous allons utiliser par la suite l'algorithme STRASS sur différents cas en diagnostic industriel (Senoussi et al., 2011(a)), (Senoussi et al., 2011(b)), (Senoussi et al., 2012).

4.2 Evaluation directe : Bases artificielles

Si nous connaissons à priori certaines variables pertinentes ou/et redondantes, nous pouvons attendre d'une méthode de sélection de variables qu'elle mette les variables pertinentes en tête de liste ordonnée ou bien sélectionnées dans le sous ensemble minimum de variables sélectionnées. De la même manière, on peut espérer ne pas trouver les variables redondantes en tête de liste ou bien sélectionnées dans le sous ensemble minimum de variables sélectionnées.

Notre algorithme a été implanté sous matlab 7.5 et nous avons utilisé les outils d'ECD de la plateforme WEKA (Witten, et al., 2000), tel que les algorithmes de sélection de variables et les algorithmes de classification : les arbres de décision (C4.5), les k plus proches voisins (IB_k) et les réseaux de neurones (MLP).

4.2.1 Bases de données utilisées

Nous avons fait une évaluation directe de l'algorithme STRASS sur 13 ensembles artificiels de données de références dont nous connaissons à priori les variables pertinentes (KDD-UCI). Nous avons privilégié ces ensembles car ces derniers sont très utilisés en fouille de données. En effet, ces ensembles de données comportent des variables qui sont très connues pour leurs grandes corrélations. Elles pourront ainsi tester notre algorithme.

LED7 et LED24 Display Domain (Breiman, et al., 1984) : deux ensembles de données composés de 7 caractéristiques correspondant aux 7 segments pour LED7 auquel il a été ajouté 16 caractéristiques générées aléatoirement pour LED24.

L'ensemble BOOL : composé à l'origine d'une fonction de 6 variables booléennes donnant une variable booléenne exogène y_{but} tel que :

$$y_{but} = (x_1 \oplus x_2) \vee (x_3 \wedge x_4) \vee (x_5 \wedge x_6) \tag{4.1}$$

Nous avons ajouté à ces 6 variables, six autres variables booléennes générées aléatoirement.

Le domaine parity : est composé d'une fonction de 3 variables booléennes comme dans le cas de BOOL auxquelles on a associé 7 variables générées aléatoirement.

$$y_{but} = x_1 \oplus x_2 \oplus x_3 \tag{4.2}$$

Cet ensemble est particulièrement intéressant, car aucune variable pertinente prise isolément ne peut être distinguées des variables non pertinentes.

L'ensemble parity+2 : est l'ensemble parity précédent auquel on a ajouté 2 variables redondantes x_{11} et x_{12}. tel que : x_1=x_{11} et x_2=x_{12}.

Ce domaine teste la capacité d'un algorithme à travailler en présence de variables redondantes (doublons) de variables pertinentes.

Coral (John, et al., 1994) : cet ensemble, comporte six variables binaires de x_1 à x_6 parmi lesquelles x_5 est non pertinente et x_6 est corrélée à 75% avec la variable but y_{but}.

$$y_{but} = (x_1 \wedge x_2) \vee (x_3 \wedge x_4) \tag{4.3}$$

Les fonctions d'Argawal

Argawal (Agrawal, et al., 1992) a mis au point des ensembles pour la classification, ces ensembles sont générés à partir de fonctions, les variables sont générées aléatoirement en fonction des distributions données par le tableau suivant :

Tableau 4-1. Variables proposées par Argawal (Agrawal, et al., 1992)

Variable	Description	Valeur
Salary	salaire	uniformément distribué entre 20,000 et 150,000
Commission	commission	Si le salaire >=75000 alors commission=0 sinon uniformément distribué entre 10000 et 75000
Age	age	uniformément distribué entre 20 et 80
Education level	niveau d'étude	uniformément distribué [0,1,...,4]
Car	marque de la voiture	uniformément distribué [0,1,...,20]
Code Zip	code zip de la ville	9 variables zipcodes
hvalue	valeur de la maison	uniformément distribué entre 0.5k10000 et 1.5k1000000, avec $k \in \{0 \ldots 9\}$ depend du zipcode.
hyears	années de location de la maison	uniformément distribué [1,2,...,30]
loan	le total	uniformément distribué entre 1 et 500000

$$\text{Fonction 1 :} \quad \begin{cases} GrpA: & ((age < 40) \vee (60 \leq age)) \\ GrpB: & ailleurs \end{cases} \tag{4.4}$$

$$\text{Fonction 2 :} \begin{cases} GrpA: & \big((age < 40) \wedge (50k \le salary \le 100k)\big) \vee \\ & \big((age \le 40 < 60) \wedge (75k \le salary \le 125k)\big) \vee \\ & \big((age \ge 60) \wedge (25k \le salary \le 75k)\big) \\ GrpB: & ailleurs \end{cases} \quad (4.5)$$

$$\text{Fonction 3 :} \begin{cases} GrpA: & \big((age<40) \wedge \\ & \big(((elevel \in [0..1]) \wedge (25k \le salary \le 75k)) \vee ((elevel \in [2..3]) \wedge (50k \le salary \le 100k))\big)\big) \vee \\ & \big((40 \le age < 60) \wedge \\ & \big(((elevel \in [1..3]) \wedge (50k \le salary \le 100k)) \vee (((elevel=4) \wedge (75k \le salary \le 1250k)))\big)\big) \vee \\ & \big((age \ge 60) \wedge \\ & \big(((elevel \in [2..4]) \wedge (50k \le salary \le 100k)) \vee ((elevel=1) \wedge (25k \le salary \le 75k)))\big)\big) \\ GrpB: & ailleurs \end{cases} \quad (4.6)$$

$$\text{Fonction 4 :} \begin{cases} GrpA: & disponible > 0 \\ GrpB: & ailleurs \end{cases} \quad (4.7)$$

$$disponible = \big(0.67.\big(salary + commission\big) - 0.2.loan - 10k\big)$$

4.2.2 Sous ensemble de variables sélectionnées

Lors de l'expérimentation, nous avons comparé nos résultats par rapport aux différentes méthodes de sélection de variables :

- L'algorithme de référence en sélection de variable : Relief (Kononenko, 1994).
- Deux méthodes enveloppe : WrapperGA(C4.5)[1] et WrapperGA(IB_1)[2].

[1] Méthode enveloppe de sélection de variables utilisant la précision de C4.5 et une stratégie de recherche par Algorithme Génétique.

[2] Méthode enveloppe de sélection de variables utilisant la précision du 1-plus proche voisin et une stratégie de recherche par Algorithme Génétique.

- L'algorithme ConsistancyGA[1] (Lanzi, 1997) et GainRatio[2] (Quilan, 1986).
- Les algorithmes contextuels de sélection de variables : mRMR (Peng, et al., 2005), CFS (Hall, 2000), FCBF (Yu, et al., 2004) et INTERACT (Zhao, et al., 2007).

Tous ces algorithmes sont disponible dans WEKA (Witten, et al., 2000), mis à part l'algorithme mRMR qui est disponible en langage Matlab en ligne à l'adresse suivante : http://penglab.janelia.org/software/Hanchuan_Peng_Software/software.html.

Toutefois, Afin de rendre comparable les résultats obtenus par ces algorithmes, nous avons :

- Fait tourner Relief sur l'ensemble total d'objets et non pas sur un certain nombre d'instances choisies aléatoirement comme préconisé à l'origine.
- Afin d'obtenir une liste ordonnée de variables, nous avons utilisé l'option de recherche Ranksearch pour CFS et l'option Ranker pour Relief et GainRatio.
- L'évaluation du mérite d'un ensemble de variables dans CFS est calculée en utilisant le gain d'information comme fonction de corrélation.

Les résultats de sélection de variables sont donnés par les tableaux (4-2) et (4-3).
- : indique l'omission d'une variable pertinente.

STRASS est le seul algorithme à avoir sélectionner les variables x_1,x_2,x_3,x_4,x_5 seulement pour les deux bases LED7 et LED24 et ainsi montrer la suffisance des 5 segments pour la représentation des chiffres.

En ce qui concerne les ensembles MONK, STRASS détecte toutes les variables pertinentes, vient ensuite la méthode enveloppe WrapperGA(IB_1) avec une bonne détection sur les ensembles MONK1 et MONK3.

[1] Algorithme de sélection de variables utilisant le critère de consistance avec une stratégie de recherche par Algorithme Génétique.

[2] Algorithme de sélection de variables utilisant le gain d'information comme critère de sélection de variable avec une stratégie de recherche de type FS.

Les algorithmes WrapperGA(C4.5), ConsistancyGA, mRMR, INTERACT et STRASS sont les seuls à avoir détecté la redondance de la variable x_6 dans l'ensemble Corral cette variable étant corrélée à 75% à la variable but est jugée pertinente par CFS, GainRatio et Relief car les trois algorithmes n'arrive pas à évaluer la pertinence d'une variable par rapport à la combinaison des autres variables.

Pour les domaines Parity, notre algorithme, Relief, WrapperGA(C4.5), WrapperGA(IB_1) et INTERACT établissent que les variables x_1, x_2, x_3 sont pertinentes alors que les 7 autres sont inutiles.

Pour l'ensemble Parity+2 notre algorithme, WrapperGA(IB_1) et INTERACT arrive à détecter respectivement la redondance de x_6, ainsi que les variables doublon x_{11} et x_{12}.

Tableau 4-2. Variables sélectionnées les algorithmes de filtrage (Senoussi, et al., 2008)

Base de données	Variables pertinentes	Relief	GainRatio	WrapperGA (C4.5)	Consistancy (GA)	WrapperGA (1NN)
LED7	x_1,x_2,x_3,x_4,x_5	toutes	toutes	Pas assez d'exemples	x_1,x_2,x_3,x_4,x_5	toutes
LED24	x_1,x_2,x_3,x_4,x_5	toutes	toutes	x_1,x_2,x_3,x_4,x_5 ,x_7,x_{16},x_{17}, x_{18}	x_3, x_4, x_5, x_9, x_{10},x_{11},x_{15}, x_{16},x_{17},x_{18}, x_{19},x_{20},x_{22}, x_{23}	x_1,x_2,x_3,x_4,x_5 ,x_6,x_7,x_9, x_{10},x_{12},x_{14}, x_{18}, $_{19},x_{20},x_{23}$
MONK1	x_1,x_2,x_5	toutes	toutes	x_1,x_2,x_3,x_4,x_5	x_1,x_2,x_5	x_1,x_2,x_5
MONK2	x_1,x_2,x_3,x_4,x_5,x_6	x_1,x_2,x_3,x_4,x_5, x_6	-,-,-,x_4,x_5,x_6	x_4,x_5,x_6	x_1,x_2,x_3,x_4,x_5, x_6	x_2,x_3,x_4,x_5,x_6
MONK3	x_5,x_4,x_2	toutes	toutes	x_2,x_5,x_6	x_1,x_2,x_4,x_5	x_5,x_4,x_2
Parity	x_1,x_2,x_3	x_1,x_2,x_3	x_{10}, x_8, x_5	x_1,x_2,x_3	x_1,x_2,x_3,x_4,x_6	x_1,x_2,x_3
Parity 2	x_1, x_2,x_3	x_1,x_2,x_3,x_{11}, x_{12}	x_{10}, x_8, x_5	x_1,x_2,x_3,x_6	x_1,x_2,x_3 ,x_5,	x_1,x_2,x_3
Corral	x_1,x_2,x_3,x_4	x_1,x_2,x_3,x_4,x_6	x_1,x_2,x_3,x_4,x_6	x_1,x_2,x_3,x_4	x_1,x_2,x_3,x_4	x_1,x_2,x_5,x_6
Bool	x_1,x_2,x_3,x_4,x_5,x_6	x_1,x_2,x_3,x_4,x_5, x_6,x_7 , x_{12}	x_3,x_4,x_5,x_6	x_1,x_2,x_3,x_4,x_5 ,x_6,x_{10}	x_1,x_2,x_3,x_4,x_5 ,x_6	x_1,x_2,x_3,x_4,x_5 ,x_6
F1	x_3	x_3	x_3	x_3	x_3	x_3
F2	x_1 , x_3	x_1,x_2,x_3	x_1	x_2 , x_3	x_1	x_1,x_2,x_3
F3	x_1, x_3, x_4	tout	x_2, x_3, x_4	x_1, x_3, x_4	x_2, x_3, x_4	x_1,x_2,x_3,x_4
F4	x_1, x_2,x_9	x_1, x_2,x_9	x_9	x_1, x_2,x_9	x_1, x_2, x_9	x_1, x_2,x_9
gain		4/13	1/13	5/13	7/13	7/13

Pour les ensembles d'Argawal notre algorithme arrive à détecter les variables pertinentes sur trois bases (F1, F2 et F3). Cependant, pour la fonction F4 la sélection des variables pertinentes est obtenue en utilisant une perte de 2% sur le pouvoir discriminant total des variables DC_{tot}.

Tableau 4-3. Variables sélectionnées par STRASS et les autres algorithmes de filtrage (Senoussi, et al., 2008)

Bases de données	Variables pertinentes	STRASS	mRMR	CFS	FCBF	INTERACT
LED	x_1,x_2,x_3,x_4,x_5	x_1,x_2,x_3,x_4,x_5	-,x_2,x_3,x_4,-, x_6,x_7	toutes	toutes	toutes
Led 24	x_1,x_2,x_3,x_4,x_5	x_1,x_2,x_3,x_4,x_5	-,x_2,x_3,x_4,x_5,-,x_7	$x_1,x_2,x_3,x_4,x_5,x_6,x_7$,$x_8,x_{14},x_{16},x_{19},x_{21}$	x_7,x_5,x_4,x_2,x_3,x_6, $x_{14},x_{19},x_{21},x_{15},x_{12}$, x_{11},x_{20}	$x_5,x_{17},x_3,x_2,x_{15}$, $x_{14},x_8,x_{16},x_7,x_4,x_6$,$x_{22},x_{23},x_{18},x_9,x_1$, x_{20},x_{19},x_{11}
MONK1	x_1,x_2,x_5	x_1,x_2,x_5	x_1,-,x_4,x_5	x_1,-,x_3,x_4,x_5	x_1,-,x_3,x_4,x_5	x_1,x_2,x_5
MONK2	x_1,x_2,x_3,x_4,x_5, x_6	x_1,x_2,x_3,x_4,x_5, x_6	x_1,x_2,x_3,x_4,x_5,x_6	-,-,-,x_4,x_5,x_6	-,-,-,x_4,x_5,x_6	x_1,x_2,x_3,x_4,x_5,x_6
MONK3	x_5,x_4,x_2	x_5,x_4,x_2	x_2,-,x_5,x_6	x_2,-,x_5,x_6	x_2,-,x_5,x_6	x_1,x_2,-,x_4,x_5,-
Parity	x_1,x_2,x_3	x_1,x_2,x_3	-,x_2,-,x_{10},x_6	x_{10}, x_8, x_5	x_{10}	x_1,x_2,x_3
Parity2	x_1,x_2,x_3	x_1,x_2,x_3	-,x_2,-,x_{10},x_6	x_{10}, x_8, x_5	x_{10}	x_1,x_2,x_3
Corral	x_1,x_2,x_3,x_4	x_1,x_2,x_3,x_4	x_1,x_2,x_3,x_4	x_1,x_2,x_3,x_4,x_6	x_3,x_1,x_2,x_4,x_6	x_1,x_2,x_3,x_4
Bool	x_1,x_2,x_3,x_4,x_5, x_6	x_1,x_2,x_3,x_4,x_5, x_6	-,-,x_3,x_4,x_5,x_6, x_7,	x_1,x_2,x_3,x_4,x_5,x_6	$x_5,x_6,x_3,x_4,x_{10},x_{12}$	x_1,x_2,x_3,x_4,x_5,x_6
F1	x_3	x_3	x_3	x_3	x_3	x_3
F2	x_1, x_3	x_1, x_3	x_1, -, x_8	x_1	x_1	x_1
F3	x_1, x_3, x_4	x_1, x_3, x_4	- ,x_2, -, x_4, x_8	x_2, x_3, x_4	x_2, x_3, x_4	x_2, x_3, x_4
F4	x_1, x_2, x_9	x_1, x_2, x_9	x_1,x_2, x_8,-	x_9	x_1, x_9	x_1,x_2, x_8,-
gain		13/13	3/13	1/13	1/13	7/13

Bien que certains algorithmes de sélection de variables on été spécialement conçu pour le traitement des variables corrélées, comme c'est le cas de : mRMR, CFS, FCBF et INTERACT, ces derniers ont donné des performances moyennes voir même très faible en présence de variables fortement corrélées. En effet, Hall (Hall, 1998) précise que CFS donne de bons résultats quand il y a une interaction modérée entre les variables.

Ceci peut s'expliquer par le fait que CFS, FCBF et mRMR calculent l'inter-corrélation entre *paires* de variables, ceci permet de déterminer les redondances entres deux variables (équivalence) et non pas la redondance d'une variable par rapport à un ensemble de variables. Par conséquent ces algorithmes n'arrivent pas à identifier les grandes interactions entre variables (la pertinence faible). De plus les algorithmes présentent une difficulté à

travailler sur les attributs redondants par partie. Problème résolu par l'algorithme proposé STRASS.

4.3 Evaluation indirecte : bases réelles

Elle consiste à induire un classifieur à partir du sous ensemble de variable sélectionnées puis à estimer sa qualité. Pour cella nous avons testé notre algorithme sur des bases réelles de différentes natures et domaines d'applications communément traitées dans la communauté d'ECD, ce sont des ensembles de référence, disponibles en ligne sur le site internet de *UCI Knowledge Discovery in Databases Archive* à l'adresse suivante : www.ics.uci.edu/~mlearn/MLRepository.html. Nous les décrivons brièvement dans le tableau (4-4). Chacune de ces bases peut contenir des variables continues ou discrètes. Cependant la plupart des algorithmes de sélection de variables ne peuvent pas manipuler des variables continues, ce qui est le cas de STRASS. Nous avons donc jugé bon de discrétiser les attributs continus avant de les présenter aux algorithmes de sélection de variables par un algorithme de discrétisation.

Tableau 4-4. Description des bases réelles

Bases de données	Exemples	Variables	Classes
Heart	270	13	2
German	1000	24	2
Pima	768	8	2
Iris	150	4	3
Ionosphere	351	34	2
Vehicle	846	18	4
Glass	214	9	7
Wine	178	13	3
Lang cancer	32	57	2
Sonar	208	60	2
Austra	690	14	2
segment	2310	19	7
waveform	5000	21	3
arrhythmia	452	279	13
Mushroom	8124	22	2
kr-vs-kp	3196	37	2
Nursery	12960	8	4

4.3.1 Choix de l'algorithme de discrétisation

Afin d'évaluer les performances des algorithmes de discrétisation que nous avons présenté au chapitre 1 et de pouvoir les comparer, nous allons donner les résultats obtenu sur 4 bases de données parmi les bases habituellement utilisées par la communauté qui travaille sur les problèmes de discrétisation. Pour la mesure d'évaluation, nous avons utilisé les arbres de décision (C4.5). Pour les algorithmes de discrétisation nous avons implémenté Chimerge et Chi2 sur Matlab et nous avons utilisé l'algorithme MDLM disponible sur la plateforme WEKA (Witten & Frank, 2000). Nous avons regroupé dans le tableau (4-5), pour chaque base de test, les performances en classification après discrétisation par les différentes méthodes.

Sur l'ensemble des bases de données, l'algorithme MDLM donne les meilleurs résultats de classification. L'algorithme MDLM (Fayyad, et al., 1993) (voir chapitre 1) est aussi l'algorithme le plus utilisé par la communauté de chercheurs qui travaille sur les problèmes de sélection de variables. Nous avons donc jugé bon de travailler avec cette méthode de discrétisation afin de comparer nos résultats aux leurs.

Tableau 4-5. Classification de C4.5 avec discrétisation des données

Bases de données	Chimerge	Chi2	MDLM
Heart	72.2	70.74	74.07
Iris	91.3	94	93.3
Wine	82.02	91.57	93.82
Glass	74.29	53.7	89.71
Gain	0/4	1/4	3/4

4.3.2 Classification après sélection de variables

Nous allons étudier l'impact de notre algorithme de filtrage sur les résultats obtenus après la fouille de données. Nous nous intéressons aussi bien au nombre de variables sélectionnées qu'aux performances de classification obtenues. Dans cette évaluation nous

allons comparer cette fois STRASS par rapport aux algorithmes contextuels seulement à savoir : mRMR, CFS, FCBF et INTERACT.

Le taux de classification a été estimé par 10-validations croisées (*10-fold cross validation*) pour les petites bases de données et par 1-validation croisée (*1-fold cross validation*) pour les grandes bases de données. Nous recensons les résultats dans les tableaux (4-6) à (4-8).

Tc: Taux de classification, S: nombre de variables sélectionnées.

Tableau 4-6. Classification de C4.5 avec et sans filtrage de données

Bases de données	**C4.5**		C4.5+ STRASS		C4.5+ mRMR		C4.5+ CFS		C4.5+ FCBF		C4.5+ INTERACT	
	Tc	Toutes	Tc	S	Tc	S	Tc	S	Tc	S	Tc	S
Heart	**83.7**	13	85.18+	8	83.7	8	83.3-	6	84.4+	5	83.33-	9
German	**75.1**	24	75	10	71.2-	10	73.3-	5	73.3-	5	74.1-	13
Pima	**77.4**	8	77.4	6	75.52-	6	76.9-	3	77.4	6	77.4	6
Iris	**94**	4	94	3	94	3	96+	2	94	2	94	3
Ionosphere	**89.1**	34	91.73+	6	89.4+	6	89.7+	12	90.3+	5	91.73+	8
Vehicle	**72.1**	18	82.54+	15	70.68-	15	73.3+	9	57.8-	4	72.1	17
Glass	**76.1**	9	76.1	6	72.89-	6	76.1	7	75.7-	5	76.1	6
Wine	**93.2**	13	95.5+	5	97.2+	5	94.3+	8	93.8+	10	94.9+	5
Sonar	**79.8**	60	82.69+	12	81.25+	12	80.28+	19	76.44-	10	79.8	12
Austra	**86.5**	14	86.5	13	86.2-	13	85.5-	7	86.5	7	86.37	13
segment	**94.76**	19	96.1+	4	96.1+	4	95.36+	8	95.58+	6	94.93	9
arrhythmia	**72.78**	279	74.11+	18	73.67+	18	71.68	20	75.44+	7	73+	22
Mushroom	**100**	22	100	3	99,4-	3	98.5-	4	99-	4	99.97	5
kr-vs-kp	**99.43**	36	99.27	27	97.21-	27	94-	7	94.48-	7	99.06-	29
per/deg			**7+/0-**		5+/7-		6+/6-		5+/6-		3+/3-	

Les symboles "+" et "-" représentent une amélioration des performances (au niveau significatif de 0.02) dans le cas où un algorithme donne de meilleures performances que l'algorithme d'apprentissage avec l'ensemble complet de données.

Nous considérons le meilleur résultat, le compromis entre la réduction de variables contre une grande performance en classification ainsi obtenue.

Tableau 4-7. Classification de IB_5 avec et sans filtrage de données

Bases de données	IB_5		IB_5+ STRASS		IB_5+ mRMR		IB_5+ CFS		IB_5+ FCBF		IB_5+ INTERACT	
	Tc	Toutes	Tc	S	Tc	S	Tc	S	Tc	S	Tc	S
Heart	**83**	13	82.9	8	84.5+	8	82.5-	6	81.9-	5	82.9	9
German	**73.1**	24	73.2	10	72.4-	10	73.6+	5	73.6+	5	73.7+	13
Pima	**76.56**	8	76.56	6	75.5-	6	76.45	3	76.56	6	76.56	6
Iris	**94.6**	4	94.66	3	94.5	3	96+	2	95.8+	2	94.66	3
Ionosphere	**89.4**	34	89.63+	6	88-	6	88.2-	12	86.8-	5	89.17-	8
Vehicle	**70.56**	18	72.63+	15	69.14-	15	66.94-	9	61-	4	70.21-	17
Glass	**69.6**	9	68.6-	6	67.75-	6	65.87-	7	66.6-	5	68.6-	6
Wine	**91**	13	93.82+	5	90.44-	5	93.4+	8	93+	10	93.25+	5
Sonar	**83.17**	60	85.57+	12	83.65+	12	84.61+	19	81.73-	10	79.80-	12
Austra	**85.5**	14	85.8+	13	84.63-	13	85.5	7	85.4	7	86+	13
segment	**91.94**	19	90.82-	4	90.82-	4	92.42+	8	91.99	6	92.16+	9
arrhythmia	**69.46**	279	69.48	18	68.58-	18	71.68+	20	70.79+	7	67.47-	22
Mushroom	**99.9**	22	99. 8	3	99.4-	3	98.52-	4	98.9-	4	99.97	5
kr-vs-kp	**96.37**	36	97.3+	27	94.49-	27	94.1-	7	94.1-	7	95.58-	29
per/deg			**6+/2-**		2+/11-		6+/6-		4+/7-		4+/6-	

Si l'on compare STRASS avec les autres algorithmes de sélection de variables, STRASS obtient les meilleurs compromis avec les arbres de décision (C4.5) tout en réduisant le nombre de variables sans toutefois, altérer le taux de classification voir même l'augmenter sur 7 bases des 14 bases de données filtrées par notre algorithme. Avec IB_5, STRASS et

CFS obtiennent six meilleurs compromis contre quatre FCBF et INTERACT et deux pour mRMR.

Tableau 4-8. Classification de MLP avec et sans filtrage de données

Bases de données	MLP		MLP+ STRASS		MLP + mRMR		MLP + CFS		MLP + FCBF		MLP + INTERACT	
	Tc	Toutes	Tc	S	Tc	S	Tc	S	Tc	S	Tc	S
Heart	**80.43**	13	83.12+	8	82.22+	8	82.61+	6	84.07+	5	81.11+	9
German	**71**	24	71.3+	10	71.7+	10	76.2+	5	76.2+	5	72.2+	13
Pima	**76.17**	8	77.34+	6	75.91-	6	76.56+	3	77.34+	6	77.34+	6
Iris	**93.33**	4	96+	3	95.33+	3	95.33+	2	95.33+	2	95.33+	3
Ionosphere	**90.75**	34	85.71-	6	86.55-	6	88.23-	12	89.07-	5	91.59+	8
Vehicle	**66.31**	18	69.44+	15	66.31	15	67.01+	9	52.43-	4	82.60+	17
Glass	**77.57**	9	79.43+	6	77.10-	6	79.43+	7	79.43+	7	79.43+	6
Wine	**98.31**	13	97.19-	5	95.50-	5	98.31	8	96.87-	10	94.94-	5
Sonar	**78.87**	60	78.87	12	76.05-	12	77.46-	19	71.83-	10	73.23-	12
Austra	**82.97**	14	82.97	13	80-	13	82.97	7	82.97	7	83.82+	13
segment	**96.81**	19	96.05-	4	97.07+	4	96.94	8	96.30-	6	96.43-	9
arrhythmia	**72.35**	279	73.37+	18	73.37+	18	78.57+	20	73.37+	7	74.02+	22
Mushroom	**100**	22	99.81	3	99.53-	3	98.84-	1	99.56-	4	100	5
kr-vs-kp	**99.44**	36	99.26	27	96.32-	27	93.74-	7	93.74-	7	99.17-	29
per/deg			7+/3-		5+/8-		7+/4-		6+/7-		**9+/4-**	

En ce qui concerne les réseaux de neurones (MLP), l'algorithme INTERACT donne les meilleures performances en classification sur 9 bases de données, suivi par STRASS et CFS sur 7 bases et FCBF sur 6 bases.

D'après les résultats obtenus, on ne peut que constater les bonnes performances en classification sur l'ensemble des bases de données après filtrage de données par notre algorithme, Relief, mRMR, CFS, FCBF et INTERACT.

4.4 Discussion

Les critères de sélection de variables utilisés par STRASS, nous permettent d'explorer trois aspects de la corrélation des données :

1. Le premier est la corrélation d'une variable sur un ensemble d'objets c'est à dire la capacité d'une variable à discriminer une partie de la population étudiée.
2. Le deuxième est la corrélation partielle d'une variable par rapport à un ensemble de variables c'est le cas ou plusieurs variables prises ensemble jouent le même rôle que la variable considérée (elles discriminent les objets de la population étudiée par la variable considérée). Cette propriété permet soit de détecter la pertinence forte soit la redondance partielle d'une variable.
3. Le troisième aspect est la capacité d'une variable une fois combinée à d'autres variables à discriminer un ensemble de la population. Il s'agit dans ce cas de la pertinence faible.

Dans les différents travaux de sélection de variables que nous avons étudié, les algorithmes sont répertorié contextuels suivant un ou deux aspects de la corrélation cité ci-dessus, à notre connaissance les critères et l'algorithme que nous proposons sont les premiers à considérer les différents aspects de la corrélation. En effet, en examinant les résultats obtenus des différents algorithmes de sélection de variables sur les données synthétiques et réelles, il se trouve que :

Relief : utilise une représentation par paires des données et pondère la pertinence d'une variable indépendamment des autres variables, de ce fait Relief traite le premier aspect de la corrélation qui est la corrélation d'une variable par rapport à un échantillon de données.

ConcinstancyGA : utilise un critère de consistance pour la sélection de variable.

WrapperGA(C4.5) et WrapperGA(IB_1) : utilisent la précision de C4.5 et les k-plus proches voisins pour la sélection d'un sous ensemble optimal de variables, en plus les méthodes enveloppes sont très couteuses en temps de calcul.

GainRation : calcul le gain d'information pour chaque attribut indépendamment des autres et donne une liste ordonnée des variables.

CFS : calcul le mérite d'un ensemble de variables donc détecte les meilleures combinaisons de variables (la pertinence faible). Il calcul aussi la redondance sur des paires de variables, il s'agit de la redondance totale d'une variable par rapport à une autre et non pas la redondance partielle (corrélation partielle) d'une variable par rapport à un ensemble de variables.

FCBF : détermine la pertinence forte d'une variable par rapport à un ensemble de variables en considérant les variables à examiner par paire ($2^{ème}$ degré d'interactions). FCBF ne peut donc pas considérer la pertinence d'une variable par rapport à un ensemble de variables ($k^{ème}$ degré d'interactions).

INTERACT : combine deux critères de sélection, information et consistance, ce qui lui permet de considérer le $k^{ème}$ degré d'interaction. Cependant, les deux critères considèrent la pertinence faible, par conséquent INTERACT n'arrive pas à déterminer les variables prédominantes à pertinence forte.

Bien que certains algorithmes traitent d'une manière ou d'une autre les différents aspects de la corrélation entre variables, aucun d'entre eux ne permet de déceler ce que l'on a nommé la redondance contextuelle ou partielle, cette dernière est la redondance d'une variable par rapport à un ensemble de variables.

4.5 Grandes bases de données : Etude de la stabilité

Afin de pouvoir rendre notre algorithme compatible aux très grandes bases de données, nous proposons de tester sa stabilité par rapport à différents échantillons de données. Pour cela on propose d'appliquer un échantillonnage aléatoire en respectant les proportions des distributions des classes. Ce type d'échantillonnage est souvent utilisé pour les grandes bases de données en fouille de données, ce dernier a prouvé sont efficacité pour différents

algorithmes de sélection de données et d'apprentissage (Skalak, 1994), (Liu, et al., 2004), (Yang, et al., 2006). Pour ce faire, le point le plus important reste à déterminer la taille approprié de l'échantillon afin de maintenir un taux de classification acceptable. Nous avons effectué différents test avec cinq échantillon de n sur l'ensemble total de données N, tel que chaque échantillon $\binom{N}{n}$ distinct de données est équiprobablement choisi. STRASS est appliqué sur différents pourcentages P_i de données afin de sélectionner les variables correspondantes. Pour chaque échantillon de données, on obtient donc différents ensembles de données filtrés par rapport aux variables sélectionnées. Nous avons aussi jugé bon de comparer la stabilité de STRASSS par rapport à Relief, un algorithme conventionnel de sélection de variable qui se base lui-même sur l'échantillonnage de données afin de sélectionner les variable ainsi que les algorithmes CFS et FCBF car ces derniers ont été spécialement conçu pour traiter les grandes bases de données. Pour évaluer l'impact de la sélection de variables sur les algorithmes d'induction nous avons utilisé l'ensemble total des exemples mais en retenant que les variables sélectionnées par les algorithmes de filtrage en utilisant les portions d'exemples réduites des données. Pour cela on a testé la stabilité de notre algorithme de filtrage avec cinq pourcentages d'échantillonnage P_i (10%, 30%, 50%, 80% et 100% de l'ensemble total de données). Le résultat de cette étude est donné dans l'annexe B.

Les résultats obtenus en prenant des échantillons sur les données montrent que STRASS améliore ou bien maintient le taux de classification de C4.5 et IB_5 sur la plus part des ensembles de données échantillonnées. STRASS est stable dans la sélection du sous ensemble pertinent de variables pour différentes portions de données quand un échantillonnage est appliqué. L'algorithme a aussi donné les meilleurs résultats pour les plus petits échantillons de données. Par conséquent, nous pouvons proposer l'utilisation de STRASS sur les très grandes bases de données.

4.6 Analyse des résultats

Les résultats obtenus sur les différentes bases (réelles et artificielles) montrent que le filtrage de données permet non seulement de réduire les données, mais aussi de rendre les algorithmes d'induction plus performants. Nous avons aussi constaté que l'algorithme de filtrage n'ai pas indépendant de l'algorithme d'induction auquel il est associé :

- La construction des arbres de décision privilégie les variables indépendantes et la propagation de ces derniers ajoute au fur et à mesure des variables (les variables partiellement corrélées) puisque c'est la combinaison des variables qui permet de discriminer entre les classes. Donc les arbres de décision utilisent les deux aspects de la pertinence (forte et faible).

- Les réseaux de neurones perceptron multicouches et les k plus proche voisins sont des classificateurs qui utilisent toutes les variables pour discriminer entre les classes.

STRASS est un algorithme qui traite les deux aspects de la pertinence de variables (faible et forte). Ce qui explique les bonnes performances de la combinaison de notre algorithme par rapport aux trois classifieurs (C4.5, MLP et IB_k) pour ce type de fonctions.

STRASS détecte aussi bien les variables indépendantes (prédominantes ou variables essentielles dites à pertinence forte) et les variables corrélées par morceaux (les variables restantes à pertinence faible) ainsi que les variables redondantes. Ceci nous a permis de faire une catégorisation de variables qui a été par la suite exploitée en diagnostic industriel afin de rendre un système de diagnostic de défauts plus fiable (Senoussi, et al., 2011(b)).

4.7 Application dans des cas réels en diagnostic

Tout en étudiant différents travaux en sélection de variables appliquée en diagnostic de défauts (voir chapitre 2), nous avons remarqué que les algorithmes de sélection de variables utilisés ne considèrent pas la corrélation partielle entre variables et généralement ces

algorithmes donnent une liste ordonnée de variables selon une évaluation individuelle de chaque attribut. Ces algorithmes avec une complexité linéaire, sont très intéressant du point de vu temps de calcul. Cependant, ces derniers (1) ne peuvent détecter les variables redondantes par parties et (2) ne peuvent faire une catégorisation de variables telle que notre algorithme STRASS effectue en même temps que la sélection de variables.

Afin de valider notre méthode en diagnostic, nous allons utiliser l'algorithme STRASS de deux différentes manières :

1. Sur un exemple réel en diagnostic de défauts : le procédé chimique *Tennessee Eastman Process* (TEP) (Senoussi et al., 2011(a)), (Senoussi et al., 2012).

2. Concevoir un système de diagnostic de défauts plus fiable en utilisant la catégorisation de variables que permet d'obtenir l'algorithme STRASS lors de son déroulement (Senoussi et al., 2011(b)).

4.7.1 Application sur le Processus TEP

Le processus *Tennessee Eastman Process* (TEP) est un procédé modélisé par la société *Eastman Chemical Company* afin de fournir une simulation d'un procédé industriel réel pour le test de méthodes d'asservissements et/ou de surveillance de procédé (Downs, et al., 1993), (Jockenhövel, et al., 2003). Le TEP a été très utilisé par la communauté de la surveillance des procédés afin de comparer certaines méthodes (Verron, et al., 2008), (Chiang, et al., 2004), (Nashalji, et al., 2009). Trois défauts sont généralement pris en considération : les défauts 4, 9 et 11 car les données présentent de grandes interactions. Notre choix pour la partie applicative s'est porté principalement sur ce processus car nous avons trouvé intéressant de voir l'impact de l'utilisation de notre algorithme de sélection de variables sur ces données fortement corrélées. En effet, STRASS a prouvé son efficacité à traiter des bases de données reconnues pour leurs grandes interactions (Senoussi, et al., 2008). En plus, les données sont disponibles en ligne à l'adresse suivante : http://brahms.scs.uiuc.edu.

Afin d'évaluer et de comparer les performances de l'algorithme de sélection de variables proposé, Nous avons repris les données utilisées dans les travaux de Chiang et al. (Chiang, et al., 2004), Verron (Verron, et al., 2008), (Verron, et al., 2006) et Nashalji, et al. (Nashalji, et al., 2009). Elles proviennent du TEP couplé à la structure d'asservissement de Lyman et Georgakis (Lyman, et al., 1995). Ce processus est présenté dans l'annexe C.

Tableau 4-9. Les défauts à grandes corrélation du TEP

Classe	Défaut
1	défaut 4: Saut de la température d'entrée du ref. liq. au réacteur
2	défaut 9: Variation aléatoire de la température d'alimentation en D
3	défaut 11: Variation aléatoire de la température d'entrée du ref. liq. au réacteur

Ces données se présentent ainsi (voir Tableau 4-9) : 480 observations d'apprentissage pour chaque type de faute et 800 observations de test pour chaque type de faute.

4.7.1.1 Sélection de variables du TEP

Le tableau (4-10) présente les variables sélectionnées par chaque algorithme de filtrage. Pour l'algorithme STRASS les variables les plus pertinentes suivant un ordre décroissant sont {51,41,38,40,37,50,9,18,19,20}. La figure (4-1) donne le DCG pour chaque variable.

Tableau 4-10. Variables sélectionnées par chaque algorithme de filtrage

	STRASS	mRMR	CFS	FCBF	INTERACT
TEP	**10** {51,41,38,40,37, 50, 9, 18,19,20}	**10** {9, 41, 18, 37, 39, 51,21, 40, 20,19}	**6** {9,18,21,37,39, 51}	**6** {9,18,21, 37,39, 51}	**8** {51,9,41,38,37,50, 40,19}

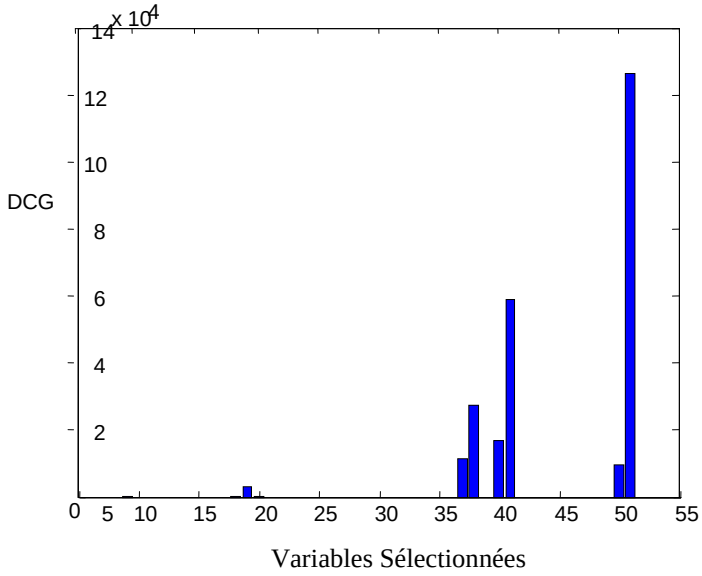

Figure 4-1 *: Gain du pouvoir discriminant but* (*DCG*) *associé à chaque variable*

4.7.1.2 Diagnostic

Sachant qu'un système de diagnostic de défauts se basant sur les méthodes de fouille de données nécessite l'utilisation de classifieurs pour la prise de décision, nous avons utilisé différentes méthodes d'induction, les arbres de décision grâce à l'algorithme C4.5, les k plus proches voisins (IB_k) et les réseaux de neurones multi-couches perceptron (MLP).

Les résultats de classification sont évalués en utilisant deux mesures de performances qui sont le taux de classification (T_C) et le Kappa statistique. Sachant que le taux de classification est obtenu en divisant le nombre d'exemples représentant réellement un

défaut sur le nombre d'exemples total. Dans le cadre multi-classes $T_C = \frac{\sum_{i=1}^{m} T_{Ci}}{m}$, avec m le nombre total de classes.

Le *Kappa* est une mesure de la différence entre l'accord constaté entre deux juges, et l'accord qui existerait si les juges classaient les exemples au hasard.

Dans Weka (Witten, et al., 2000), le jugement, c'est la classe d'un exemple, et les deux juges sont le classifieur et la classe réelle de l'exemple. Le calcul du kappa se fait de la manière suivante :

$$Kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{4.8}$$

Où Pr(a) est l'accord relatif entre codeurs et Pr(e) la probabilité d'un accord aléatoire. Cette mesure a été adaptée pour les résultats d'induction en utilisant la matrice de confusion de la manière suivante :

$$Kappa = \frac{n\sum_{i=1}^{M} x_{ii} - \sum_{i=1}^{M} x_{i.}x_{i.}}{n^2 - \sum_{i=1}^{M} x_{i.}x_{i.}} \tag{4.9}$$

Où x_{ii} représente le contenu de la diagonale c'est à dire le nombre d'exemples déduit appartenir à la classe i et appartenant réellement à la classe i, n le nombre total d'exemples, M nombre de classes, $x_{.i}$, $x_{i.}$ représentent le contenu des colonnes et lignes de la matrice de confusion.

A la différence du taux de classification, le Kappa de Cohen comptabilise les succès indépendamment pour chaque classe et les regroupe, par contre le taux de classification comptabilise tous les succès pour toutes les classes. Ce qui rend par conséquence le coefficient du kappa moins sensible aux résultats aléatoires causé par un certain nombre d'exemples dans chaque classe. Si *Kappa*< 0 il y a un désaccord, entre [0.0 0.20] accord

très faible, [0.21 0.40] accord faible, [0.41 0.60] accord modéré, [0.61 0.80] accord fort et [0.81 1.00] accord presque parfait.

Les résultats obtenus avec STRSS sont comparés avec ceux obtenus avec quatre algorithmes de sélection de variables, reconnus comme algorithmes performant par rapport au traitement de la corrélation et l'interaction entre variables. Ces algorithmes sont : mRMR[1], CFS[2], FCBF et INTERACT.

Tableau 4-11. Classification de C4.5 avec et sans filtrage

TEP_4.9.11	C4.5	C4.5+ STRASS	C4.5+ mRMR	C4.5+ CFS	C4.5+ FCBF	C4.5+ INTERACT
Tc	89.72	**95.25+**	93+	93.2+	93.2+	95.13+
Kappa	0.84	**0.92+**	0.89+	0.89+	0.89+	0.90+

Tableau 4-12. Classification de IB_k avec et sans filtrage

TEP_4.9.11	IB_1	IB_1+ STRASS	IB_1+ mRMR	IB_1+ CFS	IB_1+ FCBF	IB_1+ INTERACT
Tc	87.76	**99.2+**	96.18+	86.98-	86.98-	98.44+
Kappa	0.84	**0.99+**	0.94+	0.8-	0.8-	0.98+

Tableau 4-13. Classification des MLP avec et sans filtrage

TEP_4.9.11	MLP	MLP+ STRASS	MLP+ mRMR	MLP+ CFS	MLP+ FCBF	MLP+ INTERACT
Tc	84.8	85.35 +	**90.23+**	82.88-	82.88-	83.55-
Kappa	0.77	0.78+	**0.8+**	0.74-	0.74-	0.72-

Les tableaux (4-11) à (4-13) donnent les résultats en termes de taux de classification et Kappa. Les meilleures performances sont notées en gras. Les symboles "+" et "-" représentent respectivement une significative amélioration ou dégradation d'un algorithme d'induction par rapport aux autres algorithmes sur l'ensemble des données.

[1] Le nombre de variables à sélectionner pour mRMR est fixé au nombre de variables sélectionnées par STRASS.

[2] CFS avec une stratégie de recherche (*FS best first search*).

Nous pouvons aisément voir que la combinaison de STRASS aux algorithmes d'induction C4.5 et IB_k pour le diagnostic de défauts du processus TEP, donne les meilleures performances en classification aussi bien pour le taux de classification que le kappa. La sélection de variables par l'algorithme de filtrage mRMR donne de très bonnes performances de classification pour le classifieur MLP. Par contre, l'utilisation de STRASS comme méthode de sélection de variables avec le classifieur IB_1 donne les meilleurs résultats de diagnostic pour les trois défauts hautement corrélés du processus TEP.

Sachant que STRASS sélectionne une série de variables suivant l'ordre de pertinence de chacune en utilisant deux critères de pertinence DCG et DC et afin de voir l'influence de l'ordre des variables ainsi sélectionnées, nous avons jugé bon d'évaluer les performances de classification en utilisant différentes sous chaînes de variables en enlevant à chaque fois la variable jugée la moins pertinente de manière à créer un biais de classification.

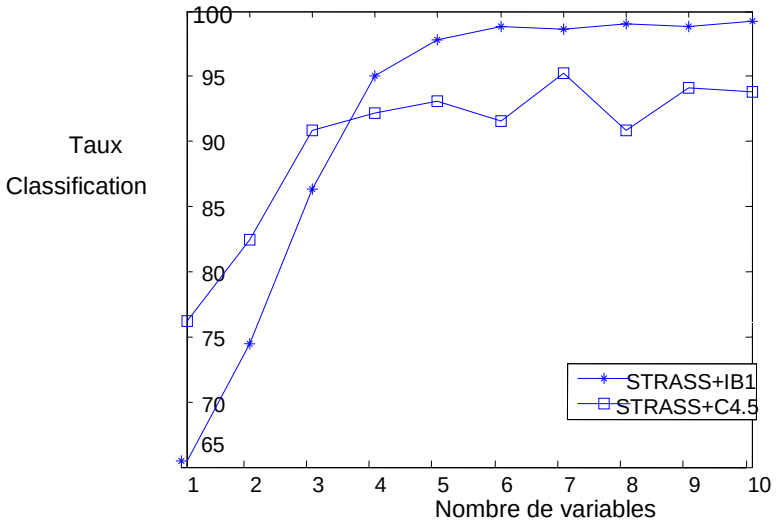

Figure 4-2 : *Résultats des classifications obtenues suivant l'ordre des variables sélectionnées par STRASS*

Les résultats obtenus en utilisant les arbres de décision (C4.5) et le 1 plus proches voisins (IB_1) sont illustrés par la figure (4-2). En examinent cette figure, nous pouvons aisément remarquer que dans le cas de C4.5, les sept premières variables sélectionnées sont non seulement suffisantes à la description du concept, mais améliorent considérablement

les résultats de classification (taux de classification et kappa). Par contre, en ce qui concerne IB_1 toutes les variables sélectionnées par STRASS sont nécessaires pour obtenir le meilleur taux de classification. Par conséquent, pour le reste de cette section le classifieur C4.5 sera combiné avec le sous ensemble des sept premières variables sélectionnées par STRASS et IB_1 avec l'ensemble total de variables sélectionnées par STRASS.

Toujours dans le but d'évaluer notre approche en diagnostic, nous avons cette fois ci comparé les résultats les plus concluants issus de la combinaison de STRASS avec C4.5 et IB_1 par rapport à d'autres travaux qui s'inscrivent dans le même cadre sur le TEP. Pour ce faire, nous avons repris les résultats de Chiang, et al. (Chiang, et al., 2004) et ceux de Verron, et al. (Verron, et al., 2006), (Verron, et al., 2008).

Afin de travailler dans les mêmes conditions nous avons utilisé les mêmes proportions de données que ce soit pour la base d'apprentissage ou bien la base de test. Ces dernières se composent de 1440 exemples d'apprentissage (480 exemples pour chaque défaut) et 2400 exemples de test (800 exemples pour chaque défaut).

Les méthodes utilisées dans Verron, et al. (Verron et al., 2008) sont l'Analyse discriminante linéaire (LDA : *Linear Discriminant Analysis*) et l'Analyse discriminante quadratique (QDA : *Quadratic Discriminant Analysis*). Les deux algorithmes d'induction LDA et QDA sont combinés à un algorithme de sélection de variable : l'algorithme de sélection multi-variables avec gain d'information (*multivariate with information gain*). Les variables sélectionnées par ces méthodes sont données par le tableau (4-14). Les variables 9 et 51 semblent être les plus informatives et se retrouvent en tête de l'ensemble sélectionné par chaque algorithme de filtrage.

Les méthodes utilisées dans Chiang, et al. (Chiang, et al., 2004) sont les Machines à vecteurs de support SVM (*Support Vector Machines*), les Machines à vecteurs de support Proximaux (PSVM : *Proximal Support Vector Machines*) et Machine à vecteurs de support Indépendants (ISVM : *Independent Support Vector Machines*).

Tableau 4-14. Ordre des 8 premières variables sélectionnées par les approches citées dans les travaux de (Verron, et al., 2008)

Approche	Variables ordonnées
Univariate	51, 9, 50, 19, 18, 20, 38, 21
Multivariate (LDA)	51, 9, 19, 29, 16, 20, 18, 17
Multivariate (QDA)	51, 9, 21, 50, 20, 38, 7, 18

Les taux de classification obtenus par les différentes méthodes ainsi que la combinaison de STRASS avec C4.5 et IB_1 sont groupés dans le tableau (4-15). Le tableau (4-16) donne la matrice de confusion de l'algorithme de sélection variables multi-variables avec gain d'information (*multivariate with information gain*) associé à l'algorithme d'induction analyse discriminante quadratique **(**QDA) (Verron, et al., 2008).

Tableau 4-15. Taux de classification obtenu par les différentes approches citées dans (Chiang, et al., 2004), (Verron, et al., 2008) et STRASS

Méthode/ variables	toutes	{51,9}	{51,9,21}	{51,41,38,40,37,50,9,18,19,20}	{51,41,38,40,37,50,9}
SVM	56%	93.5%			
PSVM	65%	94.0%			
ISVM	70.14%	94.0%			
LDA	57.96%	68.42%	67.13%		
QDA	81.17%	94.13%	94.35%		
C4.5	89.72%				
IBk	87.76%				
STRASS+C4.5					**95.5%**
STRASS+IB1				**99.2%**	

Les tableaux (4-17) et (4-18) donnent les matrices de confusions des résultats obtenus avec la combinaison des meilleurs sous-ensembles sélectionnés par STRASS associés aux arbres de décision et le 1-plus proche voisin.

Tableau 4-16. Matrice de confusion des données du processus TEP ({9,21,51}) en utilisant l'approche QDA+multi-variables (Verron, et al., 2008)

Class	défaut 4	défaut 9	défaut 11
défaut 4	794	0	34
défaut 9	0	777	73
défaut 11	6	23	693
Total	800	800	800

Tableau 4-17. Matrice de confusion de STRASS+IB_1 sur les données du processus TEP ({51,41,38,40,37,50,9,18,19,20})

Classes	défaut 4	défaut 9	défaut 11
défaut 4	794	0	6
défaut 9	0	800	10
défaut 11	6	0	784
Total	800	800	800

Tableau 4-18. Matrice de confusion de STRASS+C4.5 sur les données du processus TEP en utilisant les sept première variables sélectionnées par STRASS ({51,41,38,40,37,50,9})

Classes	défaut 4	défaut 9	défaut 11
défaut 4	782	1	24
défaut 9	6	776	45
défaut 11	12	23	731
Total	800	800	800

Sur 800 observations testées du défaut 9, le classifieur IB_1 donne 0% (0/800) d'erreur de classification. On peut aisément constater que les défauts 4, 9 et 11 sont bien discriminés ce qui n'est pas le cas pour l'approche QDA multivariables. D'après le tableau (4-16), les défauts 4 et 9 sont bien discriminés. Toutefois, le défaut 11 est moins discriminé que les deux autres parce que ce défaut est très corrélé aux deux autres défauts.

Par rapport aux autres approches, les combinaisons STRASS + IB_1 (avec l'ensemble complet de variables sélectionnées) et STRASS + C4.5 (avec les sept premières variables sélectionnées) permettent d'atteindre les meilleurs résultats pour les trois types de défauts.

Il peut être également observé que la combinaison STRASS + C4.5 surpasse les résultats obtenus lors de l'utilisation de C4.5 seul. Bien que C4.5 est un classifieur de données

qualitatives et que ce dernier effectue en même temps la sélection des variables. Cela s'explique par le fait que C4.5 effectue une sélection myope des variables en utilisant le gain d'information comme critère de sélection.

A partir de ces résultants nous pouvons aisément remarquer que l'utilisation de STRASS permet de décorréler les données, ceci est illustré par les résultats des matrices de confusions de STRASS+C4.5 et STRASS+IB_1. En effet, STRASS+IB_1 est la seule approche qui a pu détecter tous les exemples représentants le défaut 9. Les défauts (4,9) et 11 sont très bien discriminés. La méthode QDA+multi-variables arrive à bien séparer les défauts 4 et 9, cependant avec ces 107 exemples mal classés (34 attribué au défaut 4 et 73 au défaut 9), le défaut 9 est moins discriminé et reste partiellement corrélé aux autres défauts.

4.7.2 Catégorisation de variables pour la conception d'un système de diagnostic de défauts plus fiable.

La fiabilité d'un système de détection de défauts nécessite la minimisation des fausses alarmes ainsi que les erreurs de détection de défauts. Ces anomalies dans le système de détection sont généralement dues à des mesures capteurs erronées ou bien à une propagation du défaut. En pratique, ce genre de problématique est résolu par la redondance des capteurs. Par conséquent, un choix approprié des capteurs ainsi que de leur emplacement dans la chaine de mesure du système de supervision s'avère nécessaire. Deux cas de figures peuvent se présenter (Paljak, et al., 2009) :

Sur-instrumentation : la surveillance de nombreux paramètres d'un système pose des problèmes importants, tel que l'estimation d'un grand nombre de seuils et l'élaboration des règles d'identification et de diagnostic en particulier quand les systèmes sont évolutifs. D'autre part, trop de données mesurées provoquent également d'inutiles dépenses en capteurs et dégradent les performances du système de surveillance, en particulier en cas de collecte de données historiques.

Sous-instrumentation : Une réduction inadaptée de l'ensemble des mesures capteurs peut sérieusement compromettre les performances de la surveillance, tel que le temps de réaction à une erreur ou la détection et le diagnostic d'une panne.

La sélection d'un ensemble compact et suffisant de variables de contrôle est un problème clef pour la complexité de conception d'un système de diagnostic et supervision ainsi que son temps de réponse.

Dans le cadre de nos travaux de thèse nous avons proposé de concevoir un système de détection plus fiable en utilisant la catégorisation de variables que nous permet d'obtenir notre algorithme de sélection de variables STRASS (Senoussi, et al., 2011(b)). En effet, la catégorisation de variables nous a conduit à développer une mesure de confiance pratique pour le système de détection de défaut. Le système proposé est illustré par la figure (4-4). En se référent à cette figure :

1. Dans la première phase, la conception d'un classifieur pour la détection de défauts est effectuée en utilisant toutes les variables prédominantes (SR), les variables à pertinence faible mais non-redondantes (WRnr) et les variable à pertinence faible (WRr1).

2. Dans un deuxième temps, les variables redondantes sont utilisées en combinaison avec les variables prédominantes ainsi que les variables faiblement pertinentes mais non redondantes (WRnr) pour la conception d'un deuxième classifieur.

3. En cas de résultats similaires, le second classifieur servira donc à confirmer les résultats obtenus avec le premier système de détection de défauts (classifieur). Si les résultats obtenus avec le second classifieur sont différents par rapport aux résultats du premier, cela indiquera un problème dans la plateforme d'acquisition (un capteur défaillant) ou bien d'une donnée contexte (une valeur ou un paramètre erroné). L'identification des variables est déterminée par un examen minutieux des variables redondantes, ou bien l'acquisition de nouvelles données.

La catégorisation de variables, nous permet d'une part de faire un choix plus approprié pour les capteurs à mettre en redondance et cela en considérant les variables prédominantes. En effet, puisque ces dernières sont les seules à discriminer certains concepts (donc à identifier un défaut), il faut impérativement mettre les capteurs donnant les mesures qui représentent ces variables en redondance. D'une autre part, les variables redondantes quant à elles, sont utilisées afin de vérifier l'information et la fiabiliser. Le système de détection de défauts ainsi conçu permettra une diminution considérable des erreurs de détection ainsi que des fausses alarmes.

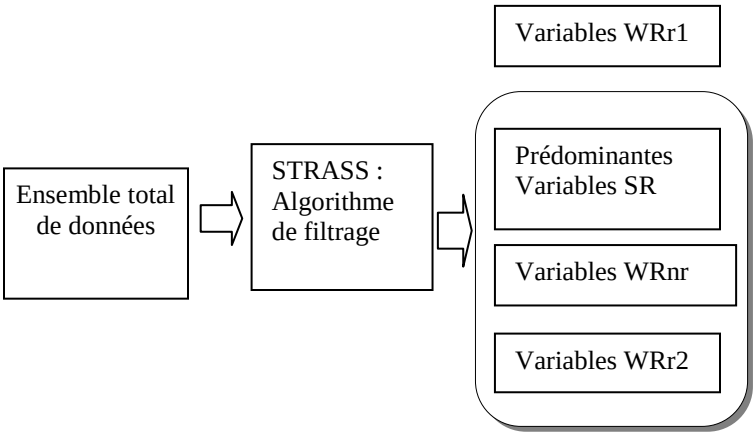

Figure 4-3 : *Catégorisation des variables*

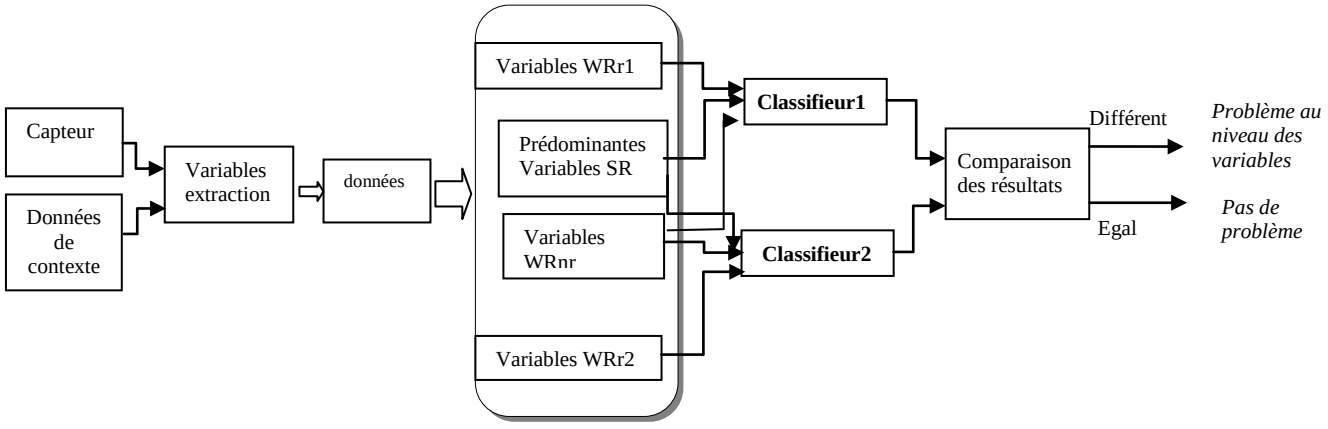

Figure 4-4 : *Conception d'un système de détection de défauts fiable en utilisant les variables redondantes déterminées par la catégorisation de variable* (Senoussi, et al., 2011(b))

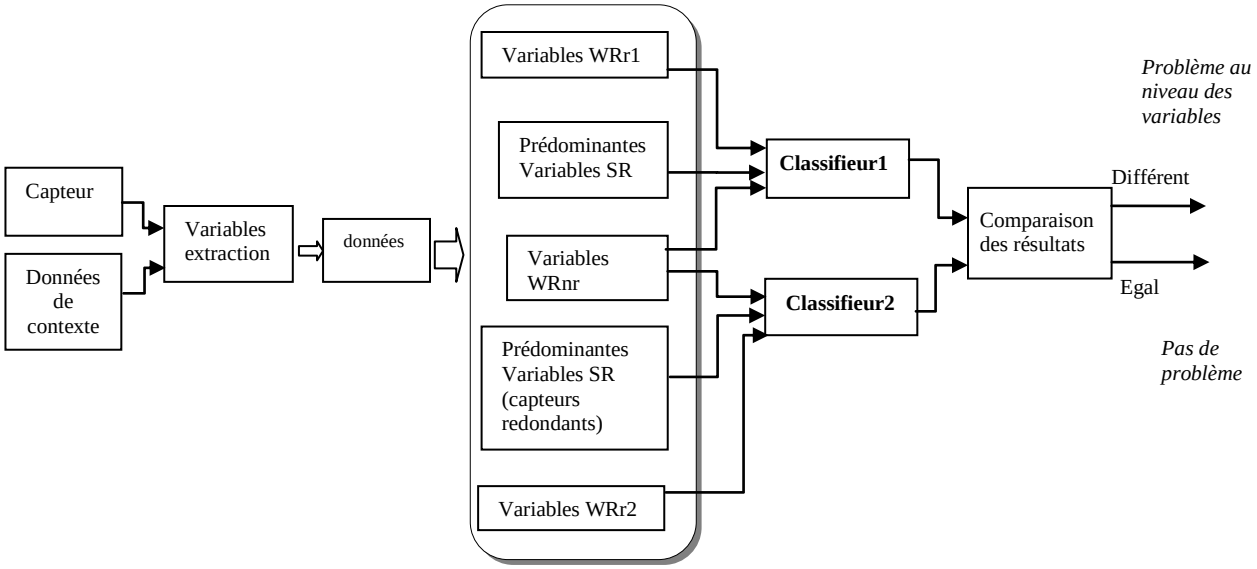

Figure 4-5: *Conception d'un système de détection de défauts plus fiable en utilisant les variables redondantes et en ajoutant une redondance de capteurs sur les variables à partir de la catégorisation de variable*

4.7.2.1 Sélection et catégorisation de variables

La méthodologie proposée a été évaluée sur des bases de données médicales (Heart, Lung cancer, Hepatitis) disponible en ligne à partir du site UCI Machine Learning à l'adresse suivante : http://kdd.ics.uci.edu, ainsi que sur deux bases de données en diagnostic de défauts (Machine, RFM) de Texas Instruments projet SEMATECH J-88 pour plus d'information voir les références (Barna, 1997) et (Barry, et al., 1999). Les caractéristiques des données sont illustrées dans le tableau (4-19).

Tableau 4-19. Description des bases de données

Données	Instances	Variables	Classes
Heart	270	14	2
Lung cancer	32	57	2
Hepatitis	20	20	2
Machine	12829	22	22
RFM	3519	72	22

Afin d'évaluer cette fois les résultats obtenus avec STRASS, nous l'avons comparé à CFS et FCBF (Senoussi et al., 2011(b)). Le tableau (4-20) présente les résultats de la sélection de variables pour chaque algorithme de filtrage en comparaison avec ceux obtenus pour l'algorithme STRASS. STRASS donne les meilleures performances en termes de nombre de variables sélectionnées (18.33% de variables sélectionnées sur l'ensemble total de variables).

Le tableau (4-21) donne les résultats de la sélection de variables par STRASS ainsi que la catégorisation qui leur a été attribuée. Nous pouvons aisément remarquer que les bases de données Heart et Hepatitis sont caractérisées par deux types de variables, les variables prédominantes (à pertinence forte) et les variables redondantes, ce qui peut permettre de construire un second classifieur afin de valider les résultats d'un premier classifieur qui sera construit autour des variables prédominantes et les variables restantes à pertinence faible.

Tableau 4-20. Nombre de variables sélectionnées par chaque algorithme de filtrage

Données	Toutes les variables	STRASS	CFS	FCBF
Heart	13	8	6	5
Lcancer	56	3	8	6
Hepatitis	19	9	9	6
Machine	21	5	10	8
RFM	71	8	18	11
Moyenne	36	6.6	10.2	7.2

Pour les bases de données Lung cancer et RFM, ces derrières ne sont caractérisées que par des variables prédominantes ou bien des variables inutiles que la phase de sélection a supprimé de l'ensemble de variables sélectionnées. Cependant, dans ce cas de figure, il serait judicieux de mettre leurs capteurs (qui donnent les mesures représentants ces variables) en double c'est-à-dire opter pour un choix de redondance pour ces mesures : dans

le cas où un capteur est défaillant, le capteur redondant le remplacera afin d'une part améliorer la détection et d'autre part la fiabiliser.

Tableau 4-21 . Catégorisation des variables par STRASS

Données	STRASS Variables Sélectionnées	SR	WRnr	WRr1=WRr2
Heart	8{3,7,8,1,2,12,9,13}	{3,7,8,1,2,12}	{9,13}	9=4=5=6 13=11
Lang cancer	3 {9,43,34}	{9,43,34}		
Hepatitis	9{11,18,17,6,14,8,12,3,2}	{11,18}	{17,6,14,8,12,2}	3=7=10
Machine	5 {1,3,7,17,13}	{3,17, 13}	{1,7}	{7=11,12,14,16}
RFM	8 {35,26, 22,14, 44,66,9,4}	{35, 26, 22,14, 44,66,9,4}		

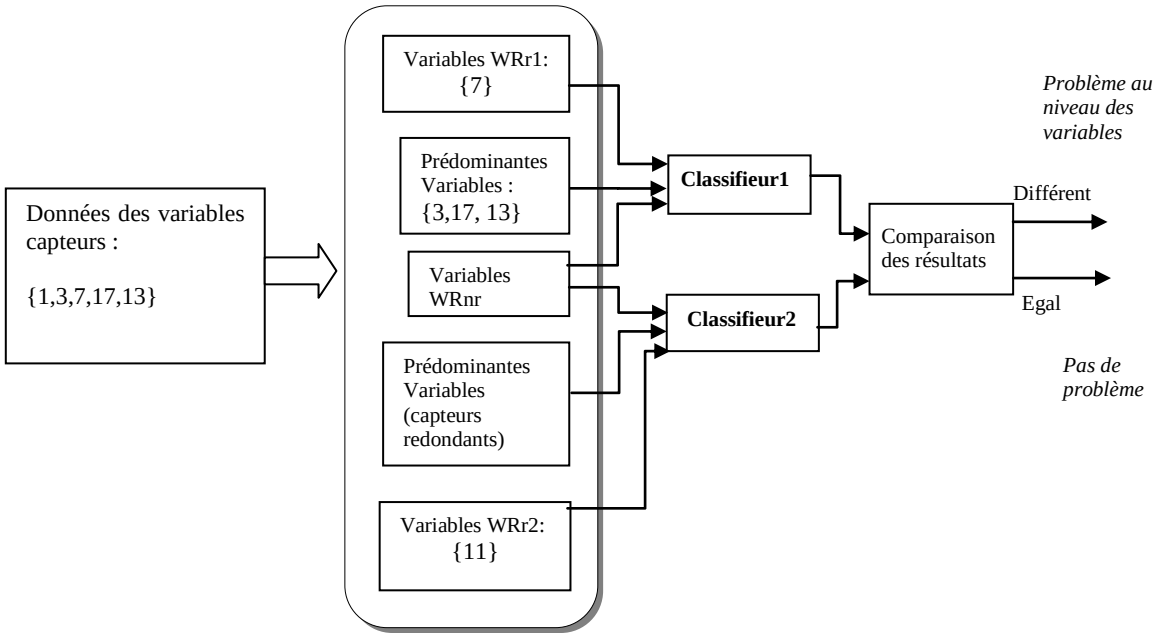

Figure 4-6 : *Exemple de conception d'un système de détection de défauts plus fiable avec la base de données Machine*

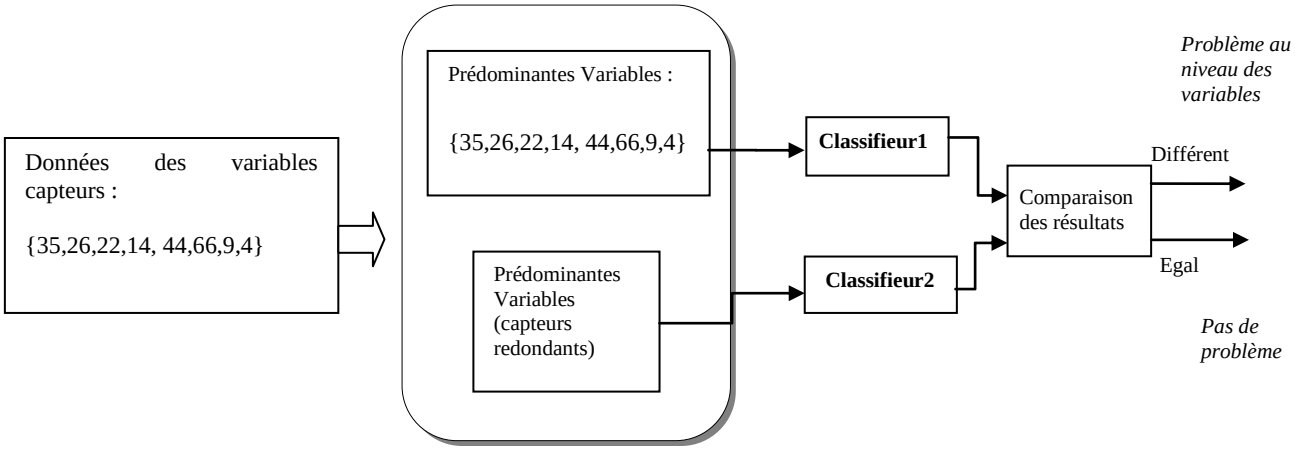

Figure 4-7 : *Exemple de conception d'un système de détection de défauts plus fiable avec la base de données RFM*

4.7.2.2 Diagnostic

Pour la classification des bases de données en diagnostic médical et en diagnostic de défauts, nous allons présenter dans cette section ceux ayant donné les meilleures performances à savoir les arbres de décision grâce à l'algorithme C4.5, les réseaux de neurones (MLP) et les k plus proches voisins (IB_k). Les résultats de classification sont obtenus avec une 10-validations croisées (10-fold cross-validation). Les symboles "+" et "-" représentent respectivement une significative amélioration ou dégradation d'un algorithme d'induction par rapport aux autres algorithmes sur l'ensemble des données.

Tableau 4-22. Classification de C4.5 avec et sans filtrage

Données	C4.5	C4.5+ STRASS	C4.5+ CFS	C4.5+ FCBF
Heart	83.7	85.18+	83.3-	84.4 +
Lang cancer	78.12	84.35 +	78.21	85.5+
Hepatitis	81.3	81.3	81.91+	80.6 -
Machine	94.58	94.72+	94.81+	94.70+
RFM	94.38	95.34+	94.07-	94.13-
Moyenne	86.41	**88.17+**	86.4	87.86+
per/deg		**4+/0-**	2+/2-	3+/2-

Tableau 4-23. Classification de IB_1 avec et sans filtrage

Données	IB_1	IB_1+ STRASS	IB_1+ CFS	IB_1+ FCBF
Heart	83.2	82.5 -	82.5-	81.9 -
Lang cancer	75	78.5 +	71.3-	71.8-
Hepatitis	83.8	85.8+	77.38-	84.5+
Machine	95.80	95.97+	93.3-	94.95-
RFM	94.65	96.06+	95.84+	94.67
Moyenne	86.49	**87.76+**	84.06-	85.56-
per/deg		**4+/1-**	1+/4-	1+/3-

Tableau 4-24. Classification de MLP avec et sans filtrage

Données	MLP	MLP+ STRASS	MLP+ CFS	MLP+ FCBF
Heart	80.43	83.12+	82.61+	84.07+
Lang cancer	67.9	86.67+	85.42+	79.58+
Hepatitis	84.23	85.21+	84.46+	85.24+
Machine	79.28	59.87-	50.74-	58.90-
RFM	90.51	90.5	89.87-	89.61-
Moyenne	80.47	**81.07+**	78.62-	79.47-
per/deg		**3+/1-**	3+/2-	3+/2-

Les résultats de classification (diagnostic) montrent l'apport de la sélection de variables sur les performances des algorithmes d'induction. En effet malgré la réduction des données dans la plus part des tests, le taux ce classification est soit supérieur ou pratiquement comparable à celui obtenu avec l'ensemble total des données. Cependant, en comparent les performances de chaque algorithme de sélection de variables par rapport à chaque algorithme d'induction, nous pouvons aisément conclure que STRASS donnent le meilleurs compromis taux de sélection (réduction des données) par rapport au taux de classification.

L'application de notre méthodologie sur les données Machine et RFM, démontre que cette approche est très efficace pour la sélection de variables pertinentes en diagnostic de défauts.

4.7.3 Analyse des résultats en diagnostic de défauts

Les résultats de diagnostic sur l'ensemble de données TEP montrent que la combinaison de STRASS avec les classifieurs 1-plus proche voisin (IB_1) et C4.5 surpasse les résultats obtenus avec LDA, QDA multi-variables (Verron, et al., 2008) et ceux obtenus avec les SVM (*Support Vector Machines*), PSVM (*proximales Support Vector Machines*) et ISVM (*independant support Vector Machines*) (Chiang, et al., 2004). Les auteurs mettent l'accent uniquement sur trois types de défauts, parce que les trois classes se chevauchent dans l'espace multidimensionnel, car il y'a une grande interaction des données. Par conséquent, ces données sont appropriées pour être traitées avec les algorithmes contextuels de sélection de variables notamment l'algorithme STRASS. Une remarque intéressante est le fait que STRASS est un algorithme apparenté aux méthodes 'filtre' de sélection de données, par contre LDA, QDA multi-variables sont des méthodes 'enveloppe'. Les méthodes enveloppe sont connues pour être plus coûteuse en temps de calcul par rapport aux méthodes filtre.

4.8 Conclusion

Les résultats obtenus dans ce chapitre montrent que le filtrage de données permet non seulement de réduire les données, mais aussi de rendre les algorithmes d'induction plus performants. Nous avons proposé l'algorithme STRASS pour traiter les grandes interactions dans les données. Ce dernier donne des performances satisfaisantes quant à la sélection d'un ensemble minimale de variables pertinentes. De nombreux tests de la méthode proposée en sélection de variables ont été effectués sur des données artificielles et des données réelles appartenant à diverses applications.

Afin de rendre STRASS exploitable pour les très grandes bases de données, nous avons étudié sa stabilité en appliquant un échantillonnage des données. Les résultats obtenus sont encouragent quand à l'efficacité de notre algorithme par rapport à des portions de données et le rendent par conséquence compatible à traiter les très grandes bases de données. En effet le challenge actuel dans la communauté sélection de variables est aussi bien la corrélation des données que le grands flux de données.

Nous avons aussi montré que l'algorithme de sélection de variables STRASS réalise une catégorisation intéressante de variables, cette dernière nous a permis de proposer une conception d'un système de diagnostic de défauts plus fiable.

STRASS donne aussi les meilleures performances en diagnostic en comparaison aux autres algorithmes contextuels de sélection de variables tels que : mRMR, CFS, FCBF, INTERACT et ce, sur différentes bases de données aussi bien en diagnostic médicale qu'en diagnostic de défauts. Ce résultat n'est pas surprenant car STRASS est un algorithme de sélection de variables qui traite les grandes interactions entre variables. Cet algorithme a prouvé son efficacité sur des bases de données fortement corrélées (Senoussi, et al., 2008).

Conclusion générale

Nous nous sommes intéressés dans cette thèse à l'application des outils d'Extraction de Connaissance à partir de Données dans le cadre du diagnostic industriel. Nous considérons dans notre approche qu'il est possible, par un apprentissage, de capitaliser les connaissances de scénario en maintenance et notamment en diagnostic de défauts afin de proposer un système de diagnostic de défaillances plus performant et fiable.

C'est pourquoi nous avons établi un parallèle entre le problème de maintenance industrielle et le processus d'ECD. Dans ce contexte nous avons d'une part développé des travaux de recherche dans le domaine de l'apprentissage à travers des algorithmes d'induction tel que, les réseaux de neurones, les *k* plus proches voisins et les arbres de décision et d'autre part proposé un nouveau algorithme contextuel de sélection de variables pertinentes.

L'algorithme de sélection de variables proposé (STRASS) est une approche originale face au problème induit par la corrélation des données. La première phase de cet algorithme permet de sélectionner les variables fortement pertinentes pour la discrimination des concepts à apprendre. La seconde phase permet de sélectionner les variables à pertinence faible. Dans la dernière phase les variables redondantes sont éliminées. Nous utilisant pour cela des critères de sélection contextuels, c'est-à-dire des critères qui évaluent la pertinence d'une variable dans le contexte des autres variables.

Afin d'analyser les performances de sélection de données de l'algorithme STRASS, nous avons réalisé une évaluation empirique, aussi bien directe que indirecte, sur divers ensembles d'apprentissage de référence. Les résultats obtenus nous montrent que STRASS est efficace lorsqu'il est associé aux réseaux de neurones, les *k* plus proches voisins et les arbres de décision. En effet, la sélection de variables permet de réduire sensiblement le nombre de données nécessaires à la construction du classifieur sans toutefois introduire de perte notable au niveau de ses performances en classification voir

même les augmenter. Afin de rendre STRASS plus adapté aux très grandes bases de données, nous avons jugé bon de l'appliquer sur différents échantillons sélectionnés aléatoirement à partir d'une base de données. Nous avons conduit une étude sur huit bases de données de différentes dimensions. Les résultats obtenus nous ont permis de conclure non seulement que STRASS est très stable en sélection de variables, mais aussi que ce dernier est adapté pour traiter les grandes bases de données.

Nous avons appliqué notre méthodologie à la problématique du diagnostic. Dans le cadre de ce travail nous avons considéré un système de diagnostic comme un classificateur de données où les classes correspondent aux défauts ou au fonctionnement normal. Nous avons, dans un premier temps sélectionné les variables descriptives pertinentes en utilisant notre algorithme de filtrage de variables STRASS. Puis, nous avons réalisé l'apprentissage suivant trois méthodes distinctes : les réseaux de neurones, les *k* plus proches voisins et arbres de décision. Les résultats de classification dans les trois cas sont satisfaisants, en effet, le système d'aide à la décision obtenu présente des capacités prédictives performantes. Nous avons aussi comparé notre méthodologie à différents algorithmes contextuels de sélection de variables tels que : mRMR, CFS, FCBF et INTERACT. Les résultats obtenus montrent que les algorithmes de sélection de variables contextuels améliorent considérablement les performances de classification des algorithmes d'induction et ainsi le diagnostic. Les tests de validation de la méthodologie proposée (la combinaison algorithme de sélection de variables contextuel et algorithme d'induction) sur différentes bases réelles de domaines d'applications variés ainsi qu'en diagnostic médicale et finalement sur des données en diagnostic de défauts ont d'une part, démontré l'efficacité de ce type d'algorithmes de sélection de données même quand les données sont hautement corrélées. D'autre part, les résultats obtenus montrent que notre algorithme de sélection de variables STRASS donne des performances soi comparables ou bien supérieures aux autres algorithmes de sélection de variables.

Nous avons aussi constaté que l'algorithme STRASS lors de son déroulement réalise une catégorisation de variables très intéressante. Exploitant cette catégorisation, nous

avons proposé dans un deuxième temps d'utiliser cette dernière afin de construire un système plus fiable de détection de défaut.

Enfin, plusieurs perspectives se dégagent suite au travail réalisé. Des directions de recherchent et extensions sont envisageables. Plus particulièrement les points suivants :

1. En ce qui concerne la méthode de sélection de variables que nous avons présentée, le problème de son indépendance par rapport à l'algorithme d'induction utilisé est un aspect que nous souhaitons étudier. La mise au point d'une méthode de filtrage efficace et indépendante de l'algorithme d'apprentissage serait un pas important en ECD.

2. Le passage paires-contingence, nous a permis de construire une variable collective qui synthétise l'information. Cette variable collective peut nous conduire vers un domaine peu éloigné de la sélection de variables qui est : la construction de variables. Il serait intéressant de faire le lien entre critère contextuel et construction de variable contextuelle.

3. Un autre point à considérer concerne la sélection d'exemples pertinents. En effet, un des challenges actuels dans le cadre de la sélection de données concerne l'intégration de la sélection des variables et des exemples pertinents dans un algorithme commun.

4. Concernant l'utilisation des outils de l'ECD dans le cadre du diagnostic, nous voudrions appliquer la phase de sélection de variables en post-traitement d'un algorithme d'extraction de variables (comme par exemple une analyse temps fréquence, analyse en composante principale, Fourier, …), afin de déterminer les descripteurs les plus pertinents à un problème donné.

Annexe A

Algorithme C4.5

L'algorithme C4.5 a été conçu par Quinlan (Quinlan, 1983). A partir de l'ensemble d'apprentissage, C4.5 extrait la régularité de règles à partir d'instances et construit un arbre de décision qui classifiera les instances avec un certain degré d'erreur tolérée. Dans le cadre de ce travail nous avons utilisé l'algorithme J48 de la plateforme Weka[1] (Witten, et al., 2000), ce dernier implante la méthode C4.5. L'algorithme C4.5 se base sur la mesure de l'entropie dans l'échantillon d'apprentissage. L'algorithme travaille sur des données symboliques que ce soient des variables catégorielles ou numériques discrètes. Les variables continues doivent être discrétisées avant la mise en œuvre de l'algorithme pour préserver l'efficacité de l'apprentissage et la pertinence du modèle produit.

Pour une instance donnée, les nœuds de l'arbre de décision sont utilisés pour tester les valeurs d'attribut. Suivre une branche ou une autre d'un nœud donné dépend des résultats du test effectué au niveau du dit nœud. A la fin du parcours de l'arbre de décision de la racine à une feuille selon les valeurs d'un problème donné, la classification trouvée à la feuille est la classification prédite du problème.

Pour la construction des sous-arbres, C4.5 utilise le taux de gain d'information (*IGR : information gain rate*) (Yazid, 2006), (Witten, et al., 2000) pour chacun des attributs possibles qui pourraient potentiellement être utilisés pour diviser les données. L'IGR est une heuristique qui évalue la capacité d'un attribut de réduire l'aspect aléatoire dans des instances non classifiées. L'attribut avec le plus grand IGR est choisi comme racine d'un sous-arbre. Bien qu'IGR soit une métrique qui prend une décision locale (puisqu'il évalue chaque attribut indépendamment des autres) par opposition à une décision

[1]Weka : Outils et méthodes dédiés au data mining implantés sous environnement Java

globalement optimale, l'attribut sélectionné est souvent le plus discriminatoire. L'algorithme peut prendre en compte différents types d'attributs ainsi que les valeurs manquantes.

La méthode d'élagage de C4.5 consiste à arrêter la construction de l'arbre une fois que l'ensemble d'apprentissage ait été suffisamment subdivisé en utilisant un critère. Elle est basé sur l'estimation du taux d'erreur de chaque sous-arbre, et remplace le sous-arbre avec un nœud feuille si l'erreur estimée de la feuille est très basse (Lereno, 2000).

Annexe B

Etude de la stabilité de l'algorithme de filtrage

Les tableaux (B-1) et (B-2) donnent les variables sélectionnées (noté S_i) par rapport à cinq pourcentages d'échantillonnage P_i (10%, 30%, 50%, 80% et 100% de l'ensemble total de données) de données filtrées par STRASS, Relief, CFS et FCBF. Tel que la moyenne sur l'ensemble des échantillons de données est donnée par : $S_{moy} = \left(\sum_{i=1}^{5} S_i \right) / 5$.

Les meilleurs résultats sont en gras. FCBF, STRASS et CFS réduisent considérablement les ensembles de données par la sélection de variables. Les tableaux (B-3 à B-6) présentent les performances ainsi que les dégradations de classification obtenues pour les algorithmes d'induction C4.5 (arbre de décision) et IB_5 (5-plus proche voisin) après filtrage des échantillons de données.

Tableau B-1. Variables sélectionnées par chaque algorithme de filtrage

Bases de données	vraiables	Fraction (%)	STRASS	Relief	CFS	FCBF
Heart	13	moyenne	7.8	8.4	5.8	4.6
		10	5	6	3	2
		30	7	9	6	6
		50	9	9	7	4
		80	9	9	7	6
		100	9	9	6	5
German	24	moyenne	2	9.8	2.2	4.12
		10	1	6	2	2
		30	2	8	1	3
		50	2	10	2	3
		80	3	10	2	4
		100	2	15	4	4
Iris	4	moyenne	1	4	1.6	1.4
		10	1	4	1	1
		30	1	4	2	2
		50	1	4	1	1
		80	1	4	1	1
		100	1	4	2	2
Ionosphere	34	moyenne	5.4	32	9.2	4.6
		10	2	32	5	5
		30	5	32	7	2
		50	5	32	12	6
		80	7	32	10	4
		100	8	32	12	5
Vehicle	18	moyenne	12.5	17	7.8	3.6
		10	11	17	4	3
		30	13	17	9	2
		50	12	17	8	5
		80	12	17	9	4
		100	13	17	9	4

Tableau B-2. Variables sélectionnées par chaque algorithme de filtrage

Bases de données	#vraiables	Fraction (%)	STRASS	Relief	CFS	FCBF
Wine	13	moyenne	4	12	7.4	8.8
		10	2	12	7	5
		30	4	12	6	8
		50	4	12	8	11
		80	5	12	8	10
		100	5	12	8	10
Mushroom	22	moyenne	2.2	16.8	1	3.6
		10	2	15	1	4
		30	2	16	1	4
		50	3	16	1	3
		80	3	16	1	3
		100	3	21	1	4
Austra	14	moyenne	11	12.4	1.8	5.4
		10	3	12	4	3
		30	10	11	2	6
		50	14	13	1	5
		80	14	13	1	6
		100	14	13	1	7
Kr-vs-kp	36	moyenne	24	21,8	5.4	6.4
		10	15	18	5	5
		30	21	21	5	6
		50	27	22	5	7
		80	27	26	6	7
		100	30	22	7	7
Nursery	8	moyenne	8	4	8	8
		10	8	4	8	8
		30	8	4	8	8
		50	8	4	8	8
		80	8	4	8	8
		100	8	4	8	8

Tableau B-3. Classification de C4.5 avec et sans filtrage de données

Bases de données	**C4.5**	Fraction (%)	STRASS		Relief		CFS		FCBF	
			Tc	S	Tc	S	Tc	S	Tc	S
Heart	83.7	moyenne	84 +	7.8	83.7	8.4	81.6-	5.8	80.42-	4.6
		10	84.44	5	84	6	72.59	3	76.3	2
		30	84.44	7	83.7	9	84	6	82.6	6
		50	83.7	9	83.7	9	84.44	7	74.8	4
		80	83.7	9	83.7	9	84	7	84	6
		100	83.7	9	83.7	9	83.3	9	84.44	5
German	**70.6**	moyenne	70.1 -	2	72.7+	9.8	70.4-	2.2	71.6 +	4.1
		10	69.7	1	71.5	6	69.9	2	69.9	2
		30	70.2	2	72.2	8	70	1	71.8	2
		50	70.2	2	72.8	10	70	2	72.7	3
		80	70.2	3	73.5	10	70	2	72.3	3
		100	70.2	2	73.5	15	72.3	4	71.5	4 4
Iris	**94**	moyenne	95.3 +	1	94-	4	95.5+	1.6	95+	1.4
		10	95.33	1	94	4	95.33	1	95.33	1
		30	95.33	1	94	4	94	2	94	2
		50	95.33	1	94	4	96	1	96	1
		80	95.33	1	94	4	96	1	96	1
		100	95.33	1	94	4	96	2	94	2
Ionosphere	**89.1**	moyenne	91,04+	5.4	89.14-	32	90.13	9.2	88.3-	4.6 8
		10	90	2	89.17	32	90.3	5	90.8	5
		30	91.16	5	89.4	32	90.88	7	80.9	2
		50	90.3	5	88.8	32	89.17	12	89.7	6
		80	91.73	7	89.17	32	90.6	10	90	4
		100	92	8	89.17	32	89.74	12	90.3	5
Vehicle	**72.1**	moyenne	72.53+	12.5	72.1	17	70.6 -	7.8	62.95-	3.6
		10	73.25	11	72.1	17	65.24	4	65.95	3
		30	72.97	13	72.1	17	70.68	9	53.54	2
		50	72.45	12	72.1	17	70.44	8	68.32	5
		80	72	12	72.1	17	73.3	9	69.1	4
		100	72	13	72.1	17	73.3	9	57.8	4

Tableau B-4. Classification de C4.5 avec et sans filtrage

Bases de données	**C4.5**	Fraction (%)	STRASS		Relief		CFS		FCBF	
			Tc	S	Tc	S	Tc	S	Tc	S
Wine	**93.25**	moyenne	94.78+	4	93.25	12	93.9 +	7.4	94.25+	8.8
		10	95.5	2	93.25	12	92.69	7	93.8	5
		30	94.3	4	93.25	12	94.4	6	94.4	8
		50	94.3	4	93.25	12	94.38	8	93.3	11
		80	95.5	5	93.25	12	93.83	8	96	10
		100	94.3	5	93.25	12	94.38	8	93.8	10
Mushroom	**100**	moyenne	99.9	2.2	100	16.8	98.52-	1	99-	3.6
		10	99.70	2	100	15	98.52	1	99	4
		30	99.70	2	100	16	98.52	1	99	4
		50	100	3	100	16	98.52	1	99	3
		80	100	3	100	16	98.52	1	99	3
		100	100	3	100	21	98.52	1	99	4
Austra	**86.68**	moyenne	86.52	11	86.6	12.4	85.5 -	1.8	86.3-	5.4
		10	86.3	3	86.6	12	85.5	4	85.5	3
		30	86.3	10	86.6	11	85.5	2	86.8	6
		50	86.68	14	86.6	13	85.5	1	87.24	5
		80	86.68	14	86.6	13	85.5	1	85.65	6
		100	86.68	14	86.6	13	85.5	1	86.5	7
Kr-vs-kp	**99.43**	moyenne	98,82-	24	97.93-	21.8	94.1-	5.4	94.2-	6.4
		10	98.2	15	97.84	18	94	5	94	5
		30	98.31	21	97.77	21	94	5	94	6
		50	99.2	27	97.77	22	94	5	94	7
		80	99.2	27	98.52	26	94.48	6	94.64	7
		100	99.2	30	97.77	22	94	7	94.48	7
Nursery	**96.19**	moyenne	96.19	8	89.56 -	4	96.19	8	96.19	8
		10	96.19	8	89.98	4	96.19	8	96.19	8
		30	96.19	8	87.91	4	96.19	8	96.19	8
		50	96.19	8	89.98	4	96.19	8	96.19	8
		80	96.19	8	89.98	4	96.19	8	96.19	8
		100	96.19	8	89.98	4	96.19	8	96.19	8

Tableau B-5. Classification de IB_k avec et sans filtrage de données

Bases de données	**IB_5**	Fraction (%)	STRASS		Relief		CFS		FCBF	
			Tc	S	Tc	S	Tc	S	Tc	S
Heart	81.8	moyenne	82.49 +	7.8	83.2+	8.4	82.5+	5.8	81.9	4.6
		10	83.7	5	84.4	6	75.55	3	76.3	2
		30	83.33	7	82.9	9	85.1	6	85.5	6
		50	81.8	9	82.9	9	81.85	7	78.5	4
		80	81.8	9	82.9	9	86.3	7	85.18	6
		100	81.8	9	82.9	9	84	6	84	5
German	73.7	moyenne	66.64 -	2	72.5-	9.8	70.66-	2.2	73.18 -	4.1
		10	64.8	1	69.4	6	70.3	2	70.3	2
		30	66.6	2	74.1	8	69.3	1	71.8	3
		50	66.6	2	72.5	10	69.3	2	73.8	3
		80	66.6	3	73	10	69.3	2	74.9	4
		100	66.6	2	73.9	15	75.1	4	75.1	4
Iris	94.6	moyenne	95.33+	1	96 +	4	96 +	1.6	95.8 +	1.4
		10	95.33	1	96	4	96	1	95.3	1
		30	95.33	1	96	4	96	2	96	2
		50	95.33	1	96	4	96	1	96	1
		80	95.33	1	96	4	96	1	96	1
		100	95.33	1	96	4	96	2	96	2
Ionosphere	89.4	moyenne	90,47 +	5.4	89.4	32	88.2-	9.2	86.8 -	4.6
		10	88.9	2	89.17	32	91.16	5	91.1	5
		30	90.3	5	89.74	32	88.31	7	82.6	2
		50	91.16	5	89.17	32	87.7	12	87.7	6
		80	92.3	7	89.45	32	86	10	85	4
		100	89.74	8	89.45	32	88	12	88	5
Vehicle	70.56	moyenne	71.61+	12.5	70.56	17	66.94 -	7.8	61 -	3.6
		10	68.9	11	70.56	17	63.94	4	64	3
		30	71.9	13	70.56	17	68.32	9	52.8	2
		50	72.57	12	70.56	17	65.83	8	65.6	5
		80	72.7	12	70.56	17	68.32	9	66.5	4
		100	72	13	70.56	17	68.32	9	56.3	4

Tableau B-6. Classification de IB_k avec et sans filtrage de données

Bases de données	**IB_5**	Fraction (%)	STRASS		Relief		CFS		FCBF	
			Tc	S	Tc	S	Tc	S	Tc	S
Wine	91	moyenne	93.25 +	4	91	12	93.4 +	7.4	93 +	8.8
		10	91.57	2	91	12	94.38	7	91.57	5
		30	92.7	4	91	12	91.5	6	93.25	8
		50	92.7	4	91	12	93.25	8	92.69	11
		80	95.5	5	91	12	94.3	8	93.25	10
		100	93.8	5	91	12	94.3	8	94.38	10
Mushroom	99.9	moyenne	99.7	2.2	99.8	16.8	98.52--	1	98.91-	3.6
		10	99.70	2	99.8	15	98.52	1	98.91	4
		30	99.70	2	99.8	16	98.52	1	98.91	4
		50	99.7	3	99.8	16	98.52	1	98.91	3
		80	99.8	3	99.8	16	98.52	1	98.91	3
		100	99.7	3	99.8	21	98.52	1	98.91	4
Austra	85.5	moyenne	86.72 +	11	85.3-	12.4	85.5	1.8	85.4	5.4
		10	86	3	85.21	12	85.6	4	83.18	3
		30	86.98	10	85	11	85.5	2	85.65	6
		50	87.14	14	85	13	85.5	1	86.23	5
		80	87.14	14	86	13	85.5	1	86.37	6
		100	86.37	14	85.5	13	85.5	1	85.65	7
Kr-vs-kp	96.37	moyenne	96,88 +	24	96.03-	21,8	94 -	5.4	93.5 -	6.4
		10	96.33	15	95.71	18	94	5	91.22	5
		30	96.9	21	96.24	21	94	5	94.15	6
		50	97.3	27	96.24	22	94	5	94.15	7
		80	97.3	27	95.65	26	94	6	94	7
		100	96.65	30	96.33	22	94.1	7	94	7
Nursery	97.84	moyenne	97.84	8	88 -	4	97.84	8	97.84	8
		10	97.84	8	86.94	4	97.84	8	97.84	8
		30	97.84	8	86.94	4	97.84	8	97.84	8
		50	97.84	8	86.94	4	97.84	8	97.84	8
		80	97.84	8	89.59	4	97.84	8	97.84	8
		100	97.84	8	89.59	4	97.84	8	97.84	8

Annexe C

Présentation du TEP

Ce procédé Figure (C-1) est composé de cinq éléments principaux : un réacteur, un compresseur, un décapeur, séparateur et un condenseur.

Le procédé produit deux composants liquides G et H à partir de quatre gaz réactifs A, C, D et E. Le système implique également un gaz B inerte (non réactif), ainsi qu'un dérivé de production F.

Tableau C-1. Variables de mesure en continu

Variable	Description	Unité
XMES(1)	Débit d'alimentation en A	kscmh
XMES(2)	Débit d'alimentation en D	kg/hr
XMES(3)	Débit d'alimentation en E	kg/hr
XMES(4)	Débit d'alimentation total	kscmh
XMES(5)	Débit de recyclage	kscmh
XMES(6)	Débit d'alimentation du réacteur	kscmh
XMES(7)	Pression du réacteur	kPa
XMES(8)	Niveau du réacteur	%
XMES(9)	Température du réacteur	˚C
XMES(10)	Débit de purge	kscmh
XMES(11)	Température du séparateur	˚C
XMES(12)	Niveau du séparateur	%
XMES(13)	Pression du séparateur	kPa
XMES(14)	Débit du séparateur	m^3 /hr
XMES(15)	Niveau du décapeur	%
XMES(16)	Pression du décapeur	kPa
XMES(17)	Débit du décapeur	m^3 /hr
XMES(18)	Température du séparateur	˚C
XMES(19)	Débit de gaz au séparateur	kg/hr
XMES(20)	Puissance du compresseur	kW
XMES(21)	Température de ref. liq. en sortie de réacteur	˚C
XMES(22)	Température de ref. liq. en sortie de séparateur	˚C

Les gaz réactifs (A, C, D et E) alimentent le réacteur où ils réagissent et forment, à l'aide d'un catalyseur, les produits G et H sous forme gazeuse. Un système de refroidissement liquide (par eau) à l'intérieur du réacteur permet l'extraction d'une grande partie de la chaleur produite par celui-ci. Les produits quittent le réacteur, alors que le catalyseur reste dans celui-ci. Le gaz produit est refroidi au moyen d'un condenseur et alimente alors le séparateur liquide vapeur. La vapeur non condensée dans le séparateur est renvoyée vers le réacteur au moyen d'un compresseur. Le gaz inerte B et le produit dérivé F sont purgés du procédé dans le séparateur (Verron, 2007).

Tableau C-2. Variables de mesures échantillonnées

Variable	Composant	Période d'échan- tillonage (en min)	Unité
XMES(23)	A	6	mol%
XMES(24)	B	6	mol%
XMES(25)	C	6	mol%
XMES(26)	D	6	mol%
XMES(27)	E	6	mol%
XMES(28)	F	6	mol%
XMES(29)	A	6	mol%
XMES(30)	B	6	mol%
XMES(31)	C	6	mol%
XMES(32)	D	6	mol%
XMES(33)	E	6	mol%
XMES(34)	F	6	mol%
XMES(35)	G	6	mol%
XMES(36)	H	6	mol%
XMES(37)	D	15	mol%
XMES(38)	E	15	mol%
XMES(39)	F	15	mol%
XMES(40)	G	15	mol%
XMES(41)	H	15	mol%

Ce procédé comporte 53 variables : 12 variables d'asservissement et 41 variables mesurables. Parmi les 41 variables mesurables, 22 sont des variables mesurables en continu (ce sont les valeurs des capteurs du procédé), alors que les autres sont des

mesures de compositions telles que des concentrations, et ne sont donc pas disponibles en continu mais échantillonné. Les 22 variables mesurables en continu sont listées dans le tableau (C-1) alors que les autres variables mesurables sont visibles dans le tableau (C-2). Les 12 variables d'asservissement sont données dans le tableau (C-3). Le schéma du TEP et de son asservissement sont donnés sur la figure (C-2).

Tableau C-3. Variables de contrôle du TEP

Variable	Description	Unité
XC(1)	Débit d'alimentation en D	kg/hr
XC(2)	Débit d'alimentation en E	kg/hr
XC(3)	Débit d'alimentation en A	kscmh
XC(4)	Débit d'alimentation en A et C	kscmh
XC(5)	Valve de recyclage du compresseur	%
XC(6)	Valve de purge	%
XC(7)	Débit d'alimentation du séparateur	m^3 /hr
XC(8)	Débit d'alimentation du séparateur	m^3 /hr
XC(9)	Valve du décapeur	%
XC(10)	Débit du refroidissement liquide au réacteur	m^3 /hr
XC(11)	Débit du refroidissement liquide au condenseur	m^3 /hr
XC(12)	Vitesse de l'agitateur	tr/min

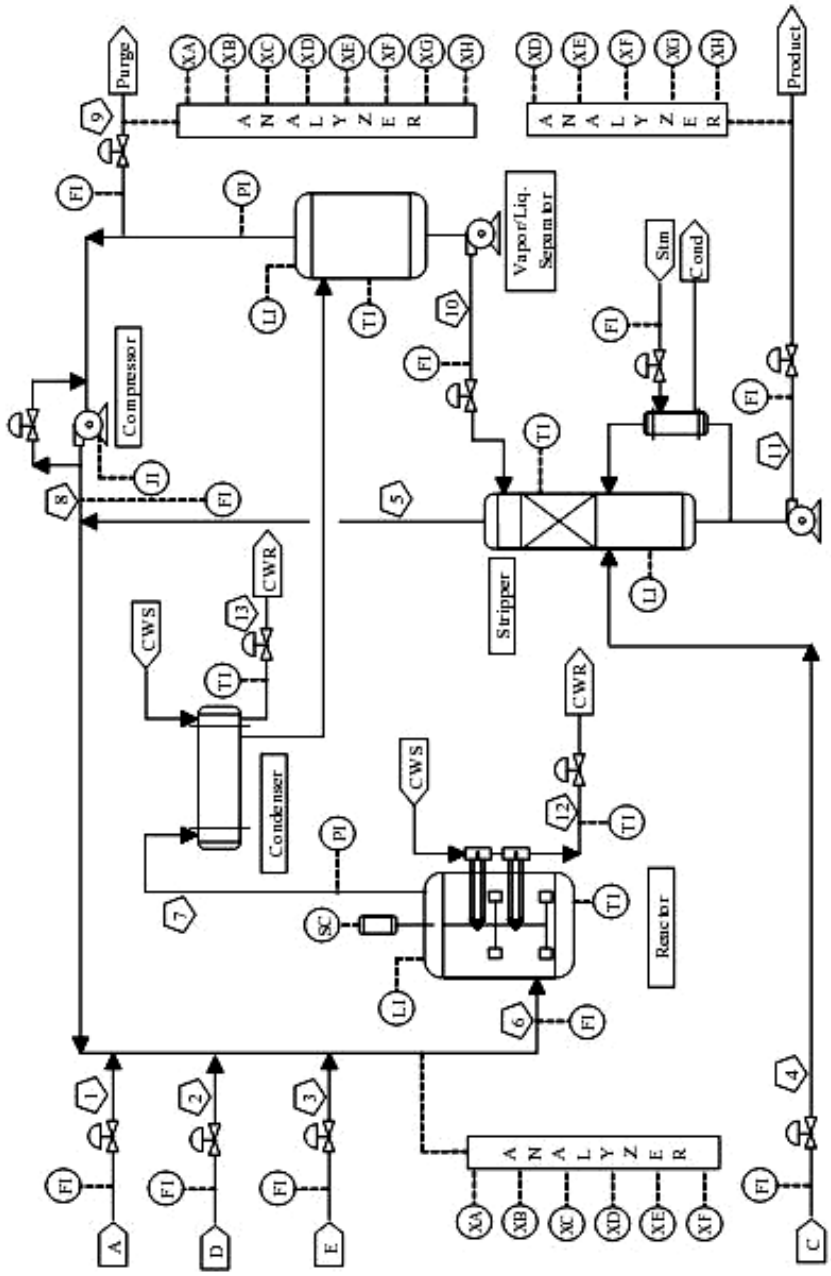

Figure C-1: *Schéma du Tennessee Eastman Process*

TEP est un procédé qui peut être soumis à 20 fautes différentes. Ces fautes sont de diverses natures. La description de ces 20 fautes est donnée dans le tableau (C-4). Les fautes F16 à F20 sont inconnues.

Tableau C-4. Les différentes fautes du TEP

Faute	Description	Type
F1	Ratio d'alimentation A/C	Saut
F2	Composition en B	Saut
F3	Temp. d'alimentation en D	Saut
F4	Temp. d'entrée du ref. liq. au réacteur	Saut
F5	Temp. d'entrée du ref. liq. au condenseur	Saut
F6	Baisse d'alimentation en A	Saut
F7	Perte de pression de l'alimentation en C	Saut
F8	Composition d'alimentation en A, B et C	Variation aléatoire
F9	Temp. d'alimentation en D	Variation aléatoire
F10	Temp. d'alimentation en C	Variation aléatoire
F11	Temp. d'entrée du ref. liq. au réacteur	Variation aléatoire
F12	Temp. d'entrée du ref. liq. au condenseur	Variation aléatoire
F13	Cinétiques des réactions	Dérive lente
F14	Valve du ref. liq. au réacteur	Bloquée
F15	Valve du ref. liq. au condenseur	Bloquée
F16	Inconnue	Inconnue
F17	Inconnue	Inconnue
F18	Inconnue	Inconnue
F19	Inconnue	Inconnue
F20	Inconnue	Inconnue

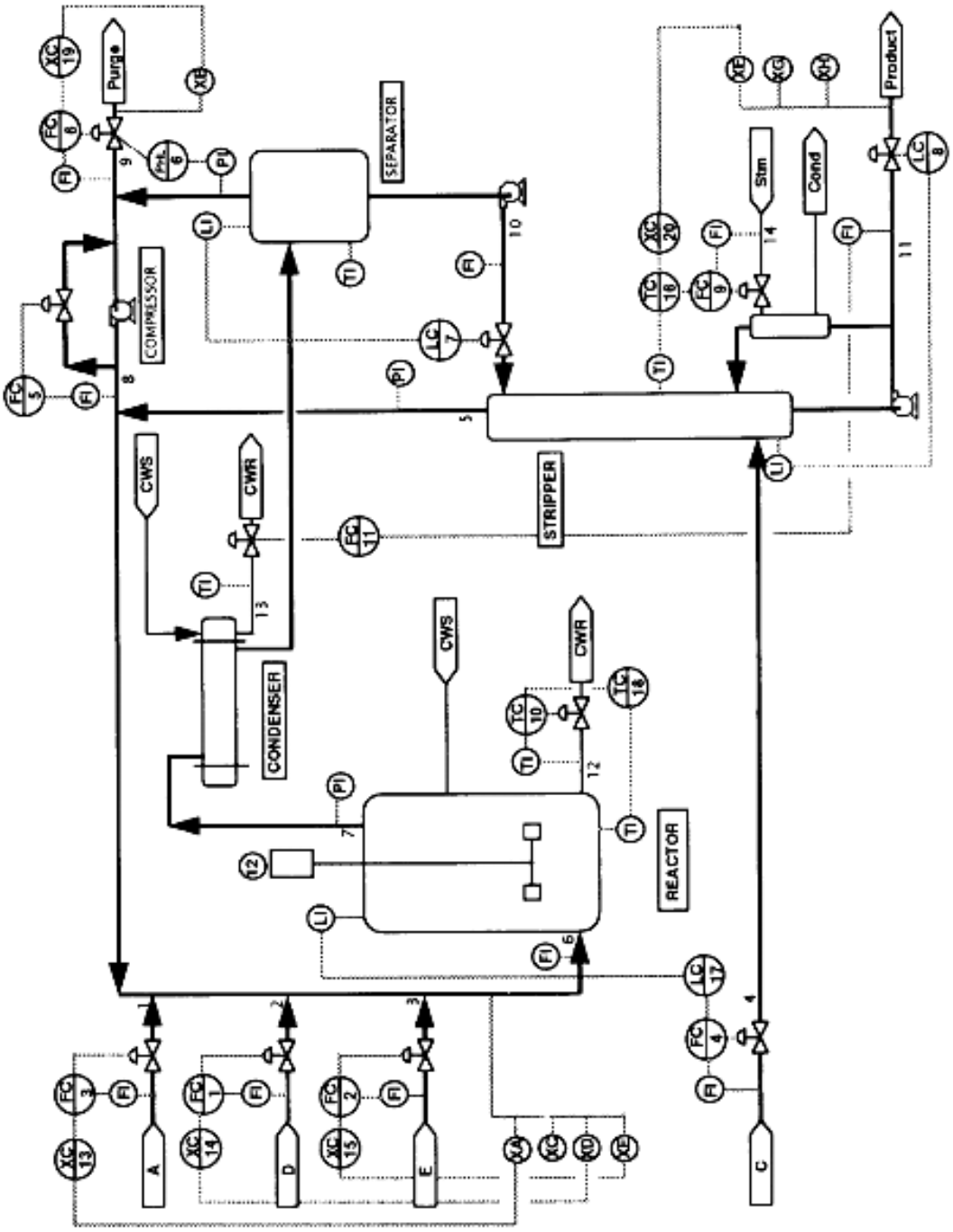

Figure C-2 : *TEP asservi par Lyman et Georgakis* (Lyman & Georgakis, 1995)

Bibliographie

AFNOR. (2001). Maintenance terminology. European standard. NF EN 13306.

Agrawal, R., Ghosh, S., Imielinski, T., Iyer, B., & Swami, A. (1992). An interval classifier for database mining applications. *VLDB Conference*.

Aha, D., & Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning ,* Vol 6, p. 37-66.

Almuallim, H., & Diettrich, T. G. (1994). Learning Boolean concepts in the presence of many irrelevant features. *Artificial Intelligence ,*Vol 69, p. 279-305.

Almuallim, H., & Diettrich, T. G. (1991). Learning with Many Irrelevant Features. *Proc. the Ninth National Conference on Artificial Intelligence*, p. 547-552.

Arsselin, J. P., & Kettaf, F. (2005). *Bases théoriques pour l'apprentissage et la décision en reconnaissance de formes.* Cépaduès.

Barna, G. (1997). Procedures for Implementing Sensor-Based Fault Detection and Classification (FDC) for Advanced Process Control (APC). SEMATECH Technical Transfer Document.

Barry, M. W., & Neal, B. (1999). PARAFAC2 Part III. Application to Fault Detection and Diagnosis in Semiconductor Etch. Gallagher Eigenvector Research, Inc. Manson, WA USA.

Battiti, R. (1994). Using Mutual Information for Selection Features In Supervised Neural Net Learning. *IEEE Transactions on Neural Networks ,* Vol 5 (4), p. 537-550.

Becker, A., & Naim, P. (1999). *Les réseaux bayésiens.* Paris: Eyrolles.

Blum, A. L. (1992). Learning Boolean Functions in a Infinite Attribute Space. *Machine Learning ,* Vol 9, p. 373-386.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence ,* Vol 97, p. 245-271.

Bobrowski, L. (1988). Feature selection based on some homogeneity coefficient. *Proc. of the Ninth International Conference on Pattern Recognition*, p. 544-546.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. *Belmont, California: Wadsworth International* .

Butine, W. (1992). Learning classification trees, Statics and computing. p. 63-73.

Casillas, J., Cordón, O., Jesus, M. d., & Herrera, F. (2001). Genetic Feature Selection in a Fuzzy Rule-Based Classification System Learning Process. *Information Sciences ,* Vol 136, p. 135-157.

Casimira, R., Boutleuxa, E., Clercb, G., & Yahouib, A. (2006). The use of features selection and nearest neighbors rule for faults diagnostic in induction motors. *Engineering Applications of Artificial Intelligence ,* Vol 19 , p. 169–177 .

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering ,* Vol 40 (1), p. 16-28.

Chiang, L., Kotanchek, M., & Kordon, A. (2004). Fault diagnosis based on fisher discriminant analysis and support vector machines. *Computers and Chemical Engineering ,* Vol 28 (8), p. 1389-1401.

Cornuéjols, A., Miclet, L., & Y.Kodratoff. (2011). *Apprentissage Artificiel: Concepts et algorithmes.* Eyrolles.

Dash, M., & Liu, H. (1997). Feature Selection for Classification. *Intelligent Data Analysis ,* Vol 1 (3), p. 131–156.

Dash, M., Liu, H., & Motoda, H. (2000). Consistency Based Feature Selection. *PAKDD*, p. 98-109.

Downs, J., & Vogel, E. (1993). Plant-wide industrial process control problem. *Computers and Chemical Engineering ,* Vol 17 (3), p. 245–255.

Dubuisson, B. (1990). *Diagnostic et reconnaissance des formes.* Hermes.

Dubuisson, B., E. B., Dague, P., Denoeux, T., Didelet, E., Gandvalet, Y., et al. (2001). *Diagnostic, Intelligence Artificielle et reconnaissance de formes.* Hermes.

Fayyad, M. U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. *Advances in Knowledge Discovery and Data Mining* , p. 1-34.

Fayyad, U. M., & Irani, K. B. (1990). The attribute selection problem in decision tree generation. *Proc. of the 10th National Conference on Artificial Intelligence ,* Vol 1, p. 749 – 754.

Fayyad, U. M., & Irani, K. B. (1993). Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. *International Joint Conference on Artificial Intelligence* , p. 1022-1029.

Graja, S. (2008). *Segmentation et classification de l'onde P d'un électrocardiogramme : détection d'un risque de fibrillation auriculaire*, Thèse. Ecole nationale supérieure des télécommunications de bretagne (France).

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research ,* Vol 3 (1), p. 1157-1182.

Haouchine, M, K. (2009). *Remémoration guidée par l'adaptation et maintenance des systèmes de diagnostic industriel par l'approche du raisonnement à partir de cas*. Thèse de Doctorat. Université de Franche-Comte (France).

Hall, M. (1998). *Correlation-based feature selection for machine learning,* Thèse. University of Waikato, Hamilton, New Zealand:.

Hall, M. (2000). Correlation-based feature selection of discrete and numeric class machine learning. *Proceedings of the International Conference on Machine Learning* , p. 359-366.

Héng, J. (2005). *Pratique de la maintenance préventive.* France: Dunod.

Herault, J. (1994). *Réseaux neuronaux et traitement de signal.* France: Hermes.

Hong, S. (1994). Use of contextual information to feature ranking and discretization. *IEEE Trans on Knowledge and Data Engineerin,* Vol 9(5), p. 718–730 .

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks ,* Vol 4, p. 251-257.

Irvine, U. *UCI Knowledge Discovery in Databases Archive*. en ligne sur : www.ics.uci.edu/~mlearn/MLRepository.html

Jack, L. B., & Nandi, A. K. (2002). Fault detection using support vector machines and artificial neural network augmented by genetic algorithms. *Mechanical systems and Signal Processing ,* Vol 16, p. 37-390.

Jakulin, A., & Bratko, I. (2004). Testing the significance of attribute interactions. *Proceedings of the twenty-first international conference on Machine learning* , p. 409-416.

Jockenhövel, T., Biegler, L. T., & Wächter, A. (2003). Tennessee Eastman Plant-wide Industrial Process Challenge Problem. Complete Model. *Computers and Chemical Engineering ,* Vol 27, p. 1513-1531.

John, G. H., & Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. *Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo*, p. 338-345.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant Features and the Subset Selection Problem. (M. Kaufmann, Éd.) *Proceedings of the Eleventh International Conference on Machine Learning* , p. 121-129.

Jun, B. H., Kim, C., Song, H., & Kim, J. (1997). A New Criterion in selection and discretization of attributes for the generation of decision trees . *IEEE Transactions on pattern analysis and machine intelligence ,* Vol 19 (12), p. 1371-1375.

Kalousis, A., Prados, J., & Hilario, M. (2005). Stability of Feature Selection Algorithms. *Proceedings of the Fifth IEEE International Conference on Data Mining* .

Karegowda, A. G., Manjunath, A. S., & M.A.Jayaram. (2010). Comparative Study Of Attribute Selection Using Gain Ratio And Correlation Based Feature Selection . *International Journal of Information Technology and Knowledge Management ,* Vol 2 (2), p. 271-277 .

KDD-UCI. en ligne sur : http://kdd.ics.uci.edu

Kerber, R. (1992). ChiMerge: Discretization of Numeric Attributes. *AAAI Press / The MIT Press* , p. 123-128.

Kira, K., & Rendell, L. A. (1992(a)). A Practical Approach to Feature Selection. *Proc. of the Ninth International Workshop* , p. 249-255.

Kira, K., & Rendell, L. (1992(b)). The feature selection problem: Traditional methods and a new algorithm. *Proceedings of the Tenth National Conference on Artificial Intelligence* , p. 129-134.

Kodratoff, Y., & Diday, E. (1991(a)). *Induction Symbolique et Numérique à partir de données.* Cepaduès.

Kodratoff, Y., & E.Diday. (1991(b)). *Numeric and Symbolic Induction From Data.* Cepadues, Cepad.

Kohavi, R., & John, G. H. (1997). Wrappers for Feature Subset Selection. *Artificial Intelligence ,* Vol 97, p. 273-324.

Koller, D., & Sahami, M. (1996). Toward Optimal Feature Selection. *the Thirteenth International Conference on Machine Learning* , p. 284-292.

Kononenko, I. (1994). Estimating Attributes: Analysis and Extensions of Relief. *Proc. of the Seventh European Conference on Machine Learning* , p. 171-182.

Kononenko, I., Simec, E., & Robnik-Sikonja, M. (1997). Overcoming the Myopia of Inductive learning algorithm with RELIEFF . *Applied Intelligence ,* Vol 7, p. 39 -55.

Langley, P. (1994). Selection of relevant features in machine learning. *Proc of the AAAI, Fall Symposium on relevance* , p. 399-406.

Langley, P., & Sage, S. (1997). Scaling to Domains with Irrelevant features, Computational learning theory and natural learning systems. *MA: MIT Press, Cambridge ,* Vol 4, p. 17-29.

Lanzi, P. (1997). Fast Feature Selection With Genetic Algorithms: A Filter Approach. *IEEE International Conference on Evolutionary Computation* , p. 537-540.

Lereno, E. (2000). *Apprentissage des problèmes d'ordonnancement: application des méthodes de filtrage de données.* Besançon (France), Thèse de doctorat. Université de Franche- Comté.

Li, W., Li, D., & Ni, J. (2003). Diagnosis of tapping process using spindle motor current. *International Journal of Machine Tools & Manufacture ,* Vol 43, p. 73–79.

Liu, H., & Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining.* Kluwer Academic .

Liu, H., & Setiono, R. (1995). chi2 : Feature Selection and discretization of numeric attributes. *7th IEEE International conference on Tools with artificial intelligence* .

Liu, H., & Yu, L. (2005). Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Trans on Knowledge and Data Engineering ,* Vol 17 (4), p. 491-502.

Liu, H., Motoda, H., & Dash, M. (1998). A Monotonic measure for optimal feature selection . *Proc. of the 10th International Conference on Machine Learning* , p. 101-106.

Liu, H., Yu, L., Dash, M., & Motoda, d. H. (2004). Active Feature Selection Using Classes. *Proc. Seventh Pacific-Asia Conf. Knowledge Discovery and Data Mining ,* Vol 159 (1-2), p. 49-74.

Lyman, P., & Georgakis, C. (1995). Plant-wide control of the tennessee eastman problem. *Computers and Chemical Engineering ,* Vol 19 (3), p. 321-331.

Mantaras, R. L. (1991). A distance based attribute selection for decision tree induction. *Machine Learning ,* Vol 6, p. 81-92.

Marcotorchino, F. (1984). Utilisation des comparaisons par paires en statistiques des contingences. *1° part Etude n° F-069 et 2 part n° F-071*. French IBM Scientific Center.

Mathieu-Dupas, E. (2010). Algorithme des K plus proches voisins pondérés (WKNN) et Application en diagnostic. *42èmes Journées de Statistique* .

Michaud, P. (1982). Agrégation à la majorité : Hommage à Condorce. *N° F-051*. French IBM Scientific Center report.

Michaut, D. (1999). *Filtrage et sélection d'attributs en apprentissage*. Thèse. Université de Franche Comté (France).

Michaut, D., & Baptiste, P. (1999). Selection of a Relevant Feature Subset for Induction Tasks. *11th of ISMIS, Warshaw, Springer Verlag Poland* .

Morello, B. C., Michaut, D., & Baptiste, P. (2001). A knowledge discovery process for a flexible manufacturing system. *Proc. of the 8th IEEE International Conference on Emerging Technologies and Factory Automation ,* Vol 1, p. 652-659.

Mucciardi, A. N., & Gose, E. (1971). A Comparison of Seven Techniques for Choosing Subsets of Pattern Recognition Properties. *IEEE Transactions on Computers*, p. 1023-1031.

Narendra, P. M., & Fukunaga, K. (1977). A Branch and Bound Algorithm for Feature Subset Selection. *IEEE Transactions on Computers ,* Vol 26 (9), p. 917-922, .

Nashalji, M. N., Shoorehdeli, M. A., & Teshnehlab, M. (2009). Fault Detection of the Tennessee Eastman Process Using Improved PCA and Neural Classifier. *International Journal of Electrical & Computer Sciences ,* Vol 9 (9), p. 54-60.

Paljak, G., Kocsis, I., Egel, Z., Toth, D., & Pataricza, A. (2010). Sensor Selection for IT Infrastructure Monitoring. *AUTONOMICS* , p. 130–143.

Paljak, G., Kocsis, I., Egel, Z., Toth, D., & Pataricza, A. (2009). Sensor Selection for IT Infrastructure Monitoring. *Third International ICST Conference on Autonomic Computing and Communication Systems* .

Peng,H. mRMR en ligne sur: *http://penglab.janelia.org/software/Hanchuan_Peng_Software/software.html*.

Peng, H., Long, F., & Ding, C. (2005). Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence ,* Vol 27 (8), p. 1226-1238.

Pudil, P., Navovicova, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters ,* Vol 15, p. 1119–1125.

Quilan, J. R. (1986). Induction of decision trees. *Machine Learning , 1*, 81-106.

Quinlan, J. R. (1983). *C4.5: Programs for Machine Learning.* M. Kaufmann, San Francisco.

Quinlan, J. R. (1979). Discovering rules by induction from large collections of examples. *Expert Systems in the Microelectronic Age* , p. 168-201.

Racoceanu, D. (2006). *Contribution à la surveillance des Systèmes de Production en utilisant les Techniques de l'Intelligence Artificielle.* HDR : Université de FRANCHE-COMTÉ (France).

Rakotomalala, R. (2005). Arbres de Décision. *Revue MODULAD* Vol 33, p. 163-187.

Ricker, N. (1996). Decentralized control of the tennessee eastman challenge process. *Journal of Process Control ,* Vol 6 (4), p. 205–221.

Riverol, C., & Carosi, C. (2008). Integration of fault diagnosis based on case based reasoning. *Brewing Sens.& Instrumen. Food Qual: Springer*, Vol 2, p. 15-20.

Russell, S. J., & Norvig, P. (2010). *Intelligence Artificielle*. Pearson Education France.

Sebban, M., & Nock, R. (2002). A hybrid Filter/wrapper approach of feature selection using information theory. *Pattern Recognition ,* Vol 35, p. 835–846.

Senoussi, H. (2001). *Localisation d'objets superposés dans une scène multi-formes.* Thèse de magistère. Université des sciences et de la technologie. Oran. Mohammed Boudiaf.

Senoussi, H., & Chebel-Morello. (2008). A New Contextual Based Feature Selection. *Proc. IEEE World Congress on Computational Intelligence*. WCCI 2008, Hong Kong, China, 1265 - 1272. disponible en ligne sur : https://hal.inria.fr/hal-00342421

Senoussi, H., Chebel-Morello, B., Denai, M., & Zerhouni, N. (2011(a)). Feature selection for fault detection systems: Application to the Tennessee Eastman Process. *Proc. IEEE Conference on Automation Science and Engineering,* Trieste*,* Italiy, p. 189-194. Abstract disponible en ligne sur :

https://ras.papercept.net/conferences/scripts/abstract.pl?ConfID=37&Number=122

Senoussi, H., Chebel-Morello, B., Denaï, M., & Zerhouni, N. (2011(b)). Feature Selection and Categorization to Design Reliable Fault Detection Systems. *Proc*. *Conference of the Prognostics and Health Management Society PHM'11,* Sep 2011, Montreal, Quebec, p. 257-266. disponible en ligne sur:

https://hal.archives-ouvertes.fr/file/index/docid/632331/filename/PHMBrigitte.pdf

http://www.phmsociety.org/sites/phmsociety.org/files/PHM11Proceeding.pdf

Senoussi, H., Chebel-Morello, B., Denai, M., Zerhouni, N., & Boudinar., A. H. (2012). A Comparative Study on Feature Selection to Design Reliable Fault Detection Systems. *International Review on Computers and Software ,* Vol 7 (5), p. 2070-2077.

Shanon, C. (1948). *The mathematical theory of communication*. University of Illinois Press, Urbana, USA. WEAVERS.

Skalak, D. (1994). Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. *Proc. 11th International Conference on Machine Learning*, p. 293-301.

Stigler, S. (2008). Karl Pearson's theoretical errors and the advances they inspired. *Statistical Science.* Vol 23, p. 261-271.

Sugumaran, V., Muralidharan, V., & Ramachandran, K. (2007). Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing. *Mechanical Systems and Signal Processing ,* Vol 21, p. 930-942.

Torkola, K., Venkatesan, S., & Liu, H. (2004). Sensor Selection for Maneuver Classification. *IEEE Intelligent TranspOltation Systems Conference , Washington, D.C., USA,* Vol 36.

Tyan, C., Wang., P., & Bahler, D. (1996). An application on intelligent control using neural network and fuzzy logic. *Neurocomputing ,* Vol 12 (4), p. 345-363.

Verron, S. (2007). Diagnostic et surveillance des processus complexes par réseaux bayésiens. Thèse : Université d'Angers (France).

Verron, S., Tiplica, T., & Kobi, A. (2008). Fault detection and identification with a new feature selection based on mutual information. *Journal of Process Control ,* Vol 18, p. 479-490.

Verron, S., Tiplica, T., & Kobi, A. (2006). Fault Diagnosis with Bayesian Networks:Application to the Tennessee Eastman Process. *IEEE International Conference on Industrial Technology. ICIT 2006*, p. 98-103.

Vignes, R., & Lebbe, J. (1992). Sélection d'un sous ensemble de descripteurs maximalement discriminant dans une base de connaissances. *3èmes journées Symbolique-Numérique*, p. 219-232.

Werbos, P. J. (1990). Backpropagation trough time : What it does and how do it. *Proceedings of IEEE ,* Vol 78 (10), p. 1550 - 1560.

Widodo, A., & B.Yang. (2007). Application of nonlinear feature extraction and support vector machines for fault diagnosis of induction motors. *Expert system with application ,* Vol 33, p. 241-253.

Witten, I. H., & Frank, E. (2000). *Data Mining—Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann, San Francisco.

Yang, B., & Widodo, A. (2008). Support Vector Machine for Machine Fault Diagnosis. *journal of system design and dynamics ,* Vol 2 (1), p. 12-23.

Yang, J., & Olafsson, S. (2006). Optimization-based feature selection with adaptive instance sampling. *Computers & Operations Research ,* Vol 33, p. 3088–3106.

Yazid, H. (2006). *Les algorithmes d'apprentissage automatique offerts par l'environnement Weka.* Montréal: Université du Québec.

Yu, L., & Liu, H. (2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research ,* Vol 5, p. 1205–1224.

Yu, L., & Liu, H. (2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research ,* Vol 5, p. 1205–1224.

Zemouri, R. (2003). *Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance.* Thèse de Doctorat : Université de Franche Comté (France).

Zhao, Z., & Liu, H. (2007). Searching for Interacting Features. *International Joint Conference on Artificial Intelligence*.

Zighed, D., Rakotomalala, R., & Feschet, F. (1997). Optimal multiple intervals discretization of continuous attributes for supervised learning. *proc of the 3th International Conference in Knowledge Discovery in Databases* , p. 295-298.

Zio, E., Baraldi, P., & Roverso, D. (2005). An extended classifiability index for feature selection in nuclear transients. *Annals of Nuclear Energy ,* Vol 32 , p. 1632–1649.

Zurada, J. M. (1992). *Artificial neural systems.* West publishing company.

Zwingelstein, G. (1995). *Diagnostic des défaillances.* Hermes.