UNIVERSITE SAAD DAHLAB DE BLIDA

Faculté des Sciences

Département d'Informatique

MEMOIRE DE MAGISTER

Spécialité : Systèmes d'information et de connaissances

CLASSIFICATION HYBRIDE A BASE DE RESEAUX DE NEURONES ET DE RESEAUX BAYESIENS

Par

Karim BOUDJEBBOUR

Devant le jury composé de :

Mme N. Benblidia	Maître de conférences, USDB	Présidente
Mr K. Kara	Maître de conférences, USDB	Examinateur
Mr F. Azouaou	Maître de conférences, ESI	Examinateur
Mme S. Oukid Khouas	Maître de conférences, USDB	Promotrice

Blida, Novembre 2010

RESUME

Les données collectées lors de l'observation d'un phénomène ou mesurées sur un système physique ne sont pas toutes aussi informatives : certaines variables peuvent correspondre à du bruit, être peu significatives, corrélées ou non pertinentes, ce qui influs considérablement sur leur classification. La sélection de variables est donc un problème complexe et fait l'objet de recherches dans de nombreuses disciplines. La sélection de variables est étudiée depuis une dizaine d'années et un certain nombre de méthodes ont émergé. Dans le présent travail nous exposons une nouvelle stratégie de classification hybride utilisant les réseaux de neurones et les réseaux bayésiens avec deux approches d'étude, appelées approche amont et approche aval, tel que le premier algorithme est un optimisateur de la base de donnée en éliminant le bruit des descripteurs non pertinents utilisant une méthode de sélection de variables, et le deuxième algorithme pour une meilleure classification des données résultantes de la première étape. Pour ce faire, nous employons deux méthodes connues de sélection de variable et nous proposons une nouvelle méthode couplée avec l'algorithme du réseau de neurones que nous baptisons MOYVAR. Pour s'assurer de l'exactitude des résultats, nous utilisons trois techniques d'évaluation de la classification a savoir : la technique classique apprentissage-test et les deux techniques de ré-échantillonnage le bootstrap 632+ et la validation croisée.

Enfin, le travail que nous présentons dans ce mémoire représente un créneau de recherche scientifique évolutif et présente des intérêts forts intéressants pour les systèmes décisionnels dans une perspective de régularisation d'un problème mal posé ou bien pour une amélioration effective de la classification notamment dans les domaines ou la prédiction pèse grand dans le comportement d'une société.

Mots clés : Data Mining, Extraction des Connaissances à partir des Données (ECD), Classification, Réseaux bayésiens, Réseaux de neurones, Hybridation, Apprentissage supervisé, techniques d'évaluation du taux d'erreur (ré-échantillonnage).

الملخص

البيانات التي يتم جمعها من خلال رصد ظاهرة أو قياسها في نظام فيزيائي لا تزودنا كلها بمعلومات ذات معنى: بعض المتغيرات قد تشكل ضوضاء، تكون قليلة التأثير على المعنى الكلي، مترابطة فيما بينها أو غير لازمة، و هذا ما يؤثر كثيرا على تصنيف المعلومات المجمعة. اختيار المتغيرات هي مشكلة معقدة وهو موضوع للبحث في عدة مجالات أين تمت مناقشته منذ عقد من الزمن حيث ظهرت عدة أساليب لدراسته. في هذا العمل نعرض إستراتيجية جديدة هجينة لتصنيف المعلومات باستخدام الشبكات العصبية و شبكات بايزية مع نهجين للدراسة، نهج المنع ونهج المصب، حيث تكون الخوازمية الأولى هي محسن لقاعدة البيانات عن طريق إزالة ضجيج المتغيرات الغير مؤثرة في التصنيف، والخوارزمية الثانية لتحسين تصنيف البيانات الناتجة عن المرحلة الأولى. للقيام بذلك، نستخدم طريقتين معروفتين لاختيار المتغيرات الجيدة و نقترح طريقة جديدة مقرونة بخوارزمية الشبكة العصبية و نسميها مويفار (**MOYVAR**). و لضمان دقة النتائج، نستخدم ثلاث تقنيات لتقييم التصنيف : التقنية الكلاسيكية تجريب–اختبار و تقنيتي الإختزال بوتسراب **632**+ و المصاقــة الملتقية.

وأخيرا ، فإن العمل الذي نقدمه يمثل مجالا جديدا للبحث العلمي الحديث و يظهر فائدة كبرى لنظم دعم اتخاذ القرارت بتسوية سوء طرح مشكلة على نحو فعال بأقل متغيرات ممكنة أو بالتحسين الفعلي للتصنيف خاصة في المجالات التي يكون فيها للشبؤ وزن كبير.

الكلمات الرئيسية : استخراج المعرفة، استخلاص المعلومات من البيانات، التصنيف ، شبكات بايزيه، الشبكات العصبية ، التهجين ، التجريب المقيد، التقييم التقني لنسبة الخطأ (الإختزال).

ABSTRACT

The data collected during observation of a phenomenon or measured on a physical system are not quite as informative: some variables may correspond to noise, be insignificant, ireelevant or correlated, which greatly affect their classification. The selection of variables is a complex problem and is the subject of research in many disciplines. The selection of variables has been discussed for a decade and a number of methods have emerged. In this work we expose a new strategy for hybrid classification using neural networks and bayesian networks with two approaches of study, called upstream approach and downstream approach, as the first algorithm is a optimizer of database by removing the noise of irrelevant descriptors using a method of variables selection, and the second algorithm for a better classification of data resulting from the first step. With this intention, we employ two known methods of variables selection and we propose a new method coupled with the algorithm of the neural network that we baptize MOYVAR. To ensure accuracy of results, we use three techniques for evaluating the classification : the classical technique learning-test and two techniques of resampling the bootstrap 632+ and the cross validation.

Finally, the work that we present represents an evolutionary niche of scientific research and has strong interests for decisional systems in case to regularize a problem badly set or to effectively improve the classification especially in the fields which the prediction press hard in the behavior of companies.

Key words: Data Mining, Knowledge Discovery in Database (KDD), Classification, Bayesian network, Neural network, Hybridization, Supervised learning, technical evaluation of error rate (resampling).

REMERCIEMENTS

Je tiens à remercier notre créateur **le tout miséricordieux** pour m'avoir permis d'achever ce travail.

Je tiens à adresser mes plus vifs remerciements à Mme OUKID-KHOUAS Salyha, ma directrice de thèse, pour sa rigueur scientifique et ses conseils avisés. Votre connaissance pointue des problématiques de recherches actuelles, et votre conscience des contraintes scientifiques m'ont permis de faire aboutir ce travail. Merci également pour avoir accepté de se pencher sur des problèmes parfois très éloignés de votre domaine de recherche, et pour l'aide que vous avez su m'apporter, tant vis-à-vis des questions posées, que vis-à-vis d'organisation du mémoire, et surtout pour m'avoir encouragé dans les moments les plus difficiles, sur le plan scientifique autant qu'humain : sans votre soutien, ce mémoire de thèse n'aurait sans doute pas vu le jour.

Je suis également très reconnaissant à Mr RICCO RAKOTOMALALA du laboratoire ERIC (France) pour tout le temps qu'il a su m'accorder, dans un agenda pourtant débordant. Son enthousiasme sans faille et les nombreuses discussions que nous avons eues ont permis d'éclairer de nombreux problèmes, allant de la procuration des BDD utilisées à la documentation des logiciels d'apprentissage. Il m'a aussi appris des choses qui dépassent même le cadre de la science, je tiens donc à lui manifester ici toute ma gratitude.

Une immense gratitude à mes parents qui n'ont pas cesser de me soutenir dans toute ma vie surtout par leurs encouragements et leurs prières. Comme je ne pourrais assez remercier mon épouse qui m'a soutenu tout au long du temps consacré à l'élaboration de ce travail avec sa patience et son encouragement comme je n'en ai jamais vu.

Je tiens à remercier aussi le président du jury et les membres du jury pour avoir accepté de juger ce travail.

Mes plus profonds remerciements pour tous mes amis et à toute personne ayant contribué de prêt ou de loin à ce travail

Tous sincères mes remerciements et mes profondes excuses à tous ceux que j'aurai pu oublier.

DEDICACES

Je dédie ce modeste travail à,

Mes chers parents et beaux parents,

Mon adorable épouse et mon petit Mohamed Imad,

Mes frères et sœurs,

Mes beaux frères et belles sœurs,

Toute la famille BOUDJEBBOUR et la famille BENZEKOUR,

Tous mes amis,

Tout les musulmans,

Et toute l'humanité.

TABLE DES MATIERES

CHAPITRE 4 : IMPLEMENTATION ET RESULTATS

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES EQUATIONS

INTRODUCTION

1. Contexte

L'ère que nous vivons actuellement est sans doute l'ère de l'informatique. Partout où nous allons, au travail, aux centres commerciaux, à la scolarité de l'université, dans les banques, dans les usines ou encore dans les laboratoires scientifiques nous trouvons toujours des machines de collecte d'information. Aujourd'hui les données sont saisies en vrac. Trois motivations principales poussent l'idée de garder ces données même si elles ont été déjà utilisées de point de vue opérationnel (pourquoi garder toutes les données sur les naissances dans un hôpital alors qu'elles datent depuis très longtemps et le nouveau né est devenu maintenant un adulte). La première motivation est le fait que les technologies de récupération et de stockage de données connaissent leur apogée depuis quelques décennies. Aujourd'hui la saisie d'une transaction nécessite quelques clics de souris, ou encore le passage d'une douchette sur le code à bar d'un produit. La deuxième motivation est le fait que le coût de stockage des données est en train de se réduire assez considérablement pour garder des téra-octets de données. La troisième motivation est la plus rationnelle. En effet, les personnes détenant ces données sont conscients que dedans existe une véritable mine d'or qu'on appelle connaissances qui est le fruit d'une analyse des données. Cette analyse ne peut être manuelle, car la taille des données est énorme. Il est évident que l'assistance des ordinateurs pour réaliser cette tâche serait la bienvenue. Le processus semi-automatique par lequel les données en masse se transforment en connaissances est appelé extraction de connaissances à partir de données (ECD). Ce terme est souvent confondu avec le Data Mining, et les deux mots sont fréquemment utilisés comme étant des synonymes.

Plusieurs sciences contribuent à l'essor que connaît aujourd'hui le Data Mining à savoir l'intelligence artificielle, les bases de données, l'apprentissage machine et les statistiques. Il faut savoir que la particularité du Data Mining par rapport à toutes les disciplines qui y interfèrent, est le fait qu'il traite une quantité énorme de données. Les solutions offertes par le Data Mining peuvent traiter des bases de données de plusieurs millions d'objets et d'attributs.

2. Contribution et Objectif

Les systèmes actuels de filtrage de l'information sont basés d'une façon directe ou indirecte sur des techniques traditionnelles de classification d'information (Méthode KNN, plus proche voisin, arbre de décision…). Notre approche consiste à séparer le processus de classification du filtrage proprement dit. Il s'agit d'effectuer un traitement reposant sur une compréhension primitive de l'information entrante permettant d'effectuer des opérations de classement des données en éliminant les descripteurs non pertinents. Ces données sont modélisées par un algorithme à l'aide d'un module d'apprentissage, l'algorithme est amélioré progressivement au fur et à mesure de son utilisation et le système devra choisir, ou plutôt trouver, parmi tout les descripteurs entrants, ceux qui classera au mieux la donnée qu'il reçoit. Cette amélioration est bien sûr définie par la nature du problème à résoudre. Donc, le but principal est de trouver une approche de classification à base de deux techniques connues améliorant l'interprétation des données à classer tout en améliorant la puissance de prédiction d'une nouvelle entrée.

Supposons qu'une banque lance une analyse sur sa base transactionnelle. Les connaissances acquises peuvent être l'analyse d'une transaction prédiction. Le service de crédit de cette banque est contraint de prédire les clients douteux et les clients sérieux avec des probabilités de ceux ne peuvent pas rembourser a temps. Le problème est que les descripteurs de ces clients sont multiples ce qui influe sur les résultats de la classification, donc il faut minimiser ces descripteurs en laissant seulement ceux qui influent sur le résultat de l'analyse et éliminer le bruit; aussi les données peuvent être manquantes ou erronées pour quelques enregistrements, ceci nous amènera a trouver un algorithme efficace pour ces problèmes de données.

3. Organisation du mémoire

Notre mémoire est organisé en quatre chapitres précédés par une introduction détaillée. Dans le premier et le deuxième chapitres nous présentons respectivement les détails sur les thèmes essentiels de notre étude : les réseaux de neurones et les réseaux bayésiens et leurs relations avec la classification. Le troisième chapitre présente l'approche proposée sur la classification hybride avec une explication détaillée du principe de fonctionnement de la solution trouvée. Enfin, le dernier chapitre est consacré à la partie implémentation et aux résultats obtenus après avoir appliqué la classification hybride sur des données réelles. Nous terminons notre mémoire par la conclusion de notre travail.

4. Le Data Mining

4.1 Présentation

Traduit littéralement par " forage des données ", le Data Mining est un processus non élémentaire de mises à jour de relations, corrélations, dépendances, associations, modèles, structures, tendances, classes, facteurs obtenus en navigant à travers de grands bases de données, navigation réalisée au moyen de méthodes mathématiques, statistiques ou algorithmiques **[1]**.

D'après Le Gartner Group **[40]**, 1996, ce processus peut être itératif et/ou interactif selon les objectifs à atteindre, et considère le Data Mining comme un processus (le plus automatisé possible) qui va des données élémentaires disponibles dans un Data Warehouse à la décision en apportant à chaque étape de ce processus une plus-value informationnelle qui peut aller jusqu'au déclenchement automatique d'actions en fonction de l'information de synthèse mise à jour. Nous comprenons, derrière le concept du Data Mining l'héritage de l'intelligence artificielle et des systèmes experts. Mais nous comprenon aussi l'utilisation des méthodes d'analyses des données qui ont pour objet de découvrir des structures, des relations entre faits au moyen de données élémentaires et de techniques mathématiques appropriées. Nous ne s'étonnerons donc pas de trouver au catalogue des méthodes de Data Mi*ning* aussi bien les réseaux de neurones, les réseaux bayésiens, les arbres dits de décision que les méthodes de visualisation multidimensionnelle.

4.2 Définition

Plusieurs définitions ont été proposées dans **[3]**, le Data Mining serait :

" la découverte de nouvelles corrélations, tendances et modèles par le tamisage d'un grand nombre de données ";

" un processus d'aide à la décision où les utilisateurs cherchent des modèles d'interprétation dans les données ";

" l'extraction d'informations originales, auparavant inconnues, potentiellement utiles à partir des données ";

" un processus de mise à jour de nouvelles corrélations, tendances et de modèles significatifs par un passage au crible des bases de données volumineuses, et par l'utilisation de modèles d'identification technique aussi bien statistiques que mathématiques ";

" le fait d’extraire automatiquement de la connaissance intéressante, intelligible et cachée dans les bases de données ";

SAS Institute définit le Data Mining comme " le processus d'exploration et de modélisation des gisements de données permettant de découvrir des informations/indicateurs inconnus pour obtenir des avantages concurrentiels " **[1]**.

Le terme de Data Mining signifie littéralement forage de données. Comme dans tout forage, son but est de pouvoir extraire un élément : la connaissance. Ces concepts s'appuient sur le constat qu'il existe au sein de chaque entreprise des informations cachées dans le gisement de données. Ils permettent, grâce à un certain nombre de techniques spécifiques, de faire apparaître des connaissances **[4]**.

Le Data Mining, ou la fouille de données est l'ensemble des méthodes et techniques destinées à l'exploration et l'analyse de grandes bases de données informatiques, de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles des associations, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information utile tout en réduisant la quantité de données permettant d'étayer les prises de décision **[4]**.

Généralement, on s'accorde à définir le Data Mining comme la découverte de connaissances dans les bases de données *(*Knowledge Discovery in Database - KDD*)*. Donc on peut dire que le Data Mining est un raisonnement mathématique qui permet d’analyser et d’interpréter un gros volume de données, de différentes sources, afin de dégager des tendances, de rassembler et classifier les éléments similaires en catégories et de formuler des hypothèses. Cette découverte englobe des outils statistiques mais, les méthodes statistiques classiques sont plus descriptives et confirmatives, tandis que les méthodes du Data Mining sont plus exploratoires et décisionnelles. En bref, le Data Mining est l'art d'extraire des informations (ou des connaissances) à partir des données.

Le Data Mining soit descriptif, soit prédictif.

- Les techniques descriptives (ou exploratoires) visent à mettre en évidence des informations présentes mais cachées par le volume de données.
- Les techniques prédictives (ou explicatives) visent à extrapoler de nouvelles informations à partir des informations présentes.

4.3 Objectifs

On peut regrouper les objectifs des méthodes de Data Mining en quatre grandes fonctions **[1]** :

- **Classifier** : on examine les caractéristiques d'un nouvel objet pour l'affecter à une classe prédéfinie. Les classes sont bien caractérisées et on possède un fichier d'apprentissage avec des exemples préclassés. On construit alors une fonction qui permettra d'affecter à telle ou telle classe un nouvel individu.
- **Estimer** : la classification se rapporte à des événements discrets (par exemple :le patient à été ou non hospitalisé). L'estimation, elle, porte sur des variables continues (par exemple : la durée d'hospitalisation).
- **Segmenter** : il s'agit de déterminer quelles observations vont naturellement ensemble sans privilégier aucune variable. On segmente une population hétérogène en un certain nombre de sous-groupes plus homogènes. Dans ce cas, les classes ne sont pas prédéfinies.
- **Prédire** : cette fonction est proche de la classification ou de l'estimation, mais les observations sont classées selon un comportement ou une valeur estimée futurs. Les techniques précédentes peuvent être adaptées à la prédiction au moyen d'exemples d'apprentissage où la valeur à prédire est déjà connue. Le modèle, construit sur les données d'exemples et appliqué à de nouvelles données, permet de prédire un comportement futur.

5. La Classification

5.1 Présentation

La classification est une branche de l'analyse statistique multidimensionnelle qui a fait l'objet de très nombreuses publications. Elle connaît, ces dernières années, un renouvellement et un développement considérables avec la multiplication de bases de données de plus en plus importantes. Les techniques de classification font appel à une démarche algorithmique et non à des techniques mathématiques complexes : les classes, obtenues après des opérations simples et répétitives, sont souvent faciles à décrire et à caractériser. Donc, classifier un ensemble d'objets, c'est attribuer à chacun une classe (ou catégorie) parmi plusieurs classes définies à l'avance. Cette tache est appelée Classification ou Discrimination. Un algorithme qui réalise automatiquement une classification est appelé classificateur.

Les statisticiens appellent aussi classification la tâche qui consiste à regrouper des données qui se ressemblent dans des classes qui ne sont pas définies à l'avance, il y a donc une certaine confusion dans les termes. Nous nous efforcerons toujours de préciser ce dont il s'agit, lorsque le contexte ne rend pas la distinction évidente. Dans notre étude, nous nous plaçons dans le cas où les classes sont connues à l'avance.

5.2 Exemples de classification

5.2.1 Avec les réseaux de neurones [43]

La modélisation non-linéaire de données statiques en bio-ingénierie

L'étude des relations structure-activité des molécules (QSAR pour Quantitative Structure-Activity Relations) est un domaine en plein essor, en raison des progrès très rapides de la simulation moléculaire. Ces travaux ont pour objectif de prédire certaines propriétés chimiques de molécules à partir de données (descripteurs) structurales (leur masse, leur volume, leur nombre d'atomes, les charges électriques portées par ceux-ci) qui peuvent être calculées a priori par ordinateur, sans qu'il soit nécessaire de synthétiser la molécule; on peut donc éviter une synthèse coûteuse si l'on peut prédire que la molécule envisagée ne possède pas les propriétés souhaitables. Cette approche est particulièrement utile dans le domaine de la bio-ingénierie, pour la prédiction de propriétés pharmacologiques de molécules et l'aide à la découverte de nouveaux médicaments, mais

elle peut évidemment être transposée à n'importe quel domaine tel que la prédiction de propriétés mécaniques de matériaux complexes à partir de leur formulation, la prédiction de paramètres thermodynamiques de mélanges par exemple prédire les propriétés mécaniques de caoutchoucs pour pneumatiques à partir de la composition des mélanges utilisés (collaboration Michelin – ESPCI),... etc.

La reconnaissance de formes

Dans le domaine de la reconnaissance de formes, la classification automatique tient un rôle important. Or, en raison de leur propriété d'approximateurs universels, les réseaux de neurones sont susceptibles d'estimer de manière précise la probabilité d'appartenance d'un objet inconnu à une classe parmi plusieurs possibles. Ainsi, des systèmes de lecture des codes postaux, utilisant des réseaux de neurones, sont opérationnels dans les centres français de tri postal. Plus difficile encore, un système de lecture automatique des montants écrits en toutes lettres sur les chèques a été conçu et réalisé par la Société A2iA ; il est en service dans des banques.

Prévision des pics de pollution par l'ozone

La généralisation des mesures de concentration en ozone, ainsi que le développement de modèles de connaissance de la pollution atmosphérique, permettent d'envisager la prévision des pics de pollution. Dans le cadre d'un groupe de travail du club « Ingénierie du traitement de l'information » de l'association ECRIN, des données relatives à la pollution par l'ozone dans la région lyonnaise ont été mises à la disposition des équipes de recherche françaises, en vue d'une prédiction « boîte noire » à l'aide de méthodes mettant en jeu un apprentissage. Les réseaux de neurones étaient donc des candidats naturels pour réaliser cette tâche. Comme il s'agissait d'une étude préliminaire de courte durée, ils se sont contentés d'utiliser les données issues d'un seul capteur d'ozone, pour lequel les données disponibles (mesures heure par heure pendant les années 1995 à 1998) étaient fiables. Les données des années 1995 à 1997 ont été utilisées pour l'apprentissage, celles de l'année 1998 pour le test. L'objectif est de prévoir, 24 heures à l'avance, si la pollution dépassera le seuil d'alerte (180 µg/m3 au moment où l'étude a été effectuée).

5.2.2 Avec les réseaux bayésiens [6]

Diagnostic médical

Les premières applications des réseaux bayésiens ont été développées dans le

domaine du diagnostic médical. Les réseaux bayésiens sont particulièrement adaptés à ce domaine parce qu'ils offrent la possibilité d'intégrer des sources de connaissances hétérogènes (expertise humaine et données statistiques), et surtout parce que leur capacité à traiter des requêtes complexes (explication la plus probable, action la plus appropriée) peuvent constituer une aide véritable et interactive pour le praticien. Le système Pathfinder, développé au début des années 1990 a été développé pour fournir une assistance au diagnostic histopathologique, c'est-à-dire basé sur l'analyse des biopsies. Il est aujourd'hui intégré au produit Intellipath, qui couvre un domaine d'une trentaine de types de pathologies. Ce produit est commercialisé par l'éditeur américain Chapman et Hall, et a été approuvé par l'American Medical Association. Dans le domaine de la santé, une application intéressante des algorithmes issus des réseaux bayésiens a permis d'améliorer considérablement la recherche de la localisation de certains gènes, dans le cadre du projet Human Genome **[55]**.

Domaine des Banques et Finances

Les applications dans le domaine de la banque et de la finance sont encore rares, ou du moins ne sont pas publiées. Mais cette technologie présente un potentiel très important pour un certain nombre d'applications relevant de ce domaine, comme l'analyse financière, le scoring, l'évaluation du risque, ou la détection de fraudes. En premier lieu, les réseaux bayésiens offrent un formalisme unifié pour la manipulation de l'incertitude, autrement dit du risque, dont la prise en compte est essentielle dès qu'il s'agit de décision financière. Récemment, les nouveaux accords de Bâle II ont ouvert un nouveau champ d'application très significatif pour les réseaux bayésiens dans le domaine bancaire. Ces accords fixent les nouvelles règles que doivent appliquer les banques pour la détermination de leurs exigences en fonds propres. Ces fonds propres doivent être dimensionnés de façon à couvrir à un niveau de probabilité élevé les différents types de risques encourus par la banque : risques de crédit, risques de marché et risques opérationnels.

5.3 Travaux d'hybridation

Diagnostic dans un système complexe (Réseau Téléphonique) [18]

Le suivi des systèmes industriels est nécessaire pour prévenir les incidents, détecter des anomalies et maintenir une bonne qualité de service. La complexité croissante de ces systèmes a motivé des efforts importants destinés à développer des méthodes de suivi automatique et de diagnostic. Cette étude est l'oeuvre de Philippe Leray et Patrick Gallinari **[18]** par l'utilisation

d’une architecture hybride de diagnostic à partir de méthodes d'apprentissage numérique (réseaux de neurones et réseaux bayésiens) permettant de répondre à différentes tâches du diagnostic comme la sélection d'indicateurs pertinents, la génération d'alarmes et plus particulièrement la prise en compte des dépendances temporelles et spatiales qui existent dans un système complexe. Cette architecture, appliquée dans le cadre de la gestion en temps réel du trafic téléphonique français, permet d'effectuer un filtrage d'alarmes spatio-temporel. L’architecture de diagnostic est partagée en deux parties. La première étape consiste à générer localement des alarmes correspondant à différentes perturbations dans le système ou elle montre comment les réseaux de neurones peuvent résoudre plusieurs problématiques associées à la sélection des indicateurs de trafic pertinents, le déclenchement des alarmes et le filtrage temporel de ces alarmes. La dernière étape consiste à utiliser les dépendances spatiales qui existent dans le réseau téléphonique pour réaliser un filtrage spatial des alarmes en utilisant un réseau bayésien permettant de prendre en compte les dépendances entre les différentes alarmes et donnant de bons résultats lorsque les alarmes locales sont manquantes ou incertaines.

Reconnaissance de Mots Manuscrits sur un Grand Vocabulaire [17]

Ce travail, réalisé par quelques chercheurs du laboratoire d’imagerie, de vision et d’intelligence artificielle de l’école de technologie supérieure de Montréal en collaboration avec le laboratoire de reconnaissance de formes et vision de Lyon, présente un système de reconnaissance hybride qui intègre des modèles de Markov cachés (MMC) et des réseaux neuronaux (RN) dans une architecture probabiliste. Les mots manuscrits sont d’abord traités par un système de reconnaissance de mots basé sur des MMC guidé par un lexique. Une liste des N meilleures hypothèses ainsi que de la segmentation de ces mots en caractères est générée. Un classificateur à base de réseau neuronal calcule un score pour chaque caractère segmenté et les résultats des classificateurs MMC et RN sont combinés pour optimiser les performances de reconnaissance. Les résultats expérimentaux montrent que sur un vocabulaire de 80,000 mots le système hybride MMC/RN augmente le taux de reconnaissance de 9% relativement au système de reconnaissance MMC seul.

CHAPITRE 1 :

RESEAUX DE NEURONES

CHAPITRE 1
RESEAU DE NEURONES

1. Introduction

On peut dire que parmi les buts essentiels de la recherche scientifique est de développer des machines intelligentes qui peuvent exécuter toute tache pénible et encombrante. Parmi les technologies qui sont consacrées à ce type de recherche : l'intelligence artificielle et les systèmes de neurones artificiels. Ces derniers sont basés essentiellement sur le mécanisme de transmission nerveuse d'un être humain. L'élément fonctionnel essentiel du système nerveux est la cellule nerveuse ou neurone qui a pour rôle d'élaborer l'information reçue et transmettre les résultats a d'autres neurones, Le cerveau humain développe mieux les solutions intelligentes qu'un ordinateur, cependant ce dernier est rapide dans l'exécution des opérations. Les différences entre l'ordinateur et le cerveau humain sont dues à l'architecture de chacun et les méthodes du traitement correspondantes. En vue de traitement de l'information, l'ordinateur utilise des programmes basés sur des algorithmes. Ces derniers opèrent avec des séquences d'instructions contrôlées par une unité centrale complexe, afin d'aboutir à un résultat en fonction des données emmagasinées dans des mémoires. Tandis que le cerveau utilise la notion de transformation, des représentations distribuées et parallèles. Ce dernier met, en communication des milliards des neurones.

Les réseaux de neurones sont des structures (la plu part de temps simulées par des algorithmes exécutés sur des ordinateurs d'usage générale, parfois sur des machines ou même des circuits spécialisés) qui prennent leur inspiration (souvent de façon assez lointaine) dans le fonctionnement des systèmes nerveux. Leur domaine d'application est essentiellement celui de résoudre les problèmes de classification, d'association, de reconnaissance de forme, d'extraction des caractéristiques et d'identification Les origines de cette discipline sont très diversifiées; En 1943, WARREN MCCULLOCH & WALTER PITTS ont proposé le premier modèle d'un système de neurones artificiels, qui est encore largement utilisé pour expliquer comment le cerveau peut réaliser les fonctions logiques.

En 1949, DONALD HEBB décrit une règle sur l'apprentissage **[5]**. Après plusieurs développement dans les modèles des réseaux de neurones, WEBBOS a développé en 1974 un algorithme nommé algorithme de rétropropagation, ce qui a encouragé d'autres chercheurs à reprendre la recherche dans ce domine après longue période.

2. Définition

Un réseau de neurones est un ensemble de méthodes d'analyse et de traitements des données permettant de construire un modèle de comportement à partir de données qui sont des exemples de ce comportement. Un réseau de neurones est constitué d'un graphe pondéré orienté dont les nœuds symbolisent les neurones. Ces neurones possèdent une fonction d'activation qui permet d'influencer les autres neurones du réseau. Les connexions entre les neurones, nommés liens synaptiques, propagent l'activité des neurones avec une pondération caractéristique de la connexion. Nous appelons poids synaptique la pondération des liens synaptiques. Les neurones peuvent être organisés de différentes manières, c'est ce qui définit l'architecture et le modèle du réseau. L'architecture la plus courante est celle dite du perceptron multicouche **[8]**.

3. Applications

Les réseaux de neurones sont essentiellement utilisés pour faire de la classification. Construit à partir d'exemples de chaque classe qu'il a appris, un réseau de neurones est normalement capable de déterminer à quelle classe appartient un nouvel élément qui lui est soumis **[4]**.

4. Fonctionnement

• La construction de la structure du réseau (généralement empirique).

• La constitution d'une base de données de vecteurs représentant au mieux le domaine à modéliser. Celle-ci est scindée en deux parties : une partie servant à l'apprentissage du réseau (on parle de base d'apprentissage) et une autre partie aux tests de cet apprentissage (on parle de base de test).

• Le paramétrage du réseau par apprentissage. Au cours de l'apprentissage, les vecteurs de données de la base d'apprentissage sont présentés séquentiellement et plusieurs fois au réseau. Un algorithme d'apprentissage ajuste le poids du réseau afin que les vecteurs soient correctement appris. L'apprentissage se termine lorsque l'algorithme atteint un état stable.

• La phase de reconnaissance qui consiste à présenter au réseau chacun des vecteurs de la base de test. La sortie correspondante est calculée en propageant les vecteurs à travers le réseau. La réponse du réseau est lue directement sur les unités de sortie et comparée à la réponse attendue. Une fois que le réseau présent des performances acceptables, il peut être utilisé pour répondre au besoin qui a été à l'origine de sa construction **[4].**

5. Modèle biologique

5.1 Définition et structure

Le bloc principal du système nerveux est le neurone. Il transmet l'information reçue vers les diverses parties du corps. Il est constitué **[5]** :

• D'un corps cellulaire nommé somma

• Des plusieurs épines semblables propagées dans le corps cellulaires nommées *dendrites.* Leur rôle est de capter les signaux qui proviennent du neurone.

• D'une seule fibre nerveuse nommé axone, qui sert à connecter le corps cellulaire aux autres neurones. L'axone est un moyen de transport pour les signaux émis par le neurone.

• Les connexions entre les neurones se font par l'intermédiaire du corps cellulaire ou les dendrites en jonctions nommées synapses. Les synapses servent à limiter plus ou moins l'amplitude des signaux qui passent d'un neurone à un autre, comme est illustré dans la figure 2.1.

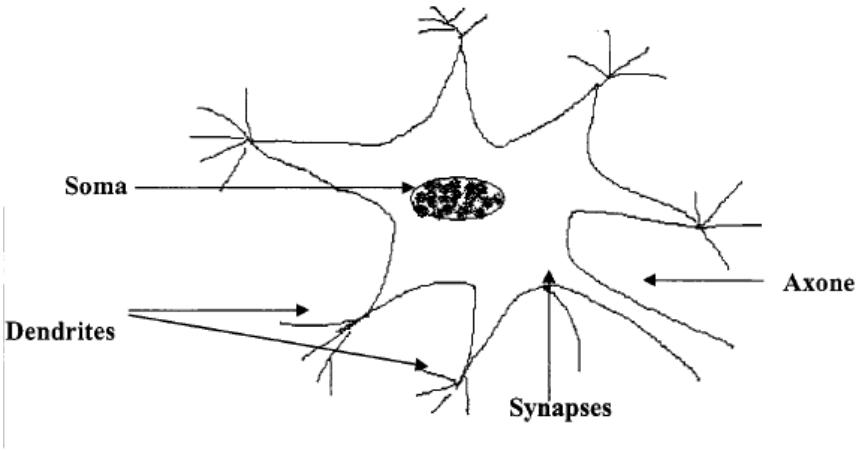

Figure 1 .1 : Représentation simplifiée de neurone biologique

5.2 Fonctionnement

Le mécanisme de fonctionnement d'un neurone est de recevoir, grâce à ces dendrites, les signaux émis par les autres neurones, puis décider, à partir des données reçues,

d'émettre ou non un signal à ses semblables le long de son axone. Plus précisément, le soma recueille l'ensemble des informations reçues par les dendrites et effectue la sommation dite spatio-temporelle. En raison de sa dimension, l'intégration somatique est aussi temporelle. Si le potentiel somatique dépasse un certain seuil, il y a émission d'un potentiel d'action ou spike. Le signal, très bref (1ms), est transmis sans atténuation le long de l'axone et réparti sur le neurone cible **[5]**.

5.3 Plasticité synaptique (règle de HEBB)

DONALD HEBB introduit la notion de plasticité synaptique, c'est à dire le mécanisme de modification progressive des couplages entres neurones. D'après HEBB, le renforcement synaptique intervient lorsqu'il y a activité conjointe du neurone prè-synaptique et du neurone post-synaptique, ce qui implique chaque neurone présente deux états *(*actif ou inactif*).* D'après cette règle, l'efficacité synaptique augmente seulement si les deux éléments sont actifs simultanément, donc elle prévoit exclusivement le renforcement des efficacités synaptiques, c'est à dire que le poids de la synapse ne peut qu'augmenter, chose qui conduit à une fatale saturation du réseau. Nous sommes donc, obligés de préciser un certain intervalle de coïncidence.

6. Étude et synthèse d'un réseau de neurone formel

La plus satisfaisante définition d'un réseau de neurone formel, est de celle de HIECHT NILSON : « un réseau de neurone est une structure de traitement parallèle et distribué d'informations comportant plusieurs éléments de traitement Neurone, qui peuvent posséder des mémoires locales et exécuter les opérations de traitements sur des informations locales. Ils sont interconnectés les uns aux autres avec des canaux des signaux unidirectionnels ».

La synthèse d'un réseau de neurone formel est basée sur des caractéristiques similaires à celle d'un réseau de neurone biologique **[3]**, Ces caractères sont :

- II est composé d'un nombre très grand d'éléments de traitement simple.
- Chaque élément de traitement est connecté à plusieurs éléments voisins.
- Le fonctionnement d'un réseau est basé sur le mécanisme de modification de poids de connexion pendant la phase d'apprentissage.

6.1 Neurone formel

Un neurone formel est un petit automate qui réalise la somme pondérée des poids W_1, W_2, ...,W_n des entrées X_1, X_2,..... X_n qu'il reçoit du reste du réseau. Chaque nœud du réseau a un niveau d'activation numérique qui lui est associé au temps T. Ce niveau d'activation est modifié, à chaque période, par la quantité totale d'activation qu'il reçoit de ses voisins en entrée. La figure 2.2 suivante montre la structure d'un neurone formel **[8]** :

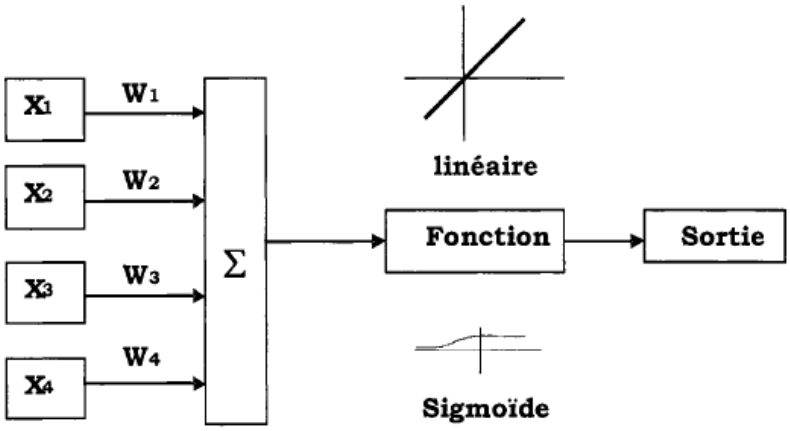

Figure 1.2 : Structure d'un neurone formel

6.2 Fonction d'activation

Afin de déterminer une valeur en sortie, une fonction appelée fonction d'activation (ou de transfert), est appliquée à cette valeur. La fonction d'activation la plus généralement rencontrée est une fonction sigmoïde telle que « si la somme des entrées est supérieure à un seuil, alors le neurone de sortie est activé; sinon, rien ». La majorité des modèles utilisés aujourd'hui préfèrent employer des fonctions d'activations continues, qui permettent de communiquer et de traiter plus d'informations à la fois dans un seul neurone. Ceci a pour conséquence d'augmenter la puissance de calcul des réseaux.

7. Structure des réseaux de neurones

La structure du réseau de neurones, encore appelée architecture ou topologie du réseau de neurones, est le nombre de couches et de nœuds, la façon dont sont interconnectés les différents nœuds (choix des fonctions de combinaison et de transfert) et le mécanisme d'ajustement des poids.

7.1 Réseau mono-couche et réseau multi-couches

Nous savons que l'organisation d'un réseau de neurone est constituée de couches, c'est à dire un tel réseau peut contenir une ou plusieurs couches.

7.1.1 Réseau mono-couches

Dans ce type de réseau, il y a une seule couche cachée, qui relie les cellules d'association (couche d'entrée) aux cellules de décision (couche de sortie). C'est la seule couche de connexion modifiable. Les neurones de la couche d'entrée d'un réseau mono-couche (perceptron) effectuent seulement un prétraitement et la classification effective est effectuée par les neurones de la couche de sortie. Ce réseau offre une grande convergence vers la solution du problème, malheureusement sa stratégie d'apprentissage n'offre que des séparations linéaires, limitées à la seule classe de problèmes linéairement séparables.

7.1.2 Réseau multi-couches

Pour surmonter les limitations d'un réseau mono-couche , on utilise un réseau multi-couche, où la sortie n'est connectée à l'entrée qu'après quelques couches de neurones intermédiaires apportant une richesse à la structure pour accroître la capacité de réseau. Notons que les couches internes n'ont aucune connexion prédéfinie, elles servent seulement à contribuer à l'obtention de résultats souhaités à la sortie. Le problème de séparation linéaire est donc résolu. Pour obtenir une séparation linéaire, on doit tenir compte de ce qui a été dit plus haut d'une part, et le bon dimensionnement en utilisant le modèle de rétropropagation d'autre part.

7.2 Réseaux récurrents et réseaux non récurrents

Les réseaux de neurones sont répartis en deux grandes classes **[8]** :

7.2.1 Les réseaux non récurrents (statiques)

Dans ce type de réseau, on utilise une structure à couche. Les neurones de la même couche ne sont pas connectés, chaque couche reçoit des signaux de la couche précédente et transmet le résultat de ces traitements à la couche suivante. En conséquence le signal d'entrée prend un sens unique de l'entrée vers la sortie tout en ayant traversé des couches

cachées. Un réseau de neurones statique est généralement organisé en plusieurs couches de neurones appelées réseaux multicouches comme il est illustré dans la figure 2.3.

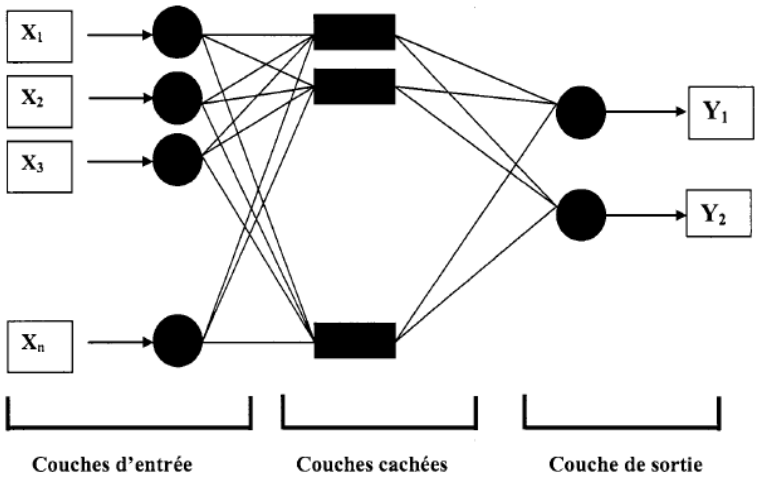

Figure 1.3 : Structure d'un réseau statique

Un réseau statique est constitué par :

• Une couche d'entrée qui reçoit ses signaux d'entrée du milieu externe.

• Une ou plusieurs couches cachées (intermédiaire)

• Une couche de sortie qui fournit les résultats de traitement du réseau.

7.2.2 Les réseaux récurrents (dynamiques)

Dans ce type de réseaux, les neurones sont entièrement connectés et les sorties des neurones de la couche sortie sont réinjectées sur les entrées des neurones précédents d'où l'existence d'une boucle de retour, cette dernière à pour rôle d'équilibrer le système lorsqu'il est soumis à un stimulus extérieur. Les décisions ne sont pas présentes instantanément, mais par étapes successives. La structure d'un réseau dynamique est donnée par la Figure 2.4 synoptique de la figure suivante :

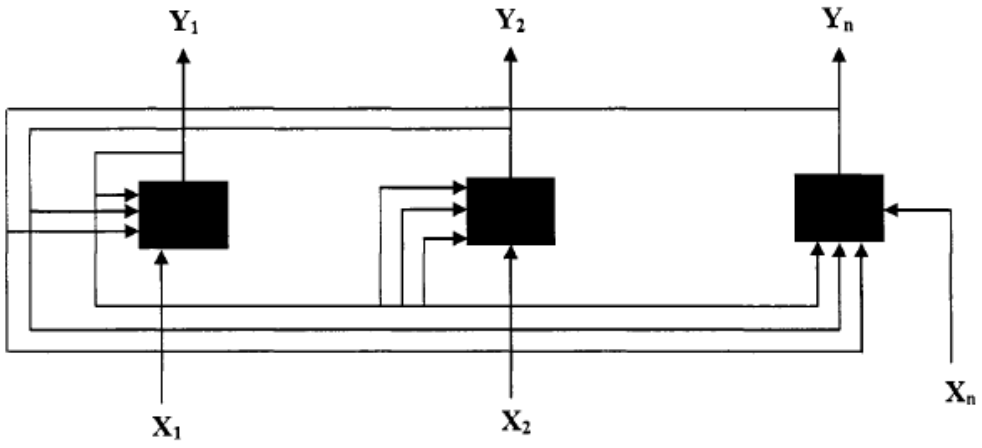

Figure 1.4 : Structure d'un réseau dynamique

7.3 Fonctionnement d'un réseau

Un réseau de neurone peut fonctionner en deux modes, parallèle ou séquentiel. Dans le mode parallèle tous les neurones calculent leurs nouvelles activations et leurs sorties, et les transmettent aux neurones auxquels ils sont connectés, à chaque top d'horloge. Contrairement, au mode séquentiel, un seul neurone calcule sa nouvelle activation et sa sortie puis les transmet aux neurones auxquels il est connecté, à chaque top d'horloge. Donc, le calcule est fait en fonction des entrées des neurones au top d'horloge précédent. Nous pouvons obtenir d'autres modes dits mixtes en combinant les deux modes précédents.

7.4 Apprentissage

L'apprentissage est définit par la modification des interactions entre neurones, donc, l'apprentissage consiste à ajouter les poids synaptiques de telle façon que le réseau présente un certain comportement désiré. Les procédures d'apprentissage peuvent se subdiviser, en deux grandes catégories : Apprentissage supervisé ou apprentissage non supervisé.

7.4.1 Apprentissage supervisé

L'apprentissage supervisé implique l'existence d'un professeur qui a pour rôle d'évaluer le succès ou l'échec du réseau quand nous lui présentons un stimulus connu. Cette supervision consiste à renvoyer au réseau une information lui permettant de faire évoluer ses connections afin de faire diminuer son taux d'échec. C'est à dire que ce professeur présente au réseau de neurones une entrée et la sortie désirée correspondante, pour faire la comparaison avec les sorties actuelles des vecteurs d'entrées. A partir de l'erreur calculée, les poids sont ajustés pour avoir des sorties correspondantes aux réponses désirées. Ce calcul se répète jusqu'à ce que l'erreur soit minimale par rapport à un critère préalable, et par conséquent les coefficients synaptiques prennent les valeurs optimales.

7.4.2 Apprentissage non supervisé

Les réseaux, utilisant l'apprentissage non supervisé, sont souvent appelés auto-organisateurs, ou encore à apprentissage compétitif. Dans ce type d'apprentissage la connaissance de la sortie désirée n'est pas nécessaire c'est à dire que le réseau s'auto-organise et organise les entrées qui sont présentées de façons à optimiser un critère de coût donné.

7.5 Choix de l'échantillon d'apprentissage

L'apprentissage du réseau de neurones sera d'autant meilleur qu'il s'effectuera sur un échantillon suffisamment riche pour représenter toutes les valeurs possibles de nœuds de toutes les couches du réseau, c'est-à-dire en particulier toutes les modalités possibles de chaque variable, en entrée ou en sortie. Il faut aussi veiller à ce que les enregistrements analysés ne soient pas triés selon un ordre significatif.

7.6 Normalisation des données

Les données utilisées dans un réseau de neurones doivent être numériques et leurs modalités comprises dans l'intervalle [0,1], ce qui implique, quand ce n'est pas le cas, une normalisation des données. Pour que le travail de normalisation soit correct, il faut, bien entendu, que le jeu de données d'apprentissage couvre toutes les valeurs rencontrées dans la population toute entière, et, en particulier, les valeurs extrêmes des variables continues.

7.6.1 Variables continues

Même en les normalisant, les variables continues peuvent connaître le problème d'écrasement des valeurs extrêmes des valeurs normales. Plusieurs moyens existent pour bien normaliser ce type de variable. Nous pouvons discrétiser la variable et la remplacer, par exemple, par ses quartiels. Nous pouvons normaliser, non pas la variable, mais le logarithme de cette variable, qui distend le début de l'échelle. On peut normaliser la variable linéairement, pour ses valeurs comprises entre -3 et +3 fois l'écart type σ autour de la moyenne μ, et envoyer les valeurs à $\mu - 3\sigma$ sur 0, et les valeurs supérieures à $\mu + 3\sigma$ sur 1 **[5]**.

7.6.2 Variables catégoriques

Un moyen fréquemment utilisé pour obvier à cette difficulté est d'avoir autant de nœuds que de modalités des variables catégoriques, en créant des variables binaires (appelées indicatrices) dont la valeur 1 ou 0 signifie que la variable catégorique a ou non cette modalité.

Remarque : Avant d'utiliser un réseau de neurones sur des données catégoriques, il faut donc réduire le plus possible le nombre de modalités de ces données.

7.7 Les principaux réseaux de neurones

Les réseaux de neurones diffèrent selon :

- Les neurones utilisés
- La structure du réseau
- Le mode de calcul

II existe différents modèles de réseaux de neurones. Les principaux, le perceptron multicouches (PMC : Multi Layer Perceptron), le réseau à fonction radiale RBF (Radial Basis Function) et le réseau de Kohonen. Les réseaux de neurones PMC et RBF sont des réseaux à apprentissage supervisé car ils appartiennent à la famille des techniques prédictives. Au contraire, le réseau de Kohonen est un réseau à apprentissage non supervisé. Il cherche à segmenter la population en groupes distincts rassemblant des éléments similaires : il appartient à la famille de techniques descriptives **[5]**.

8. Développement d'un réseau de neurones

Le cycle classique de développement peut être séparé en sept étapes **[5]** :

1. La collecte des données,
2. L'analyse des données,
3. La séparation des bases de données,
4. Le choix d'un réseau de neurones,
5. La mise en forme des données,
6. L'apprentissage,
7. La validation.

8.1 Collecte des données

L'objectif de cette étape est de recueillir des données, à la fois pour développer le réseau de neurones et pour le tester. Dans le cas d'applications sur des données réelles, l'objectif est de rassembler un nombre de données suffisant pour constituer une base représentative des données susceptibles d'intervenir en phase d'utilisation du système neuronal. La fonction réalisée résultant d'un calcul statistique, le modèle qu'il constitue n'a de validité que dans le domaine où on l'a ajusté. En d'autres termes, la présentation de données très différentes de celles qui ont été utilisées lors de l'apprentissage peut entraîner une sortie totalement imprévisible.

8.2 Analyse des données

II est souvent préférable d'effectuer une analyse des données de manière à déterminer les caractéristiques discriminantes pour détecter ou différencier ces données. Ces caractéristiques constituent l'entrée du réseau de neurones. Notons que cette étude n'est pas spécifique aux réseaux de neurones, quelque soit la méthode de détection ou de classification utilisée, il est généralement nécessaire de présenter des caractéristiques représentatives. Cette détermination des caractéristiques a des conséquences à la fois sur la taille du réseau (et donc le temps de simulation), sur les performances du système (pouvoir de séparation, taux de détection), et sur le temps de développement (temps d'apprentissage). Une étude statistique sur les données peut permettre d'écarter celles qui sont aberrantes et redondantes. Dans le cas d'un problème de classification, il appartient à l'expérimentateur de déterminer le nombre de classes auxquelles ses données appartiennent et de déterminer pour chaque donnée la classe à laquelle elle appartient.

8.3 Séparation des bases de données

Afin de développer une application à base de réseaux de neurones, il est nécessaire de disposer de deux bases de données : une base pour effectuer l'apprentissage et une autre pour tester le réseau obtenu et déterminer ses performances. Il n'y a pas de règle pour déterminer ce partage de manière quantitative. Il résulte souvent d'un compromis tenant compte du nombre de données dont on dispose et du temps imparti pour effectuer l'apprentissage. Chaque base doit cependant satisfaire aux contraintes de représentativité de chaque classe de données et doit généralement refléter la distribution réelle, c'est à dire la probabilité d'occurrence des diverses classes.

8.4 Choix d'un réseau de neurones

II existe un grand nombre de types de réseaux de neurones, avec pour chacun des avantages et des inconvénients. Le choix d'un réseau peut dépendre :

• De la tâche à effectuer (classification, association, contrôle de processus, séparation aveugle de sources...),

• De la nature des données,

• D'éventuelles contraintes d'utilisation temps-réel (certains types de réseaux de neurones, tels que la 'machine de Boltzmann', nécessitant des tirages aléatoires et un nombre de

cycles de calculs indéfini avant stabilisation du résultat en sortie, présentent plus de contraintes que d'autres réseaux pour une utilisation temps-réel),

• Des différents types de réseaux de neurones disponibles dans le logiciel de simulation que l'on compte utiliser.

Ce choix est aussi fonction de la maîtrise ou de la connaissance que nous avons de certains réseaux, ou encore du temps dont nous disposons pour tester une architecture prétendue plus performante.

8.5 Mise en forme des données pour un réseau de neurones

De manière générale, les bases de données doivent subir un prétraitement afin d'être adaptées aux entrées et sorties du réseau de neurones. Un prétraitement courant consiste à effectuer une normalisation appropriée, qui tienne compte de l'amplitude des valeurs acceptées par le réseau.

8.6 Apprentissage du réseau de neurones

Tous les modèles de réseaux de neurones requièrent un apprentissage. Plusieurs types d'apprentissages peuvent être adaptés à un même type de réseau de neurones. Les critères de choix sont souvent la rapidité de convergence ou les performances de généralisation. Le critère d'arrêt de l'apprentissage est souvent calculé à partir d'une fonction de coût, caractérisant l'écart entre les valeurs de sortie obtenues et les valeurs de références (réponses souhaitées pour chaque exemple présenté).

Les techniques de réechantinnage, qui seront précisés par la suite, permetent un arrêt adéquat de l'apprentissage pour obtenir de bonnes performances de généralisation. Certains algorithmes d'apprentissage se chargent de la détermination des paramètres architecturaux du réseau de neurones. Si nous n'utilisons pas ces techniques, l'obtention des paramètres architecturaux optimaux se fera par comparaison des performances obtenues pour différentes architectures de réseaux de neurones. Des contraintes dues à l'éventuelle réalisation matérielle du réseau peuvent être introduites lors de l'apprentissage.

8.7 Validation

Une fois le réseau de neurones entraîné (après apprentissage), il est nécessaire de le tester sur une base de données différente de celles utilisées pour l'apprentissage. Ce test permet à la fois d'apprécier les performances du système neuronal et de détecter le type de

données qui pose problème. Si les performances ne sont pas satisfaisantes, il faudra soit modifier l'architecture du réseau, soit modifier la base d'apprentissage.

9. Le perceptron multicouche

Les réseaux de neurones du type perceptron multicouche constituent sans doute l'architecture neuronale la plus utilisée dans le domaine des réseaux de neurones formels. En effet, ces réseaux ont été utilisés pour la résolution de problèmes très variés : la reconnaissance de formes, la détection de pannes, la prévision temporelle, le traitement d'images, le traitement du signal, la prédiction etc. Les performances obtenues en utilisant ces réseaux constituent l'une des principales raisons de l'intérêt croissant pour les réseaux de neurones artificiels.

9.1 Définition

Le perceptron est un modèle de réseau de neurones avec algorithme d'apprentissage créé par FRANK ROSENBLATT en 1958.

9.2 Architecture

Un réseau multicouche de neurones est constitué de plusieurs couches de neurones formels adaptatifs comme il est illustré à la figure 2.5, La principale difficulté pour les chercheurs résidait dans l'absence d'un algorithme pour corriger les poids des couches cachées, étant donné qu'on ne disposait pas pour ces neurones d'un signal d'erreur. Le seuil signal d'erreur qui pouvait être calculé était celui pour la couche de sortie puisqu'on connaît la valeur désirée pour chacun des neurones de sortie et la valeur effectivement affichée pour ces neurones.

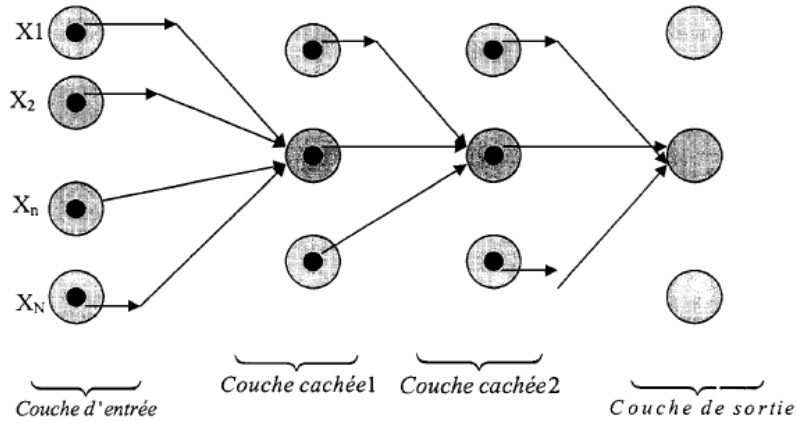

Figure 1.5 : Architecture d'un réseau multicouche de neurones

La première couche, ou couche d'entrée, a pour seule mission de présenter le vecteur associé à une forme à l'entrée du réseau. Par conséquent, les neurones qui la composent ne sont pas de véritables neurones du modèle de MCCULLOCH et PITTS **[56]**. Chaque neurone ne possède qu'une seule entrée, une valeur de polarisation nulle, une fonction d'activation linéaire et une fonction de sortie également linéaire. La valeur du poids de chaque connexion est constante et unitaire. En revanche, toutes les couches suivantes sont composées de neurones du modèle de MCCULLOCH et PITTS , La dernière couche est appelée la couche de sortie. Toutes les couches comprises entre la couche d'entrée et celle de sortie portent le nom de couches cachées et sont numérotées d'une manière séquentielle. Dans la littérature anglo-saxone, cette architecture porte le nom du MultiLayer Perceptron**[56]**. Elle correspond à une certaine réalité biologique, car la couche d'entrée peut être assimilée à la rétine, la couche de sortie à la prise de décision et les couches cachées aux différents niveaux de traitement de l'information visuelle.

9.3 L'algorithme de la rétropropagation

Plusieurs algorithmes ont été proposés pour l'apprentissage supervisé des poids synaptiques d'un réseau multicouche. La rétropropagation du signal d'erreur est l'algorithme le plus utilisé, sans doute grâce aux résultats obtenus avec cet algorithme. Désigné couramment en anglais par le terme « backpropagation », il est une généralisation de l'algorithme de WIDROW-HOFF pour un réseau multicouche. Il a d'abord été mis au point par [WERBOS, 1974] dans le cadre de sa thèse de doctorat, et donc faiblement diffusé dans la communauté scientifique qui n'y a pas porté attention en cette période où la recherche en Intelligence Artificielle était surtout orientée vers la paradigme symbolique. L'algorithme de rétropropagation du signal d'erreur a par la suite été redécouvert simultanément et indépendamment par [LE CUN, 1985], et [RUMELHART, HINTON & WILLIAMS, 1986],

9.3.1 Fonction de sortie

Une particularité de cet algorithme consiste à utiliser une fonction non linéaire du type sigmoïde au lieu de la fonction seuil utilisée dans le modèle de MCCULLOCH et PITTS. Cela a pour avantage de faciliter le calcul des différentes dérivées associées à l'évaluation des facteurs de certain poids durant la phase rétropropagation sans toutefois apporter de grandes modifications au modèle de base du neurone formel adaptatif.

La fonction sigmoïde la plus souvent utilisée a pour expression :

$$y = f(a) = \frac{1}{1+e^{-\sigma a}} \qquad \text{..... (1)}$$

Avec a : la valeur d'activation du neurones

et σ : le facteur de pente de la sigmoïde

Plus la valeur « *a* » est grande, plus la fonction sigmoïde s'approche de la fonction seuil.

La dérivée de la fonction de sortie du neurone est nécessaire au calcul du gradient. La dérivée de la fonction sigmoïde devient très simple à calculer lorsqu'on se rappelle que la fonction exponentielle est la seule fonction dont la dérivée est égale à elle-même. Des manipulations simples permettent d'exprimer la dérivée de la sigmoïde comme une fonction de la sortie seulement :

$$y' = f'(a) = \sigma y(1-y) \qquad \text{..... (2)}$$

9.3.2 Base d'apprentissage

Pour réaliser l'apprentissage des poids synaptiques d'un réseau perceptron multicouche, nous disposons d'une base d'apprentissage comportant K Couples :

$$B = \{(X_k, D_k, k=1,2,\text{……..},K)\}$$

Où $X_k=[x_1(k), x_2(k), \text{.....} X_n(k), \text{.....} x_N(k)] \in R^n$ avec k= 1,2,..., K, une des formes présentées à l'entrée, N est la dimension du vecteur d'entrée,

$D_k = (d_1(k), d_2(k), \text{.....} d_m(k), \text{.....} d_M(k)) \in \{0,1\}^m$ est le vecteur de sortie désirée correspondant à X_k, et M représente le nombre de classes à discriminer **[8]**.

9.3.3 Architecture

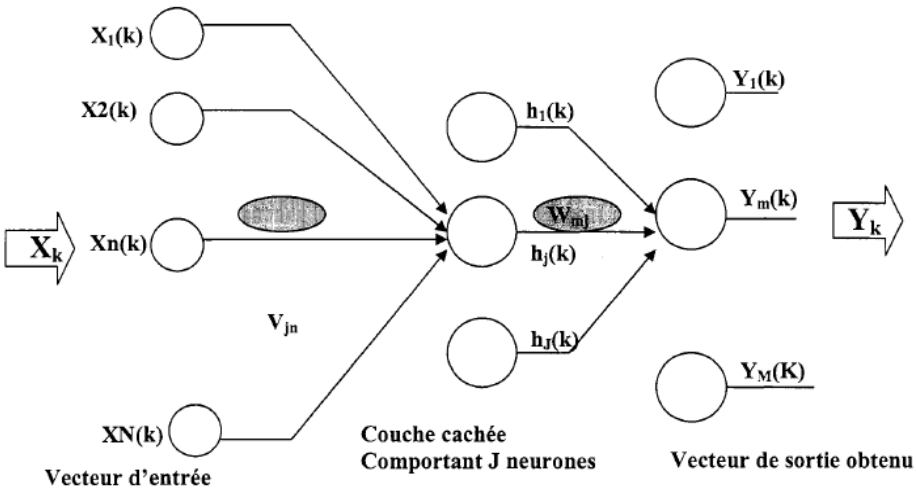

Figure 1.6 : Architecture d'un perceptron multicouche avec une couche cachée

Posons v_{Jn} le poids synaptique de pondération de la connexion entre le neurone « n » de la couche d'entrée (n=l,2,.....,N) et le neurone « j » de la couche cachée (j=l,2,....,j), où J est le nombre de neurones utilisés dans cette couche.

De même, posons w_{mj} le poids synaptique de pondération de la connexion entre le neurone «j» de la couche cachée et le neurone «m» de la couche de sortie (m=l,2,....,M). Notons à ce niveau que seul le nombre de neurones dans la couche cachée J et le paramètre de la pente de la fonction sigmoïde utilisée σ sont à déterminer avant que ne débute la phase de l'apprentissage. N (i.e. la dimension des vecteurs d'entrée) et M (i.e. le nombre de classes à discriminer) sont imposés par le problème posé.

9.3.4. Propagation directe

La première phase d'opération du perceptron multicouche consiste à propager la forme d'entrée à classifier, X_k, jusqu'à la sortie du réseau. La forme d'entrée est d'abord propagée sur la couche cachée pour produire le vecteur H_k, qui est lui-même propagé par la suite sur la couche de sortie du réseau, Y_k. La sortie du neurone h_j de la couche cachée et du neurone y_m de la couche de sortie est donnée par :

$$H_j(k) = f\left(\sum_{n=0}^{N} V_{jn} x_n(k)\right) \quad \ldots\ldots. (3)$$

$$Y_m(k) = f\left(\sum_{j=0}^{j} w_{mj} h_n(k)\right) = f\left(\sum_{j=0}^{J} w_{mj} f\left(\sum_{n=0}^{N} V_{jn} x_n(k)\right)\right) \ldots\ldots. (4)$$

Avec $v_{J0} = \beta_j$ la valeur de polarisation du neurone caché $_j$

$W_{m0} = \beta_m$ la valeur de polarisation du neurone de sortie y_m

Xo(k) = ho(k) = +1 : l'extension du vecteur avec une composante constante unitaire pour simuler la polarisation du neurone.

Chaque neurone de la couche cachée et de la couche de sortie est doté d'une connexion supplémentaire reliée à une source constante unitaire qui simule une valeur de polarisation distincte pour chacun des neurones. Cette valeur de polarisation offre un degré supplémentaire de liberté au neurone en permettant de déplacer selon l'axe horizontal la fonction de sortie du neurone (fonction sigmoïde en général, parfois la fonction linéaire pour la couche de sortie). Les valeurs de polarisation permettent globalement de déplacer par translation les courbes de séparation de classes dans l'hyper-espace de la sortie. Cette connexion supplémentaire est soumise à l'entraînement du réseau, au même titre que toutes les autres connexions synaptiques.

9.3.5 Entraînement - modification des poids synaptiques

La fonction de coût que l'on cherche à minimiser est celle de l'erreur quadratique instantanée définie par :

$$E(k) = \frac{1}{2}\sum_{m=1}^{M} e_m^2(k) = \frac{1}{2}\sum_{m=1}^{M} (d_m(k) - y_m(k))^2 \quad \ldots\ldots\ldots\ldots\ldots \quad (5)$$

Avec $e_m(k) = d_m(k)-y_m(k)$ l'erreur instantanée à la sortie du neurone *m* pour l'entrée X_k présentée à l'entrée du réseau.

L'algorithme de minimisation utilisé est celui de la descente du gradient stochastique (BLAYO & VERLEYSEN, 1996). Le gradient d'une fonction en un point est défini comme le vecteur qui pointe vers le maximum local de cette fonction le long de la pente la plus abrupte. Une technique de minimisation de l'erreur quadratique instantanée selon le négatif du gradient assure donc convergence relativement rapide vers une erreur minimum (à tout le moins localement). L'algorithme de descente de gradient stochastique consiste donc à exprimer le gradient en fonction des poids de connexion du réseau et à trouver l'amplitude et le sens des changements de poids qui minimisent le gradient de la fonction d'erreur instantanée pour la forme *Xk* présentée à l'entrée du réseau **[5]**.

9.3.6 Poids synaptiques de la couche de sortie : wmj

Au départ, la seule source d'erreur quantifiable est l'erreur de sortie. Le réseau sera donc d'abord calculé pour chacun des neurones de la couche de sortie et exprimé en fonction du poids des connexions qui parviennent à chaque neurone. La modification de poids sera apportée dans la direction du négatif du gradient. Le vecteur de l'erreur exprime en fonction du poids des connexions parvenant à la couche de sortie est un vecteur de *M x J* composantes qui pointe vers le maximum local de l'erreur quadratique instantanée. Cette rétropropagation est nécessaire afin de calculer la contribution des neurones de la couche cachée à l'erreur total que l'on peut mesurer à la sortie du réseau **[5]**.

9.3.7 Algorithme

L'algorithme complet de rétropropagation du signal d'erreur avec correction du poids des connexions selon le négatif du gradient de l'erreur est présenté ci-dessous. L'algorithme est basé sur la méthode d'apprentissage par l'exemple dans laquelle le poids des connexions est ajusté à chaque présentation d'une forme *XK* provenant de la base d'apprentissage **[56]**.

Algorithme de rétropropagation

1 - Initialisation poids Wij^k par des petites valeurs aléatoires ; Wij^k =Random

2- Présentation de la sortie désirée.

3- Présentation d'un exemple à l'entrée et calcul de la sortie de chaque couche et l'erreur correspondante.

4- Calcule des dérivées partielles par rapport à chaque poids, et adaptation des poids.

5- Retour à 3 et arrêt du processus si les sorties sont suffisamment proches des sorties désirées.

10. Les étapes de la conception d'un réseau

Le novice est souvent surpris d'apprendre que pour construire un réseau de neurones, la première chose à faire n'est pas de choisir le type de réseau mais de bien choisir ses échantillons de données d'apprentissage, de tests et validation. Ce n'est qu'ensuite que le choix du type de réseau interviendra. Afin de clarifier un peu les idées, voici chronologiquement les quatre grandes étapes qui doivent guider la création d'un réseau de neurones **[14]**.

10.1 Choix et préparation des échantillons

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix et la préparation des échantillons de données. Comme dans les cas d'analyse de données, cette étape est cruciale et va aider le concepteur à déterminer le type de réseau le plus approprié pour résoudre sont problème. La façon dont se présente l'échantillon conditionne: le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation.

10.2 Elaboration de la structure du réseau

La structure du réseau dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : un perceptron standard, un réseau de Hopfield, un réseau à décalage temporel (TDNN), un réseau de Kohonen, un ARTMAP etc... Dans le cas du perceptron par exemple, il faudra aussi choisir le nombre de neurones dans la couche cachée. Plusieurs méthodes existent et nous pouvons par exemple prendre une moyenne du nombre de neurones d'entrée et de sortie, mais rien de vaut de tester toutes les possibilités et de choisir celle qui offre les meilleurs résultats.

10.3 Apprentissage

L'apprentissage consiste tout d'abord à calculer les pondérations optimales des différentes liaisons, en utilisant un échantillon. La méthode la plus utilisée est la rétropropagation : nous entrons des valeurs des cellules d'entrée et en fonction de l'erreur obtenue en sortie (le delta), nous corrigeons les poids accordés aux pondérations. C'est un cycle qui est répété jusqu'à ce que la courbe d'erreurs du réseau ne soit croissante (il faut bien prendre garde de ne pas surentraîner le réseau qui deviendra alors moins performant). Il existe d'autres méthodes d'apprentissage telles que le quickprop **[57]** par exemple.

10.4 Validation et Tests

Alors que les tests concernent la vérification des performances d'un réseau de neurones hors échantillon et sa capacité de généralisation, la validation est parfois utilisée lors de l'apprentissage. Une fois le réseau calculé, il faut toujours procéder à des tests afin de vérifier que notre réseau réagit correctement. Il y a plusieurs méthodes pour effectuer une validation : la cross validation, le bootstrapping... mais pour les tests, dans le cas général, une partie de l'échantillon est simplement écarté de l'échantillon d'apprentissage et conservé pour les tests hors échantillon. Nous pouvons par exemple utiliser 60% de l'échantillon pour l'apprentissage, 20% pour la validation et 20% pour les tests. Dans les cas de petits échantillons, nous ne pouvons pas toujours utiliser une telle distinction, simplement parce qu'il n'est pas toujours possible d'avoir suffisamment de données dans chacun des groupes ainsi crée. Nous avons alors parfois recours à des procédures de ré-échantillonnage comme la validation croisée pour établir la structure optimale du réseau.

11. Limites des réseaux de neurones

Un des principaux reproches fait aux réseaux de neurones est l'impossibilité d'expliquer les résultats qu'ils fournissent. Les réseaux se présentent comme des boıtes noires dont les règles de fonctionnement sont inconnues. Ils créent eux-mêmes leur représentation lors de l'apprentissage. La qualité de leurs performances ne peut être mesurée que par des méthodes statistiques, ce qui amène parfois une certaine méfiance de la part des utilisateurs potentiels surtout que les résultats dans les réseaux de neurones sont beaucoup altérées par la qualité des données (données manquantes ou erronées). Le second problème qui concerne la mise en oeuvre physique ou les réseaux de neurones seront optimums quand ils auront leur propre support. Différentes solutions ont été envisagées

pour atténuer ce problème; notamment diminuer le nombre de neurones (par complexification de leur structure, par prétraitements). Le problème devient plus sérieux pour les couches cachées. Il est impossible de trouver à priori le nombre parfait de neurones. Il existe différentes techniques de décision. La plus simple et la plus grossière est de procéder par essais et erreurs. Un mauvais choix peut avoir des répercussions graves ; S'il y a trop peu de neurones, le réseau sera incapable de représenter correctement le problème. S'il y en a trop, des bruits parasites pourraient perturber les résultats, en particulier lors d'approximation de fonctions **[36]**.

12. Conclusion

Le grand avantage des réseaux de neurones réside dans leur capacité d'apprentissage automatique, ce qui permet de résoudre des problèmes sans nécessiter l'écriture de règles complexes, tout en étant tolérant aux erreurs. Cependant, ce sont de véritables boites noires qui ne permettent pas d'interpréter les modèles construits. En cas, d'erreurs du système, il est quasiment impossible d'en déterminer la cause **[4]**.

CHAPITRE 2 :

RESEAUX BAYESIENS

CHAPITRE 2
RESEAUX BAYESIENS

1. Introduction et Définition

Les réseaux bayésiens sont le résultat d'une convergence entre les méthodes statistiques tout d'abord, parce qu'elles sont précisément conçues pour permettre le passage de l'observation à la loi de probabilité, et les technologies de l'intelligence artificielle ensuite, parce que leur vocation est de permettre aux ordinateurs de traiter de la connaissance plutôt que l'information. Les réseaux bayésiens constituent aujourd'hui l'un des formalismes les plus complets et les plus cohérents pour l'acquisition, la représentation, la classification et l'utilisation de connaissances par des ordinateurs. Encore du domaine de la recherche au début des années 1990, cette technologie connaît de plus en plus d'applications, depuis le contrôle de véhicules autonomes à la modélisation des risques opérationnels, en passant par le Data Mining ou la localisation des gènes.

Les réseaux bayésiens, qui doivent leur nom aux travaux de THOMAS BAYES au 18ème siècle sur la théorie des probabilités, sont le résultat de recherches effectuées dans les années 1980, dues à J.PEARL à UCLA et à une équipe de recherche danoise à l'université de Aalborg. L'objectif initial de ces travaux était d'intégrer la notion d'incertitude dans les systèmes experts. Les chercheurs se sont rapidement aperçus que la construction d'un système expert nécessitait presque toujours la prise en compte de l'incertitude dans le raisonnement **[7]**. En effet, dans la plupart des domaines complexes, un expert humain est capable de porter un jugement sur une situation, même en l'absence de toutes les données nécessaires. En médecine, par exemple, une même combinaison de symptômes peut être observée dans différentes pathologies. Il n'y a donc pas de règle stricte qui permette de passer systématiquement d'un ensemble d'observations à un diagnostic. De plus, les informations pertinentes ne sont pas toujours observables. Pour que des systèmes experts puissent être utilisés dans de tels domaines, il faut donc qu'ils soient capables de raisonner sur des faits et des règles incertains. Dans le cadre des systèmes experts, les réseaux bayésiens constituent une approche possible pour intégrer l'incertitude dans le

raisonnement. D'autres méthodes existent, mais les réseaux bayésiens présentent l'avantage d'être une approche quantitative.

D'un autre côté, imaginons à présent un statisticien, qui s'efforce d'analyser un tableau de mesures de plusieurs variables sur une population donnée. Il va pour cela essayer de démêler les relations pertinentes entre les variables, les dépendances ou indépendances entre plusieurs groupes de variables. L'utilisation de réseaux bayésiens va lui permettre d'extraire de ce tableau une représentation compacte, sans perte d'information, à partir de laquelle il va être beaucoup plus facile de raisonner.

Le lien entre ces deux problématiques est clairement celui de la connaissance. D'un côté, un expert dispose d'une connaissance présentant certaines incertitudes. Pour la formaliser, il va utiliser des descriptions causales : A a une influence sur B ; en général, si B est observé, il y a de fortes chances que C se produise, etc. Pour rendre cette connaissance opérationnelle, il lui faut quantifier ses incertitudes, c'est-à-dire les convictions plus ou moins précises que l'expert a des liens entre les faits. D'un autre côté, un ensemble de données contient lui aussi de la connaissance, mais qui n'est pas directement accessible à un analyste, car elle est noyée dans les chiffres. Pour rendre cette connaissance interprétable, il faut la transformer en modèle de causalité, mettant en évidence les liens entre les variables observées. C'est grâce à la notion mathématique de probabilité que les réseaux bayésiens vont permettre de résoudre ces deux problèmes duaux : transformer en chiffres une connaissance subjective, et transformer en modèle interprétable une connaissance contenue dans des chiffres.

L'expert formalise sa connaissance sous forme de modèle de causalité, indiquant les liens entre les variables. Cette description graphique est transformée en une loi de probabilité équivalente**[7]**.Cette loi de probabilité permet de faire des calculs, et donc en particulier des raisonnements prenant en compte des aspects incertains. Réciproquement, à partir des données, on va mettre en évidence des propriétés (indépendances, causalités) de la relation entre les différentes variables observées. Cette relation est transformée en graphe de causalités, qui peut alors être lu et interprété par un analyste, beaucoup plus facilement que les données initiales. Ces deux opérations ne sont possibles que grâce aux trois propriétés suivantes **[7]** :

• Les probabilités subjectives (celles que l'expert utilise pour décrire les liens entre les variables) sont assimilables à des probabilités mathématiques.

• Les fréquences observées (tableau de mesures) sont assimilables à des probabilités mathématiques.

• Le graphe de causalités est une représentation fidèle d'une loi de probabilité sous-jacente: il est alors possible de raisonner sur le graphe sans revenir aux chiffres.

Les deux premières propriétés sont des hypothèses de travail, et leur discussion peut être considérée comme relevant de la philosophie. La dernière, en revanche, est un résultat très important, qui garantit que tout ce qui peut être déduit du graphe est également vrai dans la distribution de probabilité sous-jacente.

1.1 Graphe causal

Ce graphe est orienté et acyclique. Ses nœuds sont des variables d'intérêt du domaine et les arcs des relations de dépendance entre ces variables. L'ensemble des nœuds et des arcs forme ce que nous l'appelons la structure du réseau bayésien. C'est la représentation qualitative de la connaissance **[6]**.

1.2 Distributions locales de probabilité

L'ensemble des distributions de probabilité sont les paramètres du réseau. Pour chaque nœud nous disposons d'une table de probabilité P(variable/parents(variable)) qui représente la distribution locale de probabilité. Il faut remarquer que chaque nœud ne dépend que de l'état de ses parents. Il s'agit de la représentation quantitative de la connaissance **[6]**.

2. Fonctionnement

Tout comme pour les réseaux de neurones, il convient de construire un réseau bayésien spécifiquement dédié au problème que nous souhaitons traiter **[6]**.

Nous pouvons distinguer deux phases :

- la phase de construction du réseau qui peut solliciter experts et techniques d'apprentissage ;
- la phase d'utilisation qui fait appel à une capacité très intéressante des réseaux bayésiens, l'inférence.

2.1 Phase de construction

Les réseaux bayésiens permettent de combiner la connaissance d'experts avec la

connaissance extraite à partir de données. Les experts peuvent par exemple déterminer les dépendances entre les variables alors qu'un apprentissage automatique permettra de déterminer la distribution de probabilité associée à chaque variable. Il ne s'agit ici que d'un exemple car il est tout à fait envisageable que des experts définissent entièrement un réseau bayésien, graphe et distribution de probabilité. Au contraire, ces deux éléments peuvent être construits automatiquement par apprentissage du système.

2.2 Phase d'utilisation

L'utilisation des réseaux bayésiens repose sur la propagation de l'information au sein du réseau, c'est à dire des calculs de probabilités, c'est ce que nous l'appelons l'inférence. Après avoir fait une observation sur une variable, comment cette information va t-elle se répercuter sur l'état des autres variables ?

3. Applications

La première application des réseaux bayésiens est le diagnostic. Connaissant la panne, un système basé sur des réseaux bayésiens pourra déterminer les causes les plus probables ayant entraînées le problème. Toutefois, les réseaux bayésiens sont aussi utilisés pour faire de la classification. Ils vont alors se baser sur un certain nombre de caractéristiques pour pouvoir bien classer les données dans une catégorie.

4. Utilité des réseaux bayésiens

Selon le type d'application, l'utilisation pratique d'un réseau bayésien peut être envisagée au même titre que celle d'autres modèles : réseaux neuronaux, systèmes experts, arbres de décision, modèles d'analyse de données (régressions linéaires), modèles logiques, etc. Naturellement, le choix de la méthode fait intervenir différents critères, comme la facilité, le coût et le délai de mise en œuvre d'une solution. En dehors de toute considération théorique, les aspects suivants des réseaux bayésiens les rendent, dans de nombreux cas, préférables à d'autres modèles **[7]**:

4.1 Acquisition des connaissances

La possibilité de rassembler et de fusionner des connaissances de diverses natures dans un même modèle : retour d'expérience (données historiques ou empiriques), expertise (exprimée sous forme de règles logiques, d'équations, de statistiques ou de probabilités

subjectives), observations. Dans le monde industriel par exemple, chacune de ces sources d'information, quoique présente, est souvent insuffisante individuellement pour fournir une représentation précise et réaliste du système analysé.

4.1.1 Un recueil d'expertise facilité

La représentation des connaissances utilisées dans les réseaux bayésiens est la plus intuitive possible : simplement relier des causes et des effets par des flèches. Pratiquement toute représentation graphique d'un domaine de connaissances peut être présentée sous cette forme. De nombreuses expériences montrent qu'il est souvent plus facile pour un expert de formaliser ses connaissances sous forme de graphe causal que sous forme de système à base de règles, en particulier parce que la formulation de règles sous la forme SI... ALORS est très contraignante, et peut être facilement mise en défaut.

Certains auteurs considèrent qu'il existe une différence de nature entre les deux processus d'acquisition de connaissances. Lorsqu'on essaie de mettre au point un système expert, par exemple pour une application de diagnostic, l'expert doit décrire le processus de raisonnement qui le conduit de ses observations à une conclusion. En revanche, un modèle fondé sur un graphe causal décrit la perception de l'expert du fonctionnement du système. Effectuer un diagnostic n'est alors qu'une résultante de cette modélisation.

4.1.2 Un ensemble complet de méthodes d'apprentissage

Les algorithmes actuels permettent d'envisager l'apprentissage de façon très complète :

• En l'absence totale de connaissances, on peut rechercher à la fois la structure du réseau la plus adaptée, c'est-à-dire les relations de dépendance et d'indépendance entre les différentes variables, et les paramètres, ou probabilités, c'est-à-dire la quantification de ces relations.

• Si nous disposons de connaissances a priori sur la structure des causalités, et d'une base d'exemples représentative, la détermination des matrices de probabilités conditionnelles, qui sont les paramètres du réseau, peut être effectuée par simple calcul de fréquences, par détermination du maximum de vraisemblance, ou par des méthodes bayésiennes.

Ces méthodes peuvent être étendues dans le cadre de bases de données incomplètes. Dans l'optique de rechercher un compromis entre apprentissage et généralisation, il est également possible d'effectuer des apprentissages en contraignant la structure du réseau.

4.1.3 Un apprentissage incrémental

Le principe général de l'apprentissage dans les réseaux bayésiens est décrit par la formule générale : Posteriori = Vraisemblance.APriori ……...(6)

Cette formule conditionne la modification de la connaissance contenue dans le réseau par l'acquisition de nouveaux exemples. Elle s'interprète en disant que la connaissance contenue a priori, ou à un instant quelconque, dans le réseau, est transformée a posteriori en fonction de la vraisemblance de l'observation des exemples étudiés selon la connaissance initiale. Autrement dit, plus les exemples observés s'écartent de la connaissance contenue dans le réseau, plus il faut modifier celle-ci. Théoriquement, cette formule, qui n'est autre que la formule de Bayes appliquée à la connaissance, est valable aussi bien pour l'apprentissage de paramètres que pour l'apprentissage de structure. Aucune des techniques concurrentes, ni les réseaux neuronaux, ni les arbres de décision, ne permet de prendre en compte ce problème de la mise à jour des modèles de connaissance de façon aussi naturelle, même si aujourd'hui sa mise en œuvre dans les réseaux bayésiens n'est possible techniquement que dans certains cas particuliers. Nous pensons que la capacité d'apprentissage incrémental est essentielle, car elle autorise l'évolution des modèles. Toute démarche de modélisation qui ne concerne pas les sciences de la nature doit intégrer les évolutions de l'environnement modélisé, et donc faire dépendre le modèle du temps. L'apprentissage incrémental est une réponse possible à ce problème.

4.2 Représentation des connaissances

La représentation graphique d'un réseau bayésien est explicite, intuitive et compréhensible par un non spécialiste, ce qui facilite à la fois la validation du modèle, ses évolutions éventuelles et surtout son utilisation. Typiquement, un décideur est beaucoup plus enclin à s'appuyer sur un modèle dont il comprend le fonctionnement qu'à faire confiance à une boîte noire.

4.2.1 Un formalisme unificateur

La plupart des applications qui relèvent des réseaux bayésiens sont des applications d'aide à la décision. Par nature, ces applications intègrent un certain degré d'incertitude, qui est très bien pris en compte par le formalisme probabiliste des réseaux bayésiens. Par exemple, dans les applications de data mining, nous utilisons une base de données pour mettre au point un modèle prédictif. Par définition, une prévision comporte une part

d'incertitude. Or la décision, elle, doit souvent être binaire : dans une application de scoring, on doit par exemple accorder ou refuser le crédit. La façon la plus naturelle d'interpréter un score est donc une probabilité (dans l'exemple du scoring, une probabilité de défaillance). Les techniques disponibles pour traiter ce genre de problème (modèles de régression, réseaux de neurones, arbres de décision) ne sont pas construites sur un formalisme de probabilités. C'est a posteriori que nous attribuons en général une interprétation en termes de probabilités de la prévision d'un réseau neuronal ou d'un arbre de décision. Les réseaux bayésiens ne sont qu'une représentation d'une distribution de probabilité. C'est une telle distribution que nous la représentons à partir de connaissances explicites ou que nous approchons à partir d'une base de données, et c'est à partir de la distribution approchée que nous effectuons des inférences. Toute prévision issue d'un réseau bayésien est donc par construction une probabilité. De plus, les réseaux bayésiens permettent de considérer dans un même formalisme la représentation de modèles de causalités et les statistiques multivariées. Il en est de même des techniques les plus utilisées pour le data mining comme les arbres de décision ou les réseaux de neurones, qui peuvent également être représentés au sein de ce formalisme.

4.2.2 Une représentation des connaissances lisible

Les deux propriétés fondamentales des réseaux bayésiens sont, d'abord, d'être des graphes orientés, c'est-à-dire de représenter des causalités et non des simples corrélations, et, ensuite, de garantir une correspondance entre la distribution de probabilité sous-jacente et le graphe associé. Considérons le cas d'une application de datamining, où nous cherchons à comprendre les interrelations entre des variables contenues dans une base de données de clients, par exemple. Si nous se trouvons dans le cas où le réseau est entièrement mis au point à partir des données (cas de l'apprentissage de la structure et des paramètres), cela signifie que nous allons disposer d'une visualisation graphique de ces interrelations. Avant même d'utiliser ce réseau pour effectuer des inférences, nous avons disposer d'une visualisation de la connaissance, directement lisible et interprétable par des experts du domaine.

4.3 Utilisation des connaissances

Un réseau bayésien est polyvalent : on peut se servir du même modèle pour évaluer, prévoir, diagnostiquer, ou optimiser des décisions, ainsi que pour classifier des données, ce qui contribue à rentabiliser l'effort de construction du réseau bayésien.

4.3.1 Une gamme de requêtes très complète

L'utilisation première d'un réseau bayésien est le calcul de la probabilité d'une hypothèse connaissant certaines observations. Cependant, les possibilités offertes par les algorithmes d'inférence permettent d'envisager une gamme de requêtes très complète, et qui peut être extrêmement intéressante dans certains types d'applications. Tout d'abord, il n'y a aucune réelle contrainte sur les informations nécessaires pour être en mesure de calculer la probabilité d'un fait : nous pouvons connaître exactement la valeur d'une variable, savoir qu'elle est égale à l'une ou l'autre de deux valeurs, ou encore savoir avec certitude qu'une de ses valeurs possibles est exclue. Dans tous les cas, l'inférence est possible, et la nouvelle information permet de raffiner les conclusions.

Il n'y a pas d'entrées ni de sorties dans un réseau bayésien (ou de variables indépendantes et dépendantes). Le réseau peut donc être utilisé pour déterminer la valeur la plus probable d'un nœud en fonction d'informations données (prévoir ou sens entrées sorties), mais également pour connaître la cause la plus probable d'une information donnée (expliquer ou sens sorties entrées). En termes d'inférences, cette dernière requête s'appelle explication la plus probable et revient à rechercher l'état des autres variables pour lequel ce qui a été observé était le plus probable. Parmi les autres requêtes importantes, l'analyse de sensibilité à une information mesure comment la probabilité d'une hypothèse s'accroît quand nous avons fait une observation. Certaines observations peuvent être considérées comme inutiles, suffisantes ou cruciales par rapport à une hypothèse donnée.

Le mécanisme de propagation peut être également utilisé pour déterminer l'action la plus appropriée à effectuer, ou l'information la plus pertinente à rechercher. Considérons par exemple un problème de diagnostic, dans lequel manquent plusieurs des données qui permettraient de conclure. Ce mécanisme dans un réseau bayésien permet de connaître la donnée dont la connaissance apporterait le maximum d'informations. Dans le cas où la recherche de chaque donnée a un coût, il est possible de rechercher la solution optimale en tenant compte de ce coût. De plus, il est possible de chercher également une séquence optimale d'actions ou de requêtes.

4.3.2 Optimisation d'une fonction d'utilité

Imaginons un problème de classification, par exemple, un problème de détection de fraudes, sur des cartes bancaires, ou dans l'utilisation de services de télécommunications.

Rechercher le système qui donne avec la meilleure fiabilité possible, la probabilité de fraude n'est peut-être pas l'objectif réel de ce type d'application. En effet, ce que nous cherchons ici à optimiser est une utilité économique. Sachant que les fausses alarmes aussi bien que les fraudes manquées ont un coût, l'objectif est bien de minimiser le coût global. Une version spécifique des réseaux bayésiens, appelée diagramme d'influences, permet de les adapter à ce type de problème. Dans les diagrammes d'influence, on ajoute aux nœuds qui représentent des variables, deux autres types de nœuds :

• Les nœuds de décision, figurés par des carrés,

• Un nœud d'utilité, figuré par un losange.

La figure 2.1 ci-après représente un diagramme d'influence pour un problème de détection de fraude sur une carte bancaire. Les variables représentées sont les suivantes:

• La variable F est binaire et représente le fait qu'il y a ou non fraude.

• La variable B représente le résultat d'une vérification effectuée sur une base de données. Cette variable a trois modalités : le contrôle est négatif, positif, ou non effectué.

• La variable P a également trois modalités, et représente le résultat d'un contrôle d'identité du porteur.

• Le nœud de décision D représente la décision d'effectuer les contrôles complémentaires B et P. Ce nœud a donc également trois modalités : n'effectuer aucun test, effectuer le test B, ou effectuer les deux tests B et P.

• Le nœud de décision A représente la décision d'autoriser la transaction, et est donc binaire.

• Le nœud d'utilité V est une fonction de l'ensemble des variables précédentes, représentant le coût de la situation.

En outre, on suppose connus le montant de la transaction et le coût de chaque contrôle, et les tables de probabilités conditionnelles reliant les variables entre elles. L'objectif est de prendre les bonnes décisions D et A; autrement dit, de prendre les décisions qui minimisent l'espérance mathématique de V.

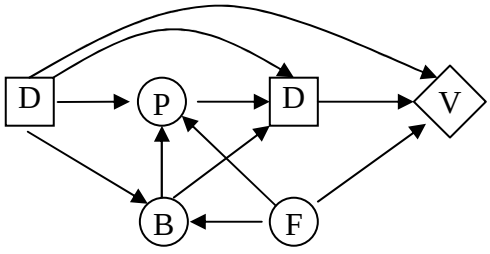

Figure 2.1 : Un diagramme d'influence pour la fraude sur carte bancaire

4.4 Limites des réseaux bayésiens

Il existe aujourd'hui de nombreux logiciels pour saisir et traiter des réseaux bayésiens. Ces outils présentent des fonctionnalités plus ou moins évoluées : apprentissage des probabilités, apprentissage de la structure du réseau, possibilité d'intégrer des variables continues, etc. Cependant les réseaux bayésiens recensent quelques limites à leur utilisation parmi les plus importants **[7]** :

4.4.1 Un recul encore insuffisant pour l'apprentissage

Dans la mesure où elle s'est surtout développée dans le cadre des systèmes experts, la technique des réseaux bayésiens n'a pas immédiatement intégré l'ensemble de la problématique de l'apprentissage, comme cela avait été le cas des réseaux neuronaux. Aujourd'hui, l'essentiel de la littérature sur l'apprentissage avec des réseaux bayésiens ignore le problème de la capacité de généralisation d'un modèle, et des précautions que cela implique au moment de la construction du modèle. La prise en compte de ce problème peut s'effectuer par le choix du critère de recherche ou de distance des distributions de probabilité. En effet, l'apprentissage de réseaux bayésiens revient à rechercher parmi un ensemble de distributions, la distribution la plus proche possible, en un certain sens, de la distribution représentée par les données. En limitant l'ensemble de recherche, on peut éviter le problème de surapprentissage, qui revient dans ce cas à calquer exactement la distribution représentée par les exemples.

4.4.2 Utilisation des probabilités

L'utilisation des graphes de causalités est une approche très intuitive. L'utilisation des probabilités pour rendre ces modèles quantitatifs était justifiée. Il reste cependant que la notion de probabilité est, au contraire, assez peu intuitive. Il est en effet assez facile de construire des paradoxes fondés sur des raisonnements probabilistes. Les modèles déterministes, formulés en termes d'entrées et de sorties, comme les modèles de régression, les réseaux de neurones, ou les arbres de décision, même s'ils peuvent être réinterprétés dans le cadre d'un formalisme probabiliste, restent d'un abord plus facile.

4.4.3 Lisibilité des graphes

En effet, même si la connaissance manipulée dans les réseaux bayésiens, ou extraites des données par les algorithmes d'apprentissage associé est lisible puisque représentée sous forme de graphes, elle reste moins lisible que celle représentée par un arbre de

décision, par exemple, surtout si ce graphe présente un grand nombre de nœuds. Notons aussi que l'information représentée par le graphe est la structure des causalités. Les probabilités ne sont pas représentables, et on n'a donc pas idée, à la simple lecture du graphe, de quel arc est important.

4.4.4 Les variables continues

L'essentiel des algorithmes développés pour l'inférence et l'apprentissage dans les réseaux bayésiens, aussi bien que les outils disponibles sur le marché pour mettre en œuvre ces algorithmes utilisent des variables discrètes. En effet, la machinerie des algorithmes d'inférence est essentiellement fondée sur une algèbre de tables de probabilités. De même, les algorithmes d'apprentissage modélisent en général les distributions de probabilité des paramètres contenus dans les tables du réseau, c'est-à-dire de probabilités discrètes. Même s'il est théoriquement possible de généraliser les techniques développées aux variables continues, il semble que la communauté de recherche travaillant sur les réseaux bayésiens n'a pas encore vraiment intégré ces problèmes. Cela pénalise cette technologie, en particulier pour des applications de data mining où variables continues et discrètes cohabitent.

4.4.5 La complexité des algorithmes

La généralité du formalisme des réseaux bayésiens aussi bien en termes de représentation que d'utilisation les rend difficiles à manipuler à partir d'une certaine taille. La complexité des réseaux bayésiens ne se traduit pas seulement en termes de compréhension par les utilisateurs. Les problèmes sous-jacents sont pratiquement tous de complexité non polynomiale, et conduisent à développer des algorithmes approchés, dont le comportement n'est pas garanti pour des problèmes de grande taille.

4.5 Comparaison avec d'autres techniques

Du point de vue des applications, les avantages et inconvénients des réseaux bayésiens par rapport à quelques-unes des techniques concurrentes peuvent se résumer sur le tableau ci-dessous. Nous avons regroupé avantages et inconvénients selon les trois rubriques utilisées précédemment, l'acquisition, la représentation et l'utilisation des connaissances **[7]**. La représentation adoptée est la suivante :

• A chaque ligne correspond une caractéristique, qui peut être un avantage, ou la prise en compte d'un problème spécifique.

• Si la technique considérée permet de prendre en compte ce problème, ou présente cet avantage, un signe + est placé dans la case correspondante.

• Un signe * est placé dans la case de la meilleure technique du point de vue de la caractéristique considérée.

Tableau 2.1 Avantages comparatifs des différents algorithmes en classification

Connaissances	Analyse de données	Arbre de décision	Systèmes experts	Réseaux neuronaux	Réseaux bayésiens
ACQUISITION					
Expertise seulement			*		
Données seulement	+	+		*	+
Mixte	+	+		+	*
Incrémental				+	*
Généralisation	+	+		*	+
Données Incomplètes				+	*
REPRESENTATION					
Incertitude			+		*
Lisibilité	+	+	+		*
facilité		*		+	
Homogénéité					*
UTILISATION					
Requêtes élaborées	+		+		*
Utilité économique	+			+	*
Performances	+			*	

5. Conception d'un réseau bayésien

Malgré la diversité des applications, la construction d'un réseau bayésien se réalise, schématiquement, en trois étapes essentielles **[7]** :

1. Identification des variables et de leur espace d'états
2. Définition de la structure du réseau bayésien
3. Loi de probabilité conjointe des variables

Chacune des trois étapes peut impliquer un recueil d'expertise.

5.1 Identification des variables et de leur espace d'états

La première étape de construction du réseau bayésien est la seule pour laquelle l'intervention humaine est absolument indispensable. Il s'agit de déterminer l'ensemble des variables X_i, catégorielles ou numériques, qui caractérisent le système. Comme dans tout travail de modélisation, un compromis entre la précision de la représentation et la faisabilité de la construction du modèle doit être trouvé, au moyen d'une discussion entre les experts et le modélisateur. Lorsque les variables sont identifiées, il est ensuite nécessaire de préciser l'espace d'états de chaque variable X_i, c'est-à-dire l'ensemble de ses valeurs possibles.

La majorité des logiciels de réseaux bayésiens ne traite que des modèles à variables discrètes, ayant un nombre fini de modalités. Si tel est le cas, il est impératif de discrétiser les plages de variation des variables continues. Cette limitation est parfois gênante en pratique, car des discrétisations trop fines peuvent conduire à des tables de probabilités de grande taille, de nature à saturer la mémoire de l'ordinateur.

5.2 Définition de la structure du réseau bayésien

La deuxième étape consiste à identifier les liens entre variables, c'est à dire à répondre à la question : pour quels couples (i, j) la variable X_i influence-t-elle la variable X_j?. Dans la grande majorité des applications, cette étape s'effectue par l'interrogation d'experts. Dans ce cas, des itérations sont souvent nécessaires pour aboutir à une description consensuelle des interactions entre les variables X_i. L'expérience montre cependant que la représentation graphique du réseau bayésien est dans cette étape un support de dialogue extrêmement précieux.

Un réseau bayésien ne doit pas comporter de circuit orienté ou boucle (figure 2.2). Cependant, le nombre et la complexité des dépendances identifiées par les experts laissent parfois supposer que la modélisation par un graphe acyclique est impossible. Il est alors important de garder à l'esprit que, quelles que soient les dépendances stochastiques entre des variables aléatoires discrètes, il existe toujours une représentation par réseau bayésien de leur loi conjointe. Ce résultat théorique est fondamental et montre bien la puissance de modélisation des réseaux bayésiens.

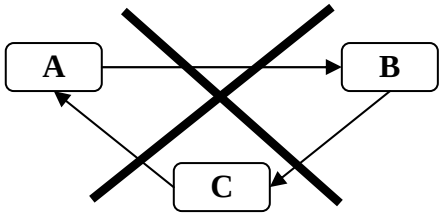

Figure 2.2 : Boucle dans un réseau bayésien

Lorsque nous disposons d'une quantité suffisante de données de retour d'expérience concernant les variables X_i, la structure du réseau bayésien peut également être apprise automatiquement par le réseau bayésien, à condition bien sûr que le logiciel utilisé soit doté de la fonctionnalité adéquate.

5.3 Loi de probabilité conjointe des variables

La dernière étape de construction du réseau bayésien consiste à renseigner les tables de probabilités associées aux différentes variables. Dans un premier temps, la connaissance des experts concernant les lois de probabilité des variables est intégrée au modèle. Concrètement, deux cas se présentent selon la position d'une variable X_i dans le réseau bayésien :

- La variable X_i n'a pas de variable parente : les experts doivent préciser la loi de probabilité marginale de X_i .
- La variable X_i possède des variables parentes : les experts doivent exprimer la dépendance de X_i en fonction des variables parentes, soit au moyen de probabilités conditionnelles, soit par une équation déterministe (que le logiciel convertira ensuite en probabilités).

Le recueil de lois de probabilités auprès d'experts est une étape délicate du processus de construction du réseau bayésien. Typiquement, les experts se montrent réticents à chiffrer la plausibilité d'un événement qu'ils n'ont jamais observé. Cependant, une discussion approfondie avec les experts, aboutissant parfois à une reformulation plus précise des variables, permet dans de nombreux cas l'obtention d'appréciations qualitatives. Ainsi, lorsqu'un événement est clairement défini, les experts sont généralement mieux à même d'exprimer si celui-ci est « probable», «peu probable», «hautement improbable», etc. Il est alors possible d'utiliser une table de conversion d'appréciations qualitatives en probabilités.

Le cas d'absence totale d'information concernant la loi de probabilité d'une variable Xi peut également être rencontré. La solution pragmatique consiste alors à affecter à Xi une loi de probabilité arbitraire, comme par exemple une loi uniforme. Lorsque la construction du réseau bayésien est achevée, l'étude de la sensibilité du modèle à cette loi permet de décider ou non de consacrer davantage de moyens à l'étude de la variable Xi. La quasi-totalité des logiciels commerciaux de réseaux bayésiens permet l'apprentissage automatique des tables de probabilités à partir de données. Par conséquent, dans un second temps, les éventuelles observations des Xi peuvent être incorporées au modèle, afin d'affiner les probabilités introduites par les experts.

Il est rare en pratique que les données soient suffisamment nombreuses et fiables

pour caractériser de manière satisfaisante la loi de probabilité conjointe des variables X_i. Cependant, si tel est le cas, l'apprentissage automatique des probabilités rend inutile la phase de renseignement du modèle par des probabilités expertes; on peut alors se contenter, dans la phase initiale, d'attribuer à chaque variable une loi de probabilité uniforme.

6. Théorème de Bayes et concepts reliés

Le raisonnement bayésien trouve son fondement théorique dans le théorème de Bayes : il permet les inférences probabilistes et il repose sur l'hypothèse que les solutions recherchées peuvent être trouvées à partir de distributions de probabilité dans les données et dans les hypothèses **[54]**.

6.1 Théorème

Le théorème de Bayes associe la probabilité a posteriori d'une hypothèse h sachant les données D, *P(h/D)*, à 3 autres probabilités **[52] [53]** :

$$P(h/D) = \frac{P(D/h) * P(h)}{P(D)} \quad \text{……. (7)}$$

où

• *P(h)* = probabilité que l'hypothèse h soit vérifiée indépendamment des données D (ce terme est également appelé probabilité a priori);

• *P(D)* = probabilité d'observer les données D indépendamment de l'hypothèse h (ce terme est également appelé évidence);

• *P(D/h)* = probabilité d'observer les données D sachant que l'hypothèse h est vérifiée (ce terme est également appelé likelihood).

6.2 Hypothèse avec probabilité a posteriori maximum

Le théorème de Bayes peut être utilisé afin de déterminer l'une des hypothèses la plus probable selon les données; i.e. une hypothèse maximisant *P(h/D)*. Cette hypothèse avec probabilité a posteriori maximum (h_{MAP}) est définie par **[54]** :

$$h_{MAP} = \text{argmax}_{\,h}\, P(h/D) \quad \text{……. (8)}$$

Une méthode de raisonnement (ou d'apprentissage) qui recherche h_{MAP} est dite méthode de probabilité a posteriori maximum.

6.3 Hypothèse avec likelihood maximum

Le théorème de Bayes peut être utilisé afin de déterminer l'une des hypothèses pour laquelle la probabilité d'observer des données est maximale; i.e. une hypothèse maximisant *P(D/h)*. Cette hypothèse avec likelihood maximum (h_{ML}) est définie par **[54]** :

$$h_{ML} = \text{argmax}_{h} P(D/h) \quad \ldots\ldots\ldots\ldots (9)$$

Une méthode de raisonnement (ou d'apprentissage) qui recherche h_{ML} est dite méthode de likelihood maximum.

6.4 Algorithme de force brute

Un algorithme de force brute recherche à travers toutes les hypothèses, soit h_{MAP} (une hypothèse maximisant la probabilité a posteriori), ou soit h_{ML} (une hypothèse maximisant le likelihood) **[54]**.

7. Classificateur bayésien naïf

Les classificateurs bayésiens utilisent des méthodes basées sur le théorème de Bayes afin de déterminer les probabilités d'associer certaines classes à certaines instances selon les données d'entraînement **[52] [53]**. Le classifieur bayesien naïf est une méthode d'apprentissage supervisé qui repose sur une hypothèse simplificatrice forte : les descripteurs sont deux à deux indépendants conditionnellement aux valeurs de la variable à prédire. Ses performances sont comparables aux autres techniques d'apprentissage **[60]**. Cette présupposition d'indépendance des attributs ne tient pas compte de la réalité dans beaucoup de domaines, d'où l'épithète naïf pour qualifier ce type de classificateur. Pour un ensemble de classes possibles C et une instance spécifiée par un ensemble d'attributs A, la valeur de classification bayésienne naïve, c, correspondant à ces attributs est définie comme suit:

$$c = \underset{c_j \varepsilon C}{\text{argmax}} P(c_j) \prod_{a_j \varepsilon A} P(a_i \mid c_j) \quad \ldots\ldots\ldots\ldots (10)$$

Un classificateur bayésien naïf se révèle robuste et efficace et, dans certains domaines, ses résultats sont compétitifs à ceux des meilleures méthodes. Cette efficacité s'applique même dans des domaines où la présupposition d'indépendance des attributs ne s'applique pas tout à fait : le domaine de la classification de document en particulier est un domaine pour lequel les classificateurs bayésiens naïfs sont souvent utilisés avec succès malgré qu'il existe une certaine dépendance entre les attributs (les mots) **[52] [53]**.

8. Conclusion

Les réseaux bayésiens sont actuellement une des techniques les plus intéressantes de l'intelligence artificielle et du Data Mining, spécialement dans le domaine de la classification car ils permettent la représentation de la connaissance par un graphe causal intuitif et compréhensible. De plus, comme ils sont basés sur des probabilités, ils intègrent l'incertitude dans le raisonnement. Malheureusement, il s'agit d'un domaine de recherche récent et l'offre logicielle est encore pauvre et incomplète **[6]**. Cependant, leur grande puissance de supporter des données bruitées et des données manquantes ont fait qu'ils sont très utiles dans la classification où les connaissances sont incertaines **[54].**

CHAPITRE 3 :

CLASSIFICATION HYBRIDE RN & RB

CHAPITRE 3
CLASSIFICATION HYBRIDE RN& RB

1. Introduction et problématique

Depuis plusieurs années les Réseaux de neurones et les Réseaux Bayésiens sont devenus des outils très populaire en classification des données dans différents systèmes. Une des raisons, est leur bonne performance pour plusieurs types d'applications à petits, moyens et grands volumes. Leurs algorithmes sont optimaux et donnent des résultats satisfaisants dans le sens ou le taux d'erreur est minime, cependant ce taux varie sensiblement lorsque le nombre de descripteurs est grand et le volume de la base de données est petit ou bien lorsque les données utilisées pour générer l'algorithme de classification sont erronées, non complètes ou manquantes, ce qui influe sur la généralisation de cet algorithme.

Par leurs caractéristiques en discrimination et en généralisation, l'utilisation des réseaux de neurones peut être avantageuse chaque fois que l'on cherche à établir une relation non linéaire entre les données, cependant l'apprentissage dans les réseaux de neurones est beaucoup influencé par la qualité des données. D'un autre côté, les Réseaux Bayésiens se révèlent très utiles pour la classification dans le cadre des données incomplètes ou erronées tout en se montrant faibles en ce qui concerne la généralisation et la discrimination de certaines données. D'un autre coté l'interprétation des résultats obtenue après une classification est généralement confuse par le nombre élevé des descripteurs. Ceci nous amène à chercher une méthode hybride permettant aux Réseaux de neurones en collaboration avec les réseaux bayésiens de palier aux défaillances citées ci-dessus. Ainsi, il semble intéressant de tenter de combiner les capacités respectives des deux techniques, pour produire de nouveaux modèles hybrides performants qui puisent leur source dans les deux formalismes. Cependant, cette combinaison n'est pas simple à réaliser car dans la classification supervisée des données, peut de modèles d'hybridation ont été étudié vu la complexité de ces algorithmes, néanmoins quelques études ont été menées

dans différents domaines tel que les modèles stochastiques, en particulier **[16]**, **[17]** et les modèles probabilistes **[18]**.

Alors que le modèle unificateur ne sera pas étudié, nous proposons dans ce travail une manière d'intégrer les réseaux neuronaux et les réseaux bayésiens dans une architecture probabiliste en tirant avantage des deux outils à savoir (figure 3.1) :

- Réseau de neurones en amont d'un réseau Bayésien,
- Réseau de neurones en aval d'un réseau Bayésien,

Tel que le premier Réseau sert comme un optimisateur de la base de donnée pour la classification et le deuxième Réseau exécute cette classification en utilisant seulement la base de données réduite par le premier réseau.

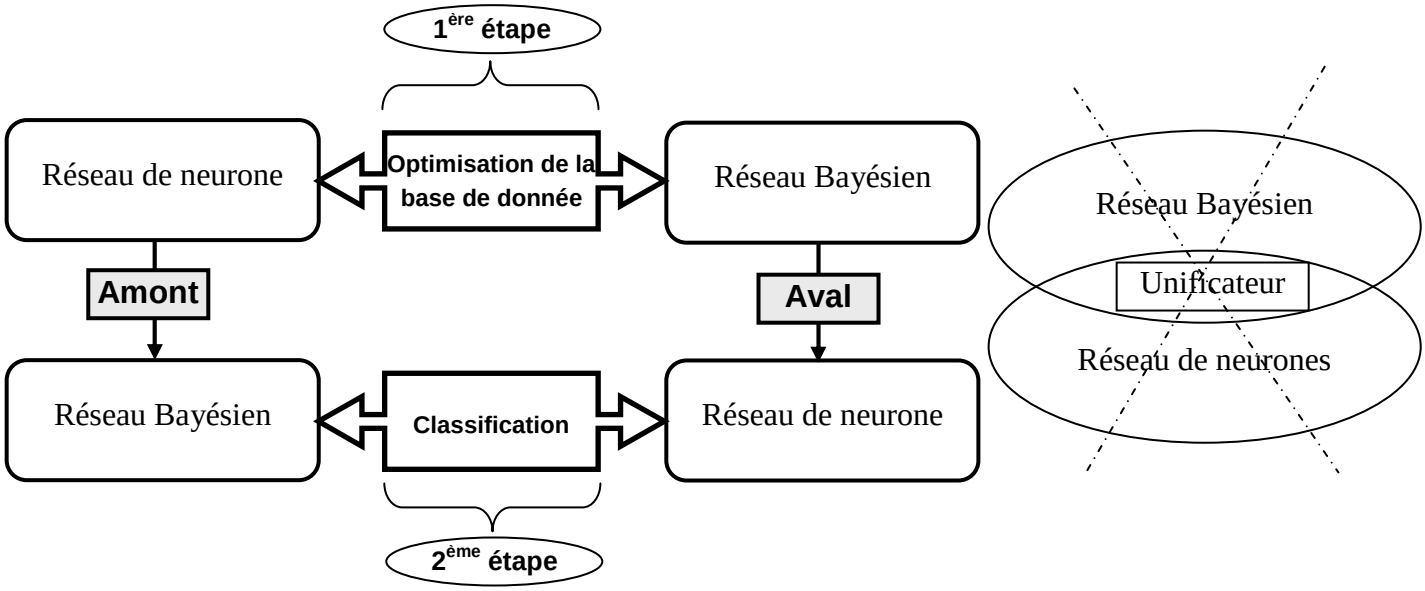

Figure 3.1 Représentation des modèles hybrides : modèle Amont, modèle Aval et modèle unificateur

1.1 L'approche hybride Amont

Dans cette approche le réseau de neurone sert de frontal au réseau bayésien, tel qu'il devient un sélecteur (ou estimateur de sélection) des descripteurs pertinents, cette sélection va nous permettre une optimisation réelle du volume des données qui seront utilisés dans la classification avec le réseau bayésien.

Plusieurs études ont démontré qu'un perceptron multicouches entraîné dans des conditions adéquates est asymptotiquement équivalent à un estimateur de probabilité a posteriori d'appartenance à une classe (HOPFIELD 1987; BOURLARD 1990; GISH 1990; MORGAN 1990; RICHARD 1991). Et grâce à ses performances obtenues avec l'algorithme du rétropropagation du signal d'erreur, nous allons l'utiliser dans notre étude comme architecture neuronale.

1.2 L'approche hybride Aval

Une autre solution trouvée pour la réalisation du système hybride consiste à utiliser le réseau neuronal comme post-processeur d'un réseau bayésien. Cette solution combine au mieux la capacité de remédier aux défaillances dans les données d'un réseau bayésien et le pouvoir discriminatoire d'un réseau de neurones. La méthode consiste à mettre en entrée du réseau de neurones les meilleurs variables obtenues par le réseau bayésien avec une méthode de sélection de variables. Nous allons étudiée aussi cette approche pour confronter les résultats obtenus avec ceux de l'approche amont mais nous nous allons pas tros détaillée dessus.

1.3 Le modèle hybride unificateur

Une autre voie d'hybridation consiste à concevoir des modèles s'appuyant sur les deux théories des réseaux bayésiens et des réseaux de neurones de façon à progresser vers un formalisme unique. En effet, les modèles présentés aux paragraphes 1.1 et 1.2 manquent d'un tel cadre formel unifié ce qui nous pousse à dire que cette unification est difficile à réaliser et sorte presque de l'impossible car la démarche de conception de chaque réseau et les méthodes les construisant se diffèrent complètement; néanmoins, ce modèle reste un créneau de recherche scientifique.

2. Nécessité de sélection de variables (descripteurs)

En pratique, une classification doit être estimée à partir de corpus de données d'apprentissage. La sélection de variables constitue un élément important dans une stratégie de conception d'un modèle par apprentissage; elle contribue en effet à la diminution de la complexité d'un modèle. Le problème de la détermination des entrées pertinentes se pose de manière très différente selon les applications envisagées. Si le processus que nous voulons modéliser est un processus industriel conçu par des ingénieurs, le problème est important mais pas crucial, car, en général, nous connaissons bien les grandeurs qui interviennent et les relations causales entre celles-ci. Ainsi, dans un procédé de soudage par points, nous faisons fondre localement les deux tôles à souder en faisant passer un courant électrique très important (quelques kilo ampères) pendant quelques millisecondes, entre deux électrodes qui exercent une pression mécanique sur les tôles (figure 3.2).

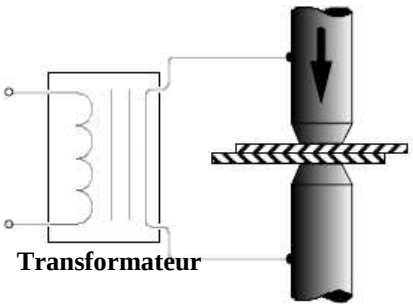

Figure 3.2 : Schéma d'un processus industriel : le soudage par points

La qualité de la soudure, qui est caractérisée par le diamètre de la zone fondue, dépend évidemment de l'intensité du courant, de la durée pendant laquelle il est appliqué, de l'effort exercé par les électrodes pendant le passage du courant et pendant la phase de solidification, de l'état de surface des électrodes, de la nature des tôles, et de quelques autres facteurs qui ont été très largement étudiés en raison de l'importance industrielle du procédé. Nous connaissons donc la nature des entrées désirables pour un modèle; il peut être néanmoins utile de faire un choix parmi ces grandeurs, en ne prenant en considération, en entrée du modèle, que celles qui agissent de manière très significative sur le processus (c'est-à-dire celles dont l'effet est plus important que l'incertitude de la mesure) **[7]**. De même, lorsque nous modélisons un processus physique ou chimique bien connu, nous déterminons généralement, par une analyse préalable du problème, les variables qui ont une influence sur le phénomène étudié; dans ce cas, une étape de sélection des variables n'est pas toujours nécessaire.

En revanche, ce n'est pas le cas lorsque nous cherchons à modéliser un processus économique, social ou financier (déterminer la solvabilité d'un client qui demande un crédit ou la qualité d'une entreprise), ou encore un processus physico-chimique complexe ou mal connu (prédire une propriété chimique d'une molécule) ; les experts du domaine peuvent donner des indications sur les facteurs qu'ils estiment pertinents, mais il s'agit souvent de jugements subjectifs qu'il faut mettre à l'épreuve des faits ou la détermination des entrées pertinentes peut être beaucoup plus délicate. Nous sommes alors conduit à retenir un grand nombre de variables candidates (descripteurs), potentiellement pertinentes. Néanmoins, la complexité du modèle croît avec le nombre de variables. Conserver un contrôle sur le nombre de variables est donc un élément important dans une stratégie de modélisation qui cherche à maîtriser la complexité des modèles. Ce problème n'est pas spécifique aux réseaux de neurones ou aux réseaux bayésiens : il se pose pour toutes les techniques de modélisation, qu'elles soient linéaires ou non. Les résultats de la sélection de

variables sont susceptibles de remettre en cause des idées reçues concernant le phénomène à modéliser, ou, au contraire, de conforter des conjectures ou des intuitions concernant l'influence des variables candidates sur la grandeur à modéliser. Au-delà de l'analyse fine des résultats, un des objectifs de la sélection de variables est de produire un espace de représentation plus performant. Nous verrons, dans ce qui suit, quelques méthodes de sélection de variables.

3. Méthodes de sélection de variables (descripteurs)

3.1 Etapes de sélection de variables

Il existe un grand nombre de méthodes de sélection de variables basées sur plusieurs approches, cependant ces méthodes nécessitent toujours :

• De définir un critère de pertinence des variables pour la prédiction de la grandeur à modéliser;

• De ranger les variables candidates par ordre de pertinence avec une procédure de recherche;

• De définir un seuil qui permette de décider que l'on conserve ou que l'on rejette une variable ou un groupe de variables.

Cette stratégie est applicable à toute méthode de sélection de variables fondée sur un classement des variables par ordre de pertinence. Nous allons essayer, dans ce qui suit, de mieux détailler ces trois critères.

3.1.1 Critère de pertinence (mesure d'évaluation)

Les mesures de pertinence associées aux méthodes de sélection de variables sont souvent basées sur des heuristiques calculant l'importance individuelle de chaque variable dans le modèle obtenu après apprentissage. Ces heuristiques sont nombreuses, mais peuvent être classées selon leurs similarités d'après l'algorithme de classification et le domaine d'application. Nous pouvons se référer à **[22]**, **[27]** pour une explication détaillée de ces mesures.

3.1.2 Procédure de recherche

La stratégie la plus naturelle pour le choix d'un ensemble de descripteurs consiste à partir d'un ensemble de variables candidates aussi grand que possible, de comparer les performances de celui-ci à tous les modèles dont les entrées sont des sous-ensembles de

l'ensemble des variables candidates. Il faudrait donc examiner, évaluer et comparer les performances de l'ensemble des 2^k -1 sous-ensembles possibles de k variables. Cette solution, dont la complexité croit exponentiellement avec le nombre de variables, est certes optimale, mais de mise en œuvre très lourde et inapplicable pour des valeurs, mêmes modérées, de k. Les procédures de recherche couramment utilisées sont donc très souvent des heuristiques basées sur des parcours séquentiels de recherche **[28]** forward ou backward (Figure 3.3). Une grande partie des méthodes de sélection de variables sont des méthodes backward, où les variables sont éliminées grâce à des considérations sur les paramètres du réseau et/ou sur les données. Il existe aussi des méthodes de construction incrémentales de réseaux qui peuvent être considérées comme des méthodes de sélection forward où des variables sont itérativement ajoutés en entrée **[29] [30]**.

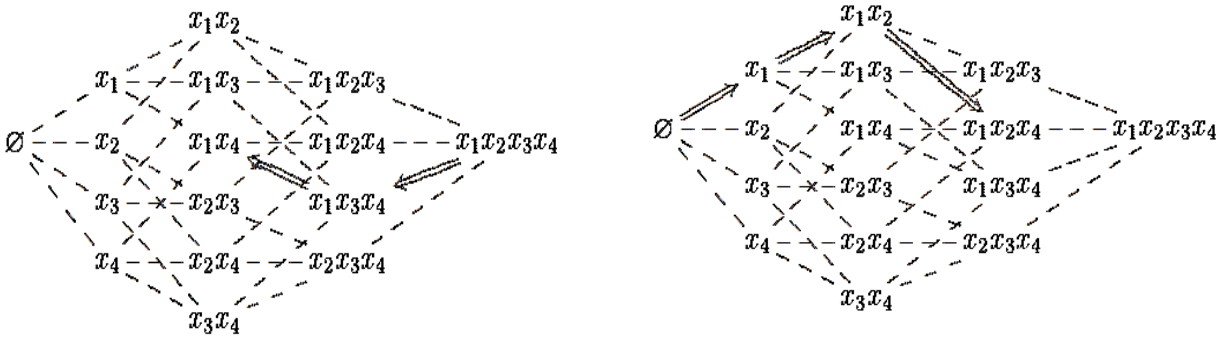

Figure 3.3 : Méthode de sélection de variables : Backward - Forward

Un problème se pose lorsque sont utilisés conjointement une évaluation individuelle des variables et un parcours séquentiel : les dépendances entre les variables ne sont pas prises en compte explicitement. De plus, à cause de la non-linéarité des Réseaux, la corrélation entre variables n'est plus un indicateur satisfaisant de leur dépendance. Certaines méthodes de sélection de variables ignorent simplement ce problème, d'autres proposent d'éliminer une variable à la fois et de réapprendre ensuite le nouveau réseau ainsi obtenu avant d'évaluer l'importance des variables restantes. Cette solution permet de tenir compte des dépendances entre variables que le réseau aura découvert grâce au ré-apprentissage. Le problème de l'initialisation des poids se pose aussi au moment du ré-apprentissage : faut-il partir des poids obtenus avec le réseau précédent ou les réinitialiser (à zéro ou aléatoirement) ? Ce problème reste ouvert, mais il semble judicieux de ré-initialiser les poids aléatoirement à chaque étape pour obtenir de meilleures performances, mais avec un apprentissage souvent plus long **[25]**.

3.1.3 Critère d'arrêt

Une fois que la méthode d'évaluation et celle de recherche ont été fixées, certaines méthodes de sélection de variables examinent tous les sous-ensembles fournis par la méthode de recherche. Une bonne heuristique, dont la complexité est suffisamment raisonnable dans la plupart des applications, est d'estimer l'erreur en généralisation pour les différents sous ensembles de variables sélectionnés. L'ensemble de variables idéal est celui qui donne les meilleures performances. L'erreur en généralisation peut être estimée grâce à un ensemble de validation, par validation croisée ou par d'autres estimations algébriques comme FPE (Final Prediction Error) **[31]**. Plusieurs mesures ont été proposées en statistiques **[34]** ou pour les réseaux de neurones **[32]**, **[33]**. La plupart des méthodes de sélection de variables utilisent des techniques assez rudimentaires pour arrêter la sélection : certaines méthodes fixent un seuil par rapport au critère de pertinence, d'autres classent juste les variables en fonction de l'estimation de l'erreur en généralisation. Ces techniques donne généralement des résultats probants surtout lorsque la méthode de sélection est choisit en tenant compte de l'algorithme de classification et des données à exploiter.

3.2 Quelques méthodes de sélection de variables

La détermination des variables importantes (pertinentes) est un problème essentiel dans l'identification de modèles. De nombreuses publications essayent de procéder à un état de l'art des différentes méthodes utilisées **[21] [22] [23] [24] [25]**. Néanmoins, il n'y a pas de méthode miracle, et dans certains cas, d'autres méthodes peuvent se révéler plus efficaces. Une synthèse très complète des méthodes modernes de sélection de variables est présentée dans l'ouvrage **[19]**. Cependant ces études comparatives permettent de tirer quelques conclusions sur les méthodes de sélection de variables. Tout d'abord, il n'existe pas de méthode de sélection qui soit meilleure que les autres. Par contre, les résultats vont dépendre de la politique choisie par rapport aux différents critères utilisés.

Parmi les méthodes les plus utilisées :

- La méthode de la variable sonde **[8]** : elle est simple, fondée sur des principes solides; elle a été validée sur une grande variété d'applications; elle comporte deus phases :
    1. Classement des entrées par ordre de pertinence décroissante par rapport à la sortie par orthogonalisation de Gram-Schmidt **[58]**,
    2. Elimination des entrées non pertinente.

- La méthode STEPDISK **[8]**: Elle vise à identifier quelle est la combinaison de variables la plus performante pour bien expliquer une typologie. Elle permet notamment d'identifier les variables qui sont pertinentes pour la suite de l'analyse et celles qui ne le sont visiblement pas. Elle fait appel aux méthodes : Forward, Backward ou Stepwise. Forward suppose que le modèle entre d'abord une variable qui explique le mieux la typologie, puis elle cherche la prochaine variable qui explique le mieux, ... et ainsi de suite. Backward introduit d'abord toutes les variables puis elle se débarrasse des variables qui n'expliquent rien. Stepwise est intermédiaire dans le sens où elle commence comme une Forward puis teste à chaque étape s'il n'y a pas moyen d'éliminer une variable entrée auparavant.
- La méthode FCBF (Fast Correlation-Based Filter) **[38]** : FCBF est un algorithme de filtre rapide basé sur la corrélation entre les données, il s'est montré efficace dans la suppression des variable non pertinentes et redondantes.
- La méthode WRAPPER **[24]** : couplée avec le modèle bayesien naif est a priori la meilleure puisqu'elle optimise explicitement le critère de performance (généralement le taux d'erreur), sa stratégie de recherche peut être; très simple avec l'approche d'ajouter et de retirer au fur et à mesure un descripteur à la solution courante; ou bien très élaborée avec des approches basées sur des métaheuristiques. L'approche simple convient dans la plupart des cas. En effet, elle lise naturellement le parcours de l'espace des solutions.
- La méthode OBD (Optimal Brain Damage) **[35]** : comme plusieurs méthodes de sélection de variables, OBD est inspirée des techniques d'élagage des poids dans le réseau. La décision de supprimer un poids est faite selon un critère de pertinence. Une connexion est coupée si sa pertinence est faible **[25]**.

On peut également souhaiter diminuer le nombre de variables en réduisant la dimension de l'espace de représentation de la grandeur que l'on cherche à modéliser (processus porte uniquement sur les entrées). Les principales méthodes utilisées dans ce but sont l'Analyse en Composantes Principales (ACP), l'Analyse en Composantes Indépendantes (ACI, ou ICA pour Independent Component Analysis) ou encore l'Analyse en Composantes Curvilignes (ACC). Ces méthodes sont bien décrites dans le chapitre 3 de l'ouvrage **[13]**.

4. Stratégie de conception hybride

4.1 Description de la stratégie

Dans cette section, nous montrons comment les différentes tâches à accomplir doivent être articulées entre elles pour concevoir un modèle hybride par apprentissage (collecte des données, sélection de variables (descripteurs), classification, apprentissage, évaluation). Nous commençons, dans un premier temps, par rappeler que le but principal de ce travail est la conception d'une méthode de classification utilisant les réseaux de neurones et les réseaux bayésiens face aux différentes données, notamment des données altérées (manquantes) et permettant une optimisation des descripteurs, et pour cela, nous supposons que les étapes de collecte des données et de prétraitement de celles-ci ont été effectuées, notamment en ce qui concerne le problème des variables continues[1] utilisées par les réseaux de neurones et variables discrètes[2] utilisées par les réseaux bayésiens.

Notre stratégie peut être résumée globalement de la façon suivante (Figure 3.4) :

1. Choix de la base de données,
2. Application de la classification sur les données intégrant la totalité des descripteurs avec la technique du premier réseau en utilisant une technique d'évaluation adéquate.
3. Evaluation du modèle obtenu et calcule du taux d'erreur obtenu qui servira de référence,
4. Sélection des descripteurs pertinents avec une méthode de sélection de variables et l'algorithme du premier réseau.
5. Réduction de la base de données avec seulement les descripteurs pertinents obtenus avec la méthode de sélection de variables,
6. Application de la classification avec la technique du premier réseau sur la nouvelle base de donnée (BDD réduite)
7. Evaluation du modèle obtenu et calcule du taux d'erreur obtenu,
8. Comparaison des résultats de 3 et 7 pour voir les conséquences de la réduction de la BDD sur le critère de performance,
9. Application de la classification sur la BDD réduite avec l'algorithme du deuxième réseau en utilisant une technique d'évaluation adéquate,

---

[1] Une variable est dite continue si elle prend ses valeurs dans un intervalle (classe)

[2] Une variable est dite discrète si elle ne prend qu'un nombre fini de valeurs (modalités)

10. Evaluation du modèle obtenu et calcule du taux d'erreur obtenu,
11. Comparaison des résultats de 3 et de 10,
12. Interprétation des résultats.

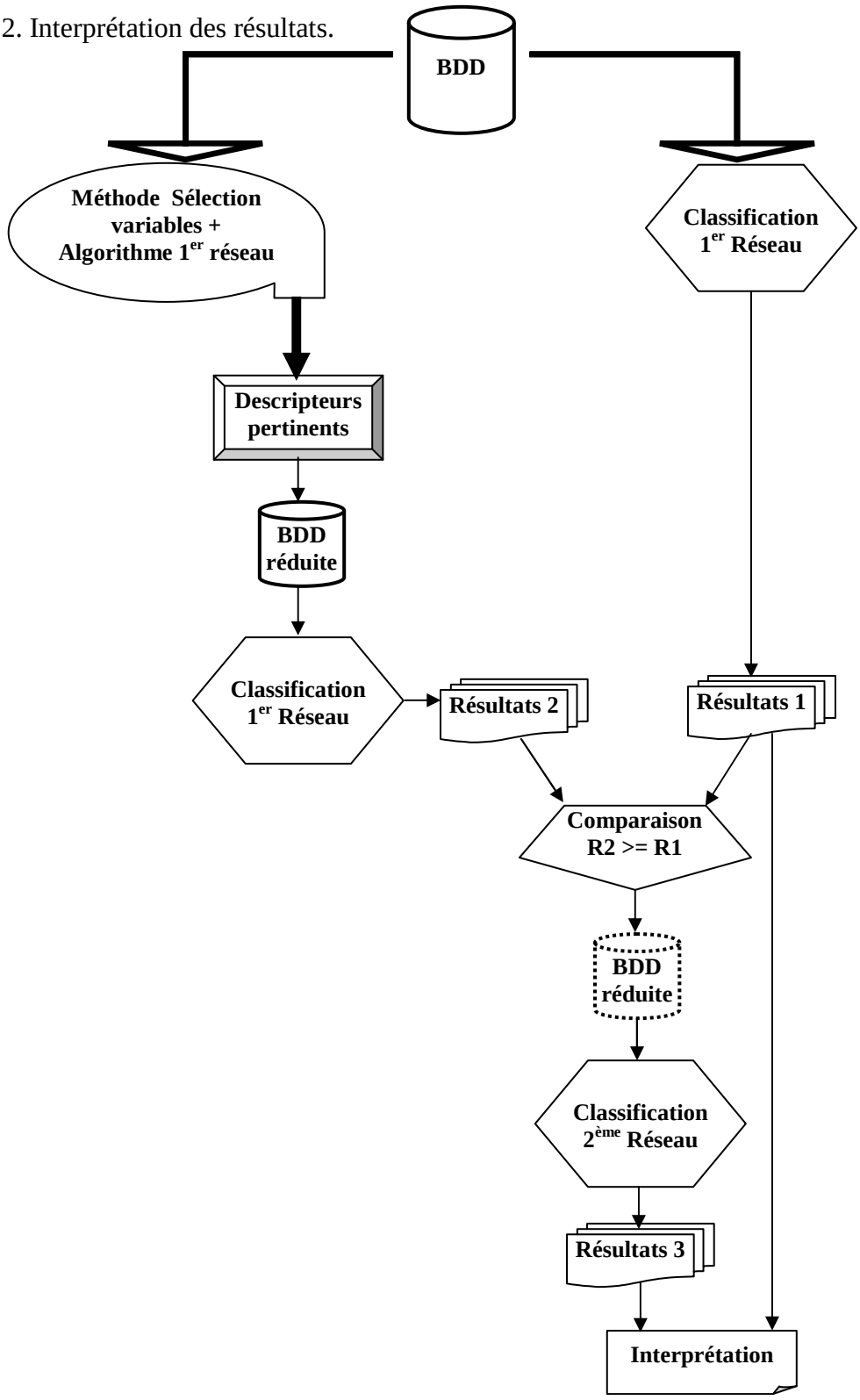

Figure 3.4 : **Stratégie de conception hybride**

Ces différentes étapes seront développées dans la suite du chapitre d'après les modèles amont et aval.

4.2 Techniques d'évaluation de la classification

4.2.1 Matrice de confusion

La matrice de confusion, dans la terminologie de l'apprentissage supervisé, est un outil servant à mesurer la qualité d'un système de classification en confrontant les vraies valeurs avec les valeurs prédites. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (ou de référence). Un des intérêts de la matrice de confusion est qu'elle montre rapidement si le système parvient à classifier correctement. Cette notion peut bien sûr s'étendre à un nombre quelconque de classes. On peut bien sûr normaliser cette matrice pour en simplifier la lecture : dans ce cas, un système de classification sera d'autant meilleur que sa matrice de confusion s'approchera d'une matrice diagonale.

		Prédite		
		Positifs	Négatifs	Total
Réelle	Vrais	A	b	a+b
	Faux	C	d	c+d
	Total	a+c	b+d	N

Figure 3.5 : Matrice de confusion à deux classes

Quelques indicateurs :

- Vrais Positifs VP = a
- Faux Positifs FP = c
- Taux d'erreur = (c+b)/n
- Taux de VP = a/(a+b)
- Taux de FP = c/(c+d)
- Précision = a/(a+c)
- Spécificité = d/(c+d) = 1 – Taux de FP

4.2.2 Taux d'erreur et apprentissage

L'évaluation des classifieurs est une question récurrente en apprentissage supervisé. Parmi les différents indicateurs existants, la performance en prédiction calculée à l'aide du taux d'erreur qui est un critère privilégié. Du moins dans les publications scientifiques car, dans les études réelles, d'autres considérations sont au moins aussi importantes : l'évaluation des performances en intégrant les coûts de mauvaise affectation, l'interprétation des résultats, etc. Le taux d'erreur théorique est défini comme la probabilité de mal classer un individu dans la population. Bien entendu, il est impossible de le calculer directement, essentiellement parce qu'il n'est pas possible d'accéder à toute la population. Nous devons produire une estimation. Qui dit estimation dit utilisation d'un échantillon, un

estimateur de bonne qualité doit être le moins biaisé possible, et le plus précis possible. L'estimateur trivial est le taux d'erreur empirique que l'on appelle également taux d'erreur en resubstitution. Il s'agit de réappliquer le modèle sur l'échantillon de données qui a servi à le construire (Figure 3.6).

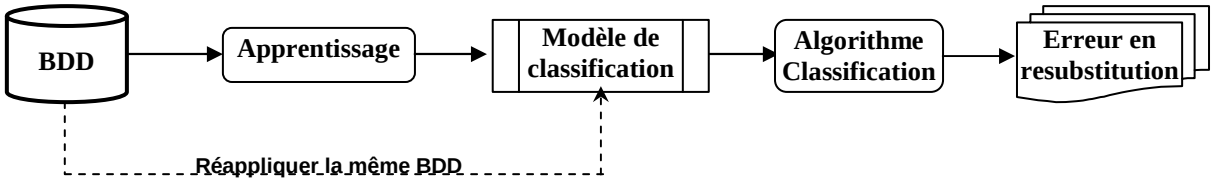

Figure 3.6 : **Calcul de l'erreur en resubstitution**

Tous les logiciels produisent cette estimation accompagnée d'un tableau de contingence, dite matrice de confusion (Figure 3.5), qui croise, pour l'ensemble des individus de l'échantillon, la vraie modalité prise par la variable à prédire et la modalité affectée par le modèle de classement. La principale reproche que l'on peut adresser à l'erreur en resubstitution est qu'elle est fortement biaisée, on parle de « biais d'optimisme». En effet, elle sous estime souvent le taux d'erreur théorique. La raison est simple, le fichier de données est à la fois « partie », il a servi à construire le modèle, et « juge », il est utilisé pour savoir si le modèle classe correctement. Donc plus une observation pèse sur sa prédiction, plus l'optimisme sera important. De manière générale, il y a un fort optimisme lorsque les techniques collent exagérément aux données (ex. un Perceptron avec trop de neurones dans la couche cachée, un arbre de décision trop grand) ou lorsque la dimensionnalité est trop importante au regard du nombre d'observations.

Pour se dégager de cet écueil, on conseille souvent de subdiviser l'échantillon en 2 parties : une première partie, dit fichier d'apprentissage, utilisée pour construire le modèle; et une seconde partie, dit fichier test, utilisée pour évaluer les performances du modèle. L'erreur ainsi mesurée est appelée « erreur en test » et elle estime de manière non biaisée l'erreur théorique (Figure 3.7).

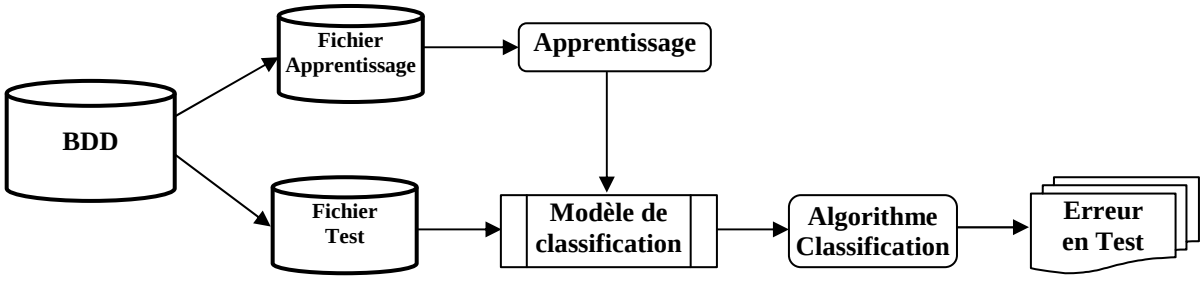

Figure 3.7 : Technique d'évaluation : Apprentissage –Test

Tout serait donc parfait si nous ne sommes pas confronté à un nouveau problème : quelle proportion des données devons nous consacrer à l'apprentissage ? La pratique veut que l'on réserve entre 60% et 70% pour l'apprentissage (généralement les deux tiers qui donnent la meilleure estimation). Mais au delà de cette règle empirique, nous devons arbitrer entre deux exigences contradictoires, d'autant plus crucial que l'échantillon est de petite taille : plus nous réservons des données pour l'apprentissage, moins l'estimation de l'erreur en test sera précise ; si nous favorisons la partie test, nous pénalisons l'apprentissage, nous retirons de l'information qui peut s'avérer déterminante pour la construction d'un modèle efficace.

Le problème qui se pose : peut-on quantifier un échantillon de petite taille, si nous devons travailler sur un problème réel ? En deçà du millier d'observations, on pourrait considérer que la base est de petite taille. En réalité, il faut surtout appréhender le problème sous l'angle du rapport entre la complexité du modèle et le nombre d'observations. Dans un contexte où les observations sont relativement rares, comment estimer au mieux le taux d'erreur théorique si nous souhaitons consacrer l'ensemble du fichier à la construction du modèle ?. Les techniques de rééchantillonnage permettent de répondre à cette question ou nous devons nous tourner vers ces techniques lorsqu'il n'est pas possible de réserver une partie des données pour l'évaluation des modèles. Nous étudierons plus particulièrement la Validation Croisée et le Bootstrap. Ces techniques utilisent les données disponibles en les séparant (de manières différentes) en deux ensemble, que nous appellerons ensemble d'apprentissage et ensemble de validation. Il s'agit de répéter plusieurs fois, sous des configurations pré définies, le schéma apprentissage-test. Cette façon de procéder entraîne le fait que toutes les données (apprentissage et validation) sont utilisées pour construire. Les modèles intermédiaires, élaborés lors des apprentissages répétés, servent uniquement à l'évaluation de l'erreur. Ils ne sont pas accessibles à l'utilisateur, ils n'ont pas d'utilité intrinsèque. L'estimation de l'erreur de généralisation obtenue est donc optimiste et plus réaliste.

4.2.3 Le Bootstrap

Le Bootstrap **[44]** est une technique statistique récente car elle repose sur des calculateurs puissants. Proposée en 1979 par BRADLEY EFRON, elle permet d'estimer l'écart entre l'erreur d'apprentissage (risque empirique) et l'erreur de généralisation (risque fonctionnel). Dans le cadre du Data Mining, le « bootstrap.632 » pour le calcul d'erreurs en

classification, permet d'engendrer des modèles plus robustes, c'est à dire aux performances d'un niveau sensiblement constant face à des données de petite taille ou de valeurs inconnues. Il s'agit d'une technique de rééchantillonnage avec remise sans recours à de nouvelles observations, qui génère des échantillons de taille N et inclut donc la possibilité d'avoir les mêmes données dans des échantillons différents de même taille. La méthode présente un avantage important : Le nombre *B* d'échantillons aléatoires différents possibles à partir de *N* individus est pratiquement infini dès que *N* est de quelques dizaines car $B = N^N$.

Algorithme Bootstrap

• Répéter B fois (on parle de réplications)

- ✓ Tirage avec remise d'un échantillon de taille n → Ω_b
- ✓ Distinguer les individus non échantillonnés $\Omega_{(b)}$
- ✓ Apprentissage du modèle sur Ω_b
- ✓ Erreur en resubstitution sur Ω_b
- ✓ Erreur en test sur $\Omega_{(b)}$
- ✓ Calcul de l'optimisme O_b (l'optimisme est l'écart entre l'erreur en test et l'erreur en resubstitution)

• Sur l'échantillon complet, calcul de l'erreur en resubstitution

1. $e_B = e_r + \frac{\sum_b O_b}{B}$ …..……....(11) C'est l'optimisme qui est estimé, Il est utilisé pour corrigé l'erreur en resubstitution

2. $e_{0.632B} = 0.368 \times e_r + 0.632 \times \frac{\sum_b O_b}{B}$ 0.632 Bootstrap …(12)→Pondérer par la probabilité qu'un individu fasse partie de Ω_b sur une réplication

3. Il existe une troisième formule pour corriger le biais induit par la méthode d'apprentissage, c'est une méthode utilisé pour le calcul de l'erreur estimé lors des méthodes de classification des données : 0.632+ bootstrap **[45][46][47][48][49]**

$$e_{0.632+B} = (1-\varpi) \times e_r + \varpi \times \frac{\sum_b O_b}{B} \quad \text{…..……(13)}$$

Tel que :

$$\varpi = \frac{0.632}{1 - 0.368R} \quad \text{……(14)}$$

$$R = \frac{\frac{\sum_b O_b}{B} - e_r}{\gamma - e_r} \qquad \text{……(15)}$$

$\gamma = \sum p_l(1-q_l)$ ……(16) ; p_l et q_l sont les probabilités respectives a priori et posteriori des classes

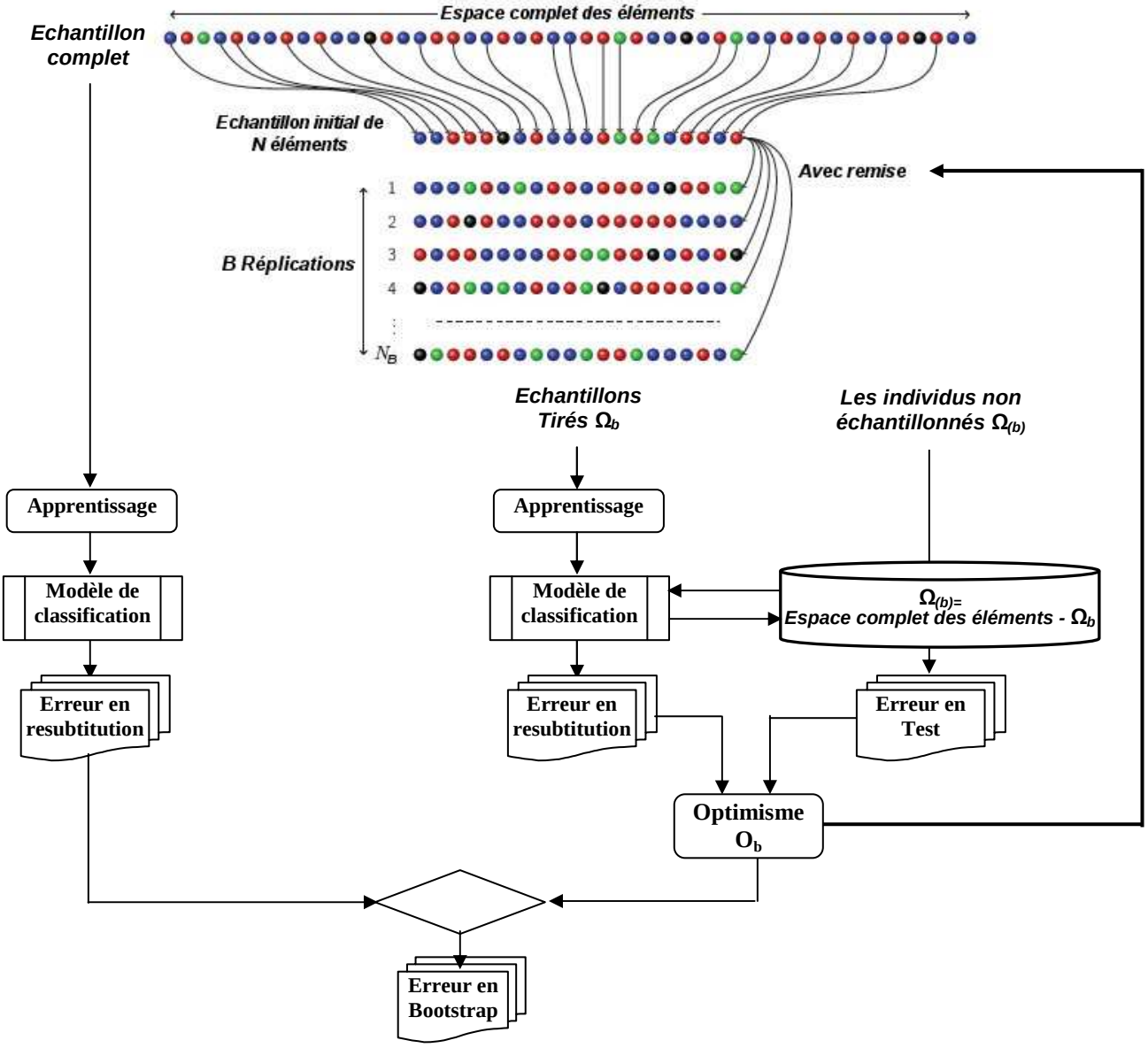

Figure 3.8 : Technique d'évaluation : Bootstrap

4.2.4 La validation croisée

La validation croisée **[44]** est une fonction importante dans le développement et l'optimisation de modèles d'exploration de données, elle est systématiquement proposée dans la très grande majorité des logiciels de Data Mining. Schématiquement, la validation

consiste a subdiviser aléatoirement les données en K blocs. Nous réitérons le processus suivant, en faisant tourner les sous-échantillons : apprentissage du modèle sur les (K-1) blocs, évaluation du taux d'erreur en prédiction sur le $K^{ème}$ bloc. Le taux d'erreur en validation croisée est la moyenne des taux d'erreurs ainsi collectés. C'est un estimateur de meilleure qualité que le taux d'erreur en resubstitution.

Algorithme validation croisée

- Subdiviser l'échantillon en K blocs
- Pour chaque k :
  - ✓ Construire le modèle d'apprentissage avec les $n-n_k$ individus
  - ✓ Calculer l'erreur en test sur n_k → e_k
- Calculer la moyenne e_{cv} des erreurs en test

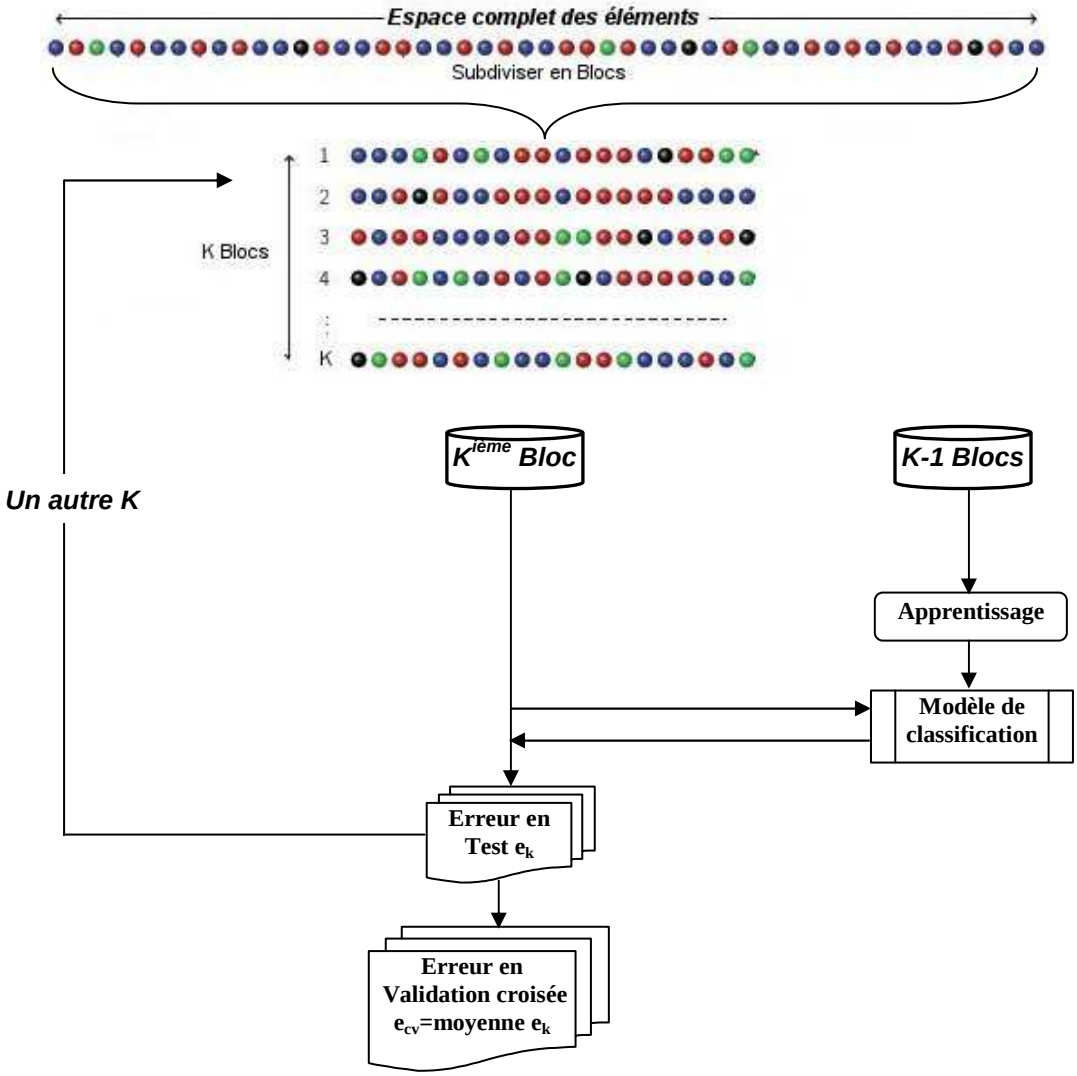

Figure 3.9 : **Technique d'évaluation : Validation Croisée**

5. Conclusion

Dans ce chapitre, nous avons réussit à trouver une démarche conceptuelle qui nous permet de concrétiser notre objectif de combiner les réseaux de neurones et les Réseaux Bayésiens pour une meilleure classification des données, ou nous avons présenté deux idées d'hybridation (amont et aval) dans le cadre de la sélection de variables pour l'apprentissage supervisée. Cependant, il faut choisir les types de réseaux de neurones et de réseaux bayésiens ainsi que la méthode de sélection de variables et trouver les outils adéquats pour mettre en œuvre cette idée en prenant en charge les différentes méthodes d'évaluation qui donnent une confiance certaine aux résultats.

.

CHAPITRE 4 :

IMPLEMENTATION & RESULTATS

CHAPITRE 4
IMPLEMENTATION & RESULTATS

1. Introduction

La fouille de données, ou de manière générique l'Extraction de Connaissance à partir de Données est un domaine de recherche qui a véritablement pris son essor au milieu des années 90. S'il est toujours possible de discuter quant à sa véritable originalité par rapport au traitement statistique des données qui existe déjà depuis longtemps, il est indéniable en revanche que son avènement s'est accompagné d'une forte accélération de la diffusion de logiciels spécialisés estampillés Data Mining. Les raisons sont multiples; nous pouvons citer entre autres : la rencontre entre des communautés différentes (apprentissage automatique, bases de données, analyse de données, statistiques); le développement d'Internet qui a permis la diffusion à peu de frais des logiciels, avec pour certains des codes sources; l'élargissement du champ d'application du traitement des données. Phénomène révélateur de cette métamorphose, des logiciels de traitement statistique ayant pignon sur rue depuis plusieurs années ont modifié leur positionnement, en se contentant bien souvent d'un remodelage de leur interface et de l'adjonction de méthodes issues de l'apprentissage automatique. Nous pouvons classer les logiciels en deux catégories très distinctes : les logiciels commerciaux et les plates-formes d'expérimentation. Les premiers proposent une interface graphique conviviale, ils sont destinés à la mise en oeuvre de traitements sur des données en vue du déploiement des résultats. Les méthodes disponibles sont peu référencées, il est de toute manière impossible d'accéder à l'implémentation. Les seconds sont constitués d'un assemblage de bibliothèques de programmes. Un chercheur peut facilement accéder au code source, vérifier les implémentations, ajouter ses propres variantes et mener de nouvelles expérimentations comparatives. Ces plates-formes ne sont guère accessibles à des utilisateurs non informaticiens.

Dans ce chapitre, nous montrons comment mettre en œuvre notre approche et comment lire et exploiter les résultats.

2. Présentation des logiciels utilisé

2.1 Introduction

Pour implémenter notre stratégie de conception, il fallait trouvé les outils adéquats pour réussir une bonne schématisation des données et ainsi réussir une interprétation facile et juste des résultats, pour cela, nous avons opter pour les logiciels open source qui permettre une meilleure maîtrise des algorithmes, voir même développer de nouveaux composants représentant l'approche suivie. Le choix d'un seul logiciel était difficile, il a été principalement défini sur les performances des logiciels disponibles en open source et gratuitement, ainsi que la disponibilité des méthodes de sélection et des algorithmes utilisés dans notre étude, ceci, nous a amené à tester notre étude sur les deux logiciels TANAGRA **[39]** et SIPINA **[59]**.

2.2 TANAGRA

Présentation

TANAGRA est un logiciel open source gratuit dédié à la fouille de données. Il s'adresse à deux types de publics. D'un côté, il présente une interface graphique aux normes des logiciels de fouille de données actuels, y compris les logiciels commerciaux, le rendant ainsi accessible à une utilisation de type « chargé d'études » sur des données réelles. De l'autre coté, du fait que le code source est librement disponible et l'architecture interne très simplifiée, il se prête à une utilisation de chercheurs qui veulent avant tout expérimenter de nouvelles techniques en améliorant celles déjà implémentées ou en introduisant de nouvelles dans le domaine de la statistique exploratoire, de l'analyse de données, de l'apprentissage automatique et des bases de données. Donc, il offre aux chercheurs et aux étudiants une plate-forme de Data Mining facile d'accès, respectant les standards des logiciels du domaine, notamment en matière d'interface et de mode de fonctionnement, et permettant de mener des études sur des données réelles et/ou synthétiques. Sa présentation est adoptée sous forme de diagramme de traitements où l'enchaînement des opérations est symbolisé par un graphe orienté dans lequel transitent les données, les noeuds du graphe représentent un traitement effectué sur les données. Le logiciel se comporte comme une plate-forme d'expérimentation dans laquelle les chercheurs puissent ajouter simplement leurs méthodes. Son architecture logicielle est la plus simplifiée possible de manière à porter l'essentiel de l'effort au développement des

méthodes. Le chercheur est allégé de toute la partie ingrate de la programmation de ce type de logiciel, notamment la gestion de données et la mise en forme des sorties. Point très important, la disponibilité du code source est un gage de crédibilité scientifique; elle assure la reproductibilité des expérimentations publiées par les chercheurs et, surtout, elle permet la comparaison et la vérification des implémentations. Le logiciel est référencé sur le principal portail anglo-saxon du data mining « http://www.kdnuggets.com ».

Diagramme de traitement

Les séquences d'opérations appliquées sur les données sont visualisées à l'aide d'un graphe. Chaque noeud représente un opérateur soit de fouille de données, soit de modélisation, soit de transformation. Il est donc susceptible de produire de nouvelles données. Il est désigné également sous le terme de composant en référence au vocabulaire utilisé dans les outils de programmation visuelle. L'arête reliant deux noeuds représente le flux des données vers l'opérateur suivant. Ce mode de représentation qui est le standard actuel des logiciels de fouille de données autorise (au contraire des logiciels pilotés par menus) la définition d'enchaînement d'opérations sur les données. En même temps il affranchit l'utilisateur (au contraire des outils fonctionnant avec un langage de script) de l'apprentissage d'un langage de programmation. Dans TANAGRA, seule la représentation arborescente est autorisée, la source de données à traiter est unique. La fenêtre principale du logiciel est subdivisée en trois grandes zones (Figure 4.1) :

(a) en bas la série des composants disponibles regroupés en catégories ;

(b) sur la gauche, le diagramme de traitements, représentant l'analyse courante ;

(c) dans le cadre de droite, l'affichage des résultats consécutifs à l'exécution de l'opérateur sélectionné.

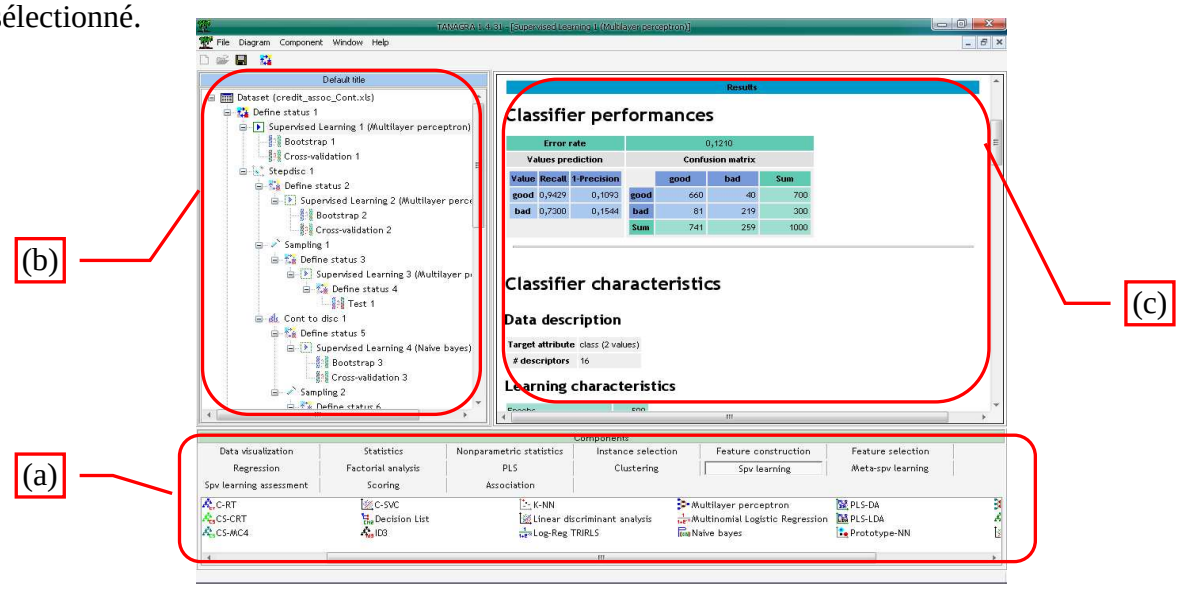

Figure 4.1 : La fenêtre principale du logiciel TANAGRA

La sauvegarde de la séquence d'instructions (le programme en quelque sorte) définie par un utilisateur peut être réalisée soit sous un format binaire, soit sous la forme d'un fichier texte. Seul le programme est sauvegardé, les résultats ne le sont pas. Le format texte permet à un utilisateur avancé de le manipuler directement afin de définir un nouveau diagramme de traitements. C'est le format préconisé si le volume de données à traiter n'est pas trop important, en effet il suffit de relancer l'exécution pour obtenir automatiquement les nouveaux résultats lorsque les données ont été modifiées. En revanche, si la base est de grande taille, il est préférable d'utiliser le format binaire qui optimise le temps d'exécution sur les entrées-sorties. Son principal inconvénient étant la nécessité de ré-importer les données si elles sont mises à jour. La possibilité d'enchaîner des méthodes d'apprentissage à travers le diagramme de traitements rend aisé la combinaison des méthodes sans avoir à utiliser un langage de script. Il est ainsi très facile de mettre en oeuvre des méthodes spécifiques.

Les composants

Un composant représente un algorithme de traitement de données. Toutes les méthodes sont référencées. Le code source étant accessible librement, il est possible de consulter ce qui est implémenté. Les composants ont pour point commun de prendre en entrée des données en provenance du composant qui le précède; de procéder à des calculs donnant lieu à l'affichage d'un rapport au format HTML (Figure 4.3); ils sont le plus souvent paramétrables (Figure 4.2); et enfin, ils transmettent aux composants en aval les données en y ajoutant éventuellement des données produites localement, les prédictions par exemple pour les méthodes supervisées.

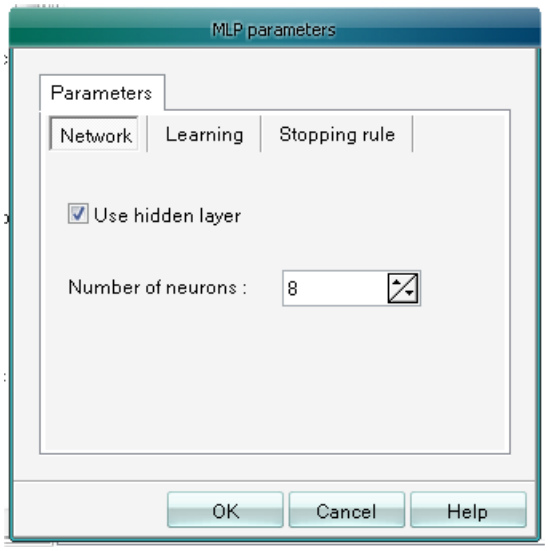

Figure 4.2 : Boîte de paramétrage de la méthode réseau de neurones

En termes de programmation, l'insertion d'un nouveau composant dans le logiciel est simplifiée. La procédure est toujours la même : dériver quelques classes à partir des prototypes existants, dessiner une icône appropriée, mettre à jour le fichier de configuration. La hiérarchie interne des classes respectant le découpage en famille des composants, il est aisé d'identifier rapidement la classe ancêtre adéquate.

Standardisation des sorties

La mise en forme des sorties constitue un travail important dans les logiciels commerciaux.. TANAGRA produit directement des sorties au format HTML. La solution est souple, autant que pour la production de sorties au format texte; elle permet des mises en formes élaborées sans avoir à procéder à une normalisation compliquée. Cette standardisation permet d'exporter facilement les résultats vers un logiciel d'édition, EXCEL par exemple, pour un éventuel post-traitement. Les sorties comportent généralement deux parties : la description des paramètres du traitement demandé, et les résultats associés. Dans la figure 4.3, nous montrons un exemple de sorties de la méthode des réseaux de neurones. La construction des rapports HTML constitue une fraction faible de l'implémentation des méthodes, ce qui était le but recherché.

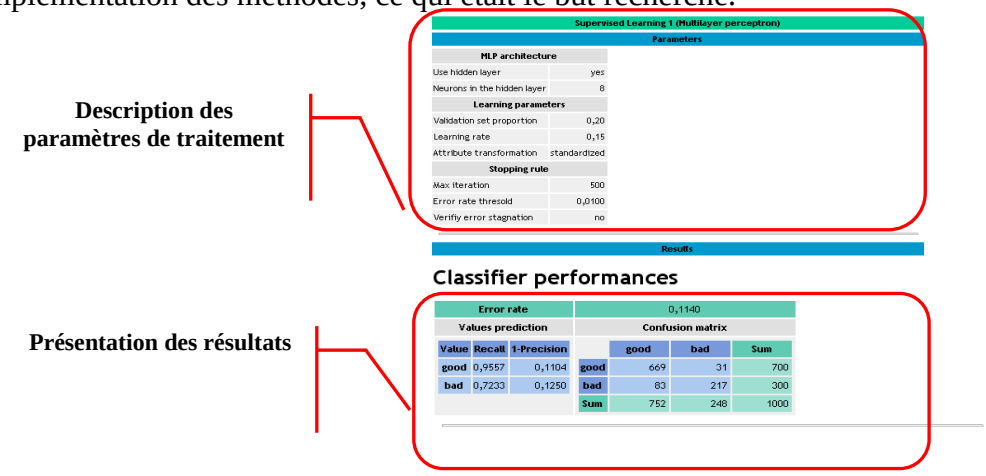

Figure 4.3 : Rapport au format HTML du réseau de neurones avec un perceptron multicouches

Mode d'exécution : interactif ou par lots (batch)

A l'instar des logiciels utilisant la représentation des traitements sous forme de diagramme, TANAGRA peut être piloté de manière interactive. En s'appuyant sur les outils de l'interface, le praticien peut construire manuellement ses traitements, puis lancer l'exécution de l'ensemble. Ce diagramme peut être sauvegardé pour une exécution ultérieure; dans ce cas le praticien a le choix entre deux options : soit il charge le diagramme via les menus adéquats puis lance l'exécution; soit il passe le diagramme à l'exécutable via la ligne de commande. Cette deuxième option définit un autre mode

d’exécution : le traitement par lots. Dans ce cas, le logiciel charge le diagramme, l’exécute puis s’arrête, sans jamais faire apparaître la fenêtre principale du logiciel. Tous les résultats sont consignés dans des rapports au format HTML générés automatiquement. Certains composants, les évaluations de l’apprentissage supervisé notamment, ont la possibilité d’inscrire leurs résultats dans un fichier commun défini à la conception du diagramme. Ce mode d’exécution ouvre la porte aux expérimentations. En effet, il est possible de générer de nombreuses variantes d’un scénario de traitements, en modulant les paramètres d’une méthode de fouille de données. La sauvegarde du diagramme au format texte se prête remarquablement bien à ce type d’étude. Grâce à sa simplicité et sa lisibilité, le chercheur peut le manipuler aisément; il peut aussi le générer automatiquement par programme pour les expérimentations à grande échelle.

Structures internes et performances

L’interface de TANAGRA est en langue anglaise. Le code source est commenté en français. Le logiciel est implémenté en Pascal Objet, il est compilé avec la version 6 de Borland Delphi, il est aisé de recompiler le logiciel à partir du code source. L’exécutable est un fichier binaire qui fonctionne exclusivement sur le système d’exploitation windows. Une particularité du logiciel réside dans l’obligation de charger, sous forme recodée, la totalité des données en mémoire. Une observation est codée sur 1 octet pour une variable discrète qui ne pourra donc pas prendre plus de 255 modalités; une variable continue est codée sur 4 octets, limitée à 8 chiffres significatifs. Un fichier d’un million d’observations avec 1000 variables continues occupe donc approximativement 382 Mo en mémoire centrale. Tout dépend dès lors des problèmes que l’on souhaite appréhender. Un PC de bureau doté de 512 Mo de mémoire vive par exemple peut traiter l’ensemble des clients d’une grande banque régionale pour un ciblage marketing. En revanche, traiter l’ensemble des transactions journalières d’une enseigne de grande distribution en chargeant les données en mémoire paraît inconcevable. Certaines techniques telles que les règles d’association, très gourmandes dès lors que l’on charge tout en mémoire, s’avèrent très vite impraticables lorsque la taille de la base augmente.

2.3 SIPINA

SIPINA est un logiciel gratuit de Data Mining avec menus spécialisé dans l'induction des arbres de décision. Curieusement, c'est un des très rares outils en libre accès intégrant des fonctionnalités interactives lors de la construction d'un arbre de décision.

Fonctionnalités qui, pourtant, font tout le sel de cette méthode dans une activité de fouille de données. SIPINA implémente également d'autres méthodes supervisées à l'instar des réseaux de neurones et des réseaux bayésiens. Depuis le développement et la diffusion de TANAGRA en 2004, il est conseillé systématiquement d'utiliser ce dernier. Il comporte non seulement les méthodes supervisées mais également une grande majorité des techniques de statistique et d'analyse de données telles que les analyses factorielles, la classification automatique, etc., et la possibilité de les faire coopérer entre elles. Les différentes versions de SIPINA sont disponibles sur le web depuis 1995. La version actuelle n'a plus évolué depuis 2000. Elle est néanmoins distribuée car il y a très peu d'équivalents gratuits au monde. Le site de distribution en anglais est régulièrement consulté encore à ce jour, et le logiciel téléchargé car il implémente des méthodes non disponibles dans d'autres logiciels tel que la méthode WRAPPER pour la sélection des variables, et cet avantage qui a poussé à choisir ce logiciel pour notre étude.

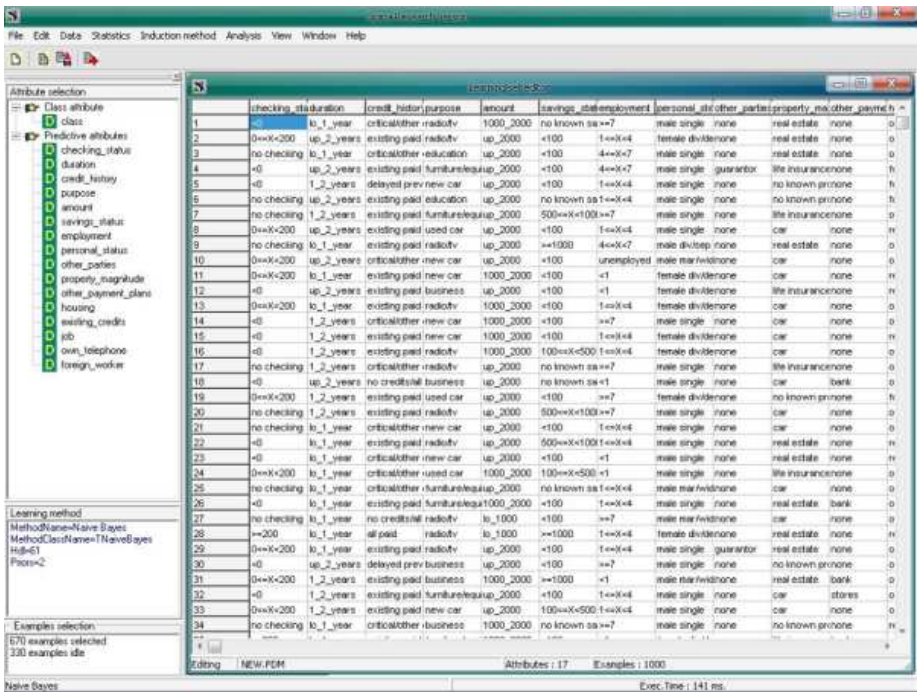

Figure 4.4 : La fenêtre principale du logiciel SIPINA

3. Choix de la BDD

Après avoir choisit deux logiciels de Datamining pour réaliser notre démarche et ainsi éliminer la contrainte de l'influence de la plate de forme sur les résultats, il fallait trouver un moyen pour s'assurer que la base de données choisit pour notre étude n'aura pas aussi d'influence sur les résultats de la stratégie de conception proposée, pour cela nous avons tester notre démarche sur trois base de données de petits volumes, car le volume joue surtout sur la qualité de classification (avec une BDD volumineuse = classification

meilleure = taux d'erreur est minime), et la chose qui nous intéresse c'est l'amélioration faite de la classification hybride.

Comme indiqué précédemment, le but principal de cette étude n'est pas la prédiction d'un nouveau cas mais plutôt c'est de montrer les performances de l'hybridation des réseaux de neurones et des réseaux bayésiens dans la classification des données. Cela minimise l'importance de la nature de la base de données dans notre étude, pour cela nous avons opté pour une étude sur des bases de données de différentes natures :

- La première concerne le ciblage marketing (ou scoring) qui est certainement une des applications les plus populaires de la classification en datamining. Prenons le cas d'une campagne qui offre à la vente un produit ou un service et qui cible une base de donnée clientèle. En règle générale, environ 1% de la base de clientèle réagiront, c'est-à-dire achèteront le produit ou le service qu'il leur est proposé. Un publipostage envoyé à 100 000 clients choisis au hasard générera ainsi environ 1 000 ventes. Les techniques de classification permettent un marketing fondé sur la relation avec la clientèle, en identifiant quels clients risquent le plus de réagir à la campagne. Si le taux de réponse peut être augmenté de 1% à 1,5%, par exemple, alors 1 000 ventes pourront être réalisées avec 66 666 envois seulement, ce qui réduit le coût du publipostage d'un tiers. La base de données utilisée est fournit par Mr GARY SAARENVIRTA[3]. Dans cette base de données, chaque cas représente un compte. La variable objective est une variable de réponse indiquant si un consommateur a réagi ou non à une campagne de mailing direct pour un produit spécifique. "Vrai" ou "réponse" est représenté par positive, "Faux" ou "non-réponse" par négative. Les données sont extraites d'un jeu de données beaucoup plus large et les données utilisées concernent 1 079 personnes ayant réagi, ainsi que 1 079 personnes n'ayant pas réagi, soit un total de 2 158 cas. Le fichier contient 177 descripteurs, la majorité des variables ont été normalisées.

- La deuxième concerne la classification des champignons. Cet ensemble de données comprend des descriptions d'échantillons hypothétiques correspondant à 23 espèces de champignons. Chaque espèce est identifiée comme étant comestible ou vénéneuse à partir de sa description (taille, couleur, etc). La variable Classe est celle que l'on veut prédire. Il n'y a pas de règle simple pour déterminer la comestibilité d'un champignon. Cette base de donnée a été choisit pour quatre raisons principales :

[3] Anciennement manager dans la banque The Loyalty Group, maintenant chez IBM

1. le volume de la base de données qui est de l'ordre de 8124 enregistrements, chacun identifié par 22 descripteurs.

2. la consistance de cette base de données utilisée dans plusieurs travaux de recherches ou les données sont indépendantes entre eux.

3. l'existence de quelques enregistrements avec des données altérées ou manquantes.

4. le risque des champignons sauvages pour la vie humaine et les cas d'intoxication et même de décès enregistrés durant l'année 2008 en Algérie ou il est très difficile de distinguer les champignons toxiques des champignons comestibles.

- La troisième concerne la classification des clients douteux, ou une banque lance une analyse sur sa base transactionnelle lorsque un client se présente pour demander un crédit, la banque est devant un embarras, surtout pour les clients qu'elle ne connaît pas encore. Le banquier cherche à savoir si le crédit va être rembourser ou non car le client ne dira jamais qu'il ne va pas remboursé. Va-t-il accepter la demande de prêt, ce qui est légitime pour toute banque en vue d'accroître le profit ? ou bien va-t-il refuser la demande pour ne pas risquer de tomber sur un mauvais payeur, dans ce cas, se sera une perte sèche pour la banque?. Pour répondre à ces questions, le banquier peut inférer sur le comportement futur du client à partir de la base de données des anciens clients. Les connaissances acquises peuvent être l'analyse d'une transaction prédiction. Le service de crédit de cette banque est contraint de prédire les clients douteux et les clients sérieux avec un minimum de descripteurs de ces clients et avec des probabilités de ceux ne pouvant pas rembourser a temps. Ce choix a été poussé par l'intérêt de ce sujet mondialement surtout avec les conséquences de la crise économique 2008 qui a touché presque tout les pays du monde due essentiellement aux crédits bancaires non remboursés à temps, pour cela, nous allons utilisé une base de données réelle de 1000 enregistrements avec 16 descripteurs des clients et des données hautement corrélés.

Dans les deux premières bases de données, nous avons utilisé un fichier de données dans lequel nous avons introduit une colonne supplémentaire permettant de désigner les individus à utiliser pour l'apprentissage et ceux à utiliser lors de l'évaluation afin de mieux évaluer les performances avec la méthode apprentissage-test et éliminer l'influence des données sur les résultats de l'algorithme de prédiction.

Le tableau suivant montre les principales caractéristiques des trois Bases de données :

Tableau 4.1 : Caractéristiques des BDD utilisées

Abréviation BDD	$BDD_{Scoring}$	$BDD_{Champignon}$	BDD_{Credit}
Définition BDD	Ciblage clientèles (scoring)	Classification des champignons	Classification des clients douteux
Nombre d'enregistrements	2 158	8 124	1 000
Nombre de descripteurs	177	22	16
Corrélation des données	Normale	Très Faible	Haute
Qualité BDD	Bien remplie	Manquante	Bien remplie

4. Réseau de neurones en amont d'un Réseau Bayésien

4.1 Présentation

Notre étude est basée surtout sur cette approche. Le réseau de neurones choisi est un Perceptron Multicouches, utilisant l'algorithme de rétropropagation d'erreur, l'architecture du réseau adopté comporte trois couches, une couche d'entrée, une couche cachée et une couche de sortie. La taille de la couche d'entrée a été fixé d'après le nombre de descripteurs de la base de données, la taille de la couche cachée est variable entre les différentes base de données essayant une optimisation du taux d'erreur alors que nous avons fixé le nombre de neurones de la couche de sortie à un qui correspondent au nombre de classe car nous avons utilisé dans toute l'étude un problème à une classe, et pour une étude plus juste et plus crédible, il était évident qu'il fallait opter pour la même architecture du réseau de neurones dans l'étude d'une même base de données. Un tel choix nous permet d'éliminer une éventuelle influence de cette architecture sur les résultats produits par l'algorithme dans les différents cas.

Pour réalisé le schéma en dessous (Figure 4.5), il fallait trouvé une méthode d'optimisation des descripteurs performante, ceci étant très difficile avec le grand nombre de méthodes disponibles, néanmoins, nous avons choisis la méthode STEPDISC qui donne des résultats satisfaisants surtout avec un estimateur tel que le réseau de neurones, comme nous proposons une autre méthode que nous avons appelé la méthode MOYVAR qui est basée sur l'algorithme du réseau de neurones pour déterminer les descripteurs pertinents.

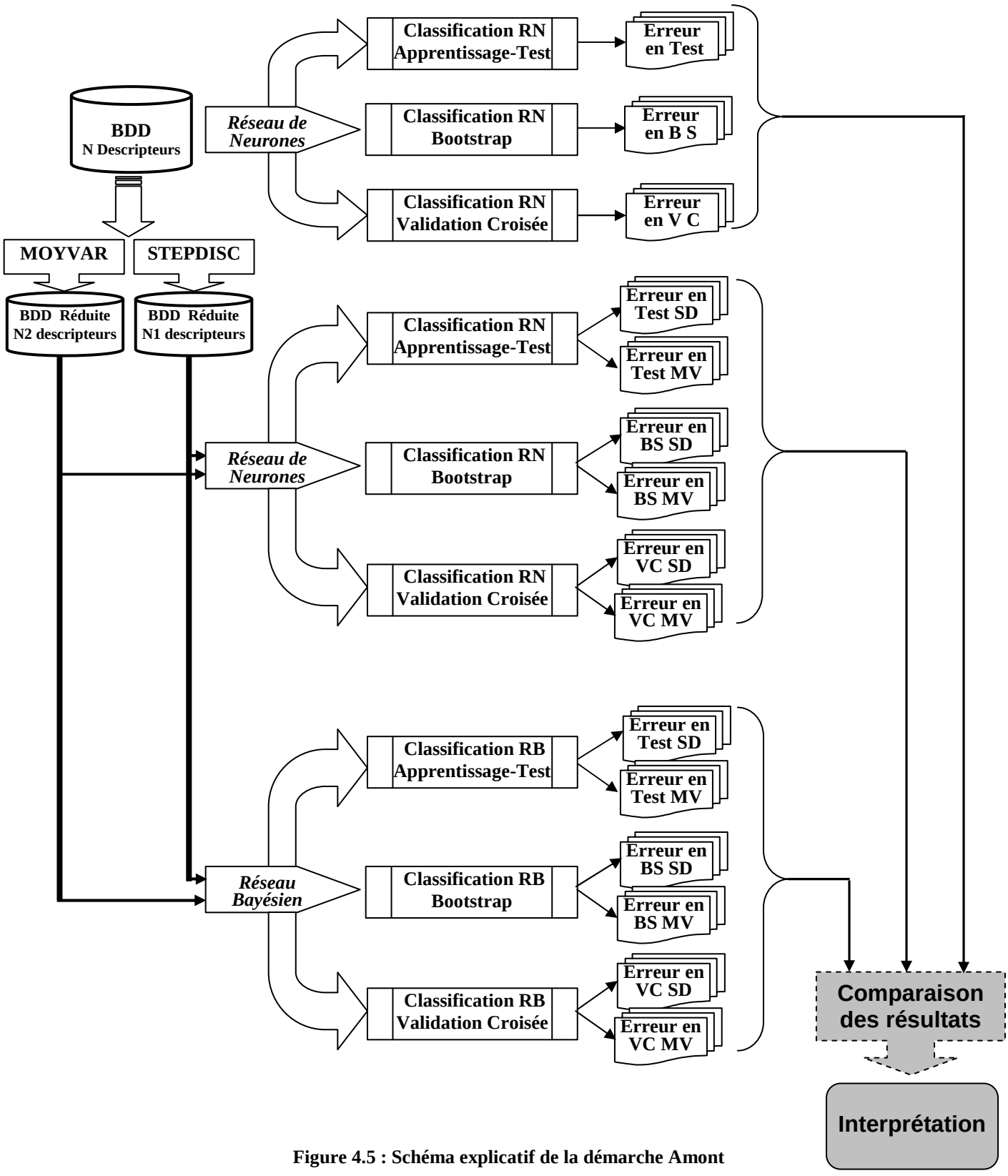

Figure 4.5 : Schéma explicatif de la démarche Amont

4.2 STEPDISC

La méthode STEPDISC **[8]** est une méthode de sélection de variables pertinentes en apprentissage supervisé, elle utilise le critère du Lambda de Wilks **[45]**. Géométriquement, il s'agit de trouver le sous-espace de représentation qui permet un écartement maximal entre les centres de gravité des nuages de points conditionnels c.-à-d. les nuages de points associés à chaque valeur de la variable à prédire. Elle est donc particulièrement bien adaptée à l'analyse avec les réseaux de neurones qui a un caractère discriminatoire. Le Lamda de Wilks représente le rapport entre l'inertie intra classes et l'inertie totale **[51]**. si les nuages sont totalement confondue, Lambda =1; plus Lambda se rapproche de 0, plus les nuages conditionnels sont distincts (Figure 4.6). Donc le but de Lambda de Wilks est de tester si plusieurs groupes d'observations multivariées ont des moyennes significativement différentes.

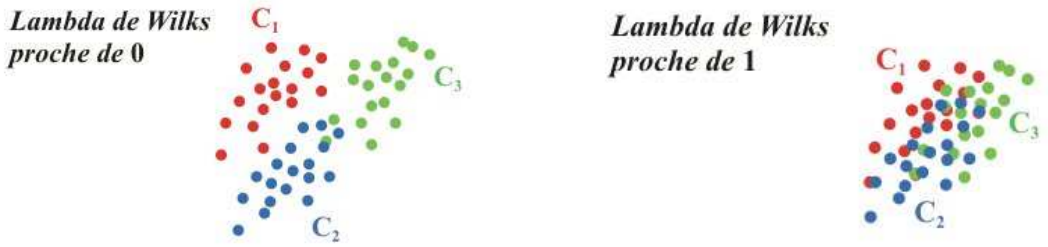

Figure 4.6 : Représentation de Lambda de Wilks

Les logiciels affichent parfois la valeur de Lambda de Wilks relatif à chacune des variables prises une par une. Ces valeurs peuvent alors être considérées comme des mesures des pouvoirs discriminants individuels des variables. Le Lambda de Wilks est utilisé pour la sélection de variables en classification supervisée. Sa distribution est très complexe, dont les distributions sont approximativement une distribution en Chi-2, ou une distribution *F* de Fisher. Heureusement, des transformations mathématiques simples le transforme en d'autres statistiques selon la distribution F de Fisher **[45]** relatifs à un certain sous-ensemble de variables et à ce même sous-ensemble augmenté d'une variable supplémentaire. Donc, un test F permet alors de déterminer laquelle des variables non encore incorporées au modèle augmente le plus la séparabilité des classes.

La méthode STEPDISC fait appel aux trois méthodes suivantes :

- La méthode STEPDISC Backward **[8]**: nous cherchons à éliminer du modèle complet le descripteur le moins significatif; à cette fin nous comparons tous les sous-modèles à k-1 variables (k nombre de descripteurs), nous choisissons le meilleur d'entre eux en recherchant la variable dont le retrait entraînerait la dégradation la plus faible du

lambda, et nous le comparons au modèle complet; si le sous-modèle est meilleur que le modèle complet, nous conservons ce sous-modèle si la dégradation n'est pas statistiquement significative et nous l'itérons la procédure à partir de celui-ci; si le modèle complet est meilleur, nous le conservons.

- La méthode STEPDISC Forward **[8]** : nous commençons par le modèle le plus simple, dont la sortie indépendante des entrées, est simplement la moyenne des sorties du processus : c'est donc un modèle à zéro descripteurs; nous le comparons aux k modèles à un descripteur, nous choisissons le meilleur induisant la meilleure amélioration du Lambda et le sélectionner si l'amélioration est statistiquement significative, et nous l'itérons la procédure jusqu'à ce que l'ajout d'un descripteur n'améliore plus la qualité du modèle.
- La méthode STEPDISC Stepwise **[8]** : cette méthode mixe les deux approches Foward et Backward après avoir ajouté une variable, nous regardons s'il n'est pas nécessaire de retirer certaines variables parmi celles qui ont déjà été introduites. Puis nous regardons à nouveau s'il n'est pas possible d'en ajouter de nouvelles, etc.

4.3 MOYVAR

Cette méthode utilise purement les calculs et les résultats des réseaux de neurones. L'idée nous a été parvenu en collaboration avec M^r^ Ricco Rakotomalala[4] **[50]**. Dans cette approche nous comparons le taux d'erreur (e) du réseau de neurones ou toutes les variables (descripteurs) sont actives avec le taux d'erreur (e1) du réseau de neurones si on désactivait une des variables (nous remplaçons toutes les valeurs par la moyenne). Et pour mesurer l'importance de l'écart, nous utilisons un test de conformité d'une proportion à un standard **[37]** avec la formule suivante :

$$D = \frac{(e1 - e)}{\sqrt{e(1-e)/n}} \qquad \text{…….(17)}$$

Ou n est le nombre total d'observations (enregistrements).

A vrai dire, ce test n'est pas un vrai test au sens statistique puisque le standard est aussi estimé à partir de l'échantillon, et les échantillons ne sont pas vraiment indépendants, néanmoins, l'idée est d'essayer de déterminer les variables qui sont les plus importantes dans la prédiction. Donc, si D est proche de zéro alors la variables a une faible contribution dans la prédiction et son élimination n'influx pas beaucoup sur le taux d'erreur général. Ce

[4] Spécialiste dans la recherche dans : Knowledge Discovery in Databases - Data Mining - Machine Learning

test est représenté avec la variable « statistics » dans le tableau « Attribute Contribution » du logiciel Tanagra (Figure 4.7), le taux d'erreur (e) du réseau, qui sert de référence, est désigné dans la ligne ou aucun descripteur n'est désactivé (none excluded), et le taux d'erreur (e1) de chaque variable est la valeur de « Error rate » dans chaque ligne correspondante à la valeur de la colonne « excluded attribute ».

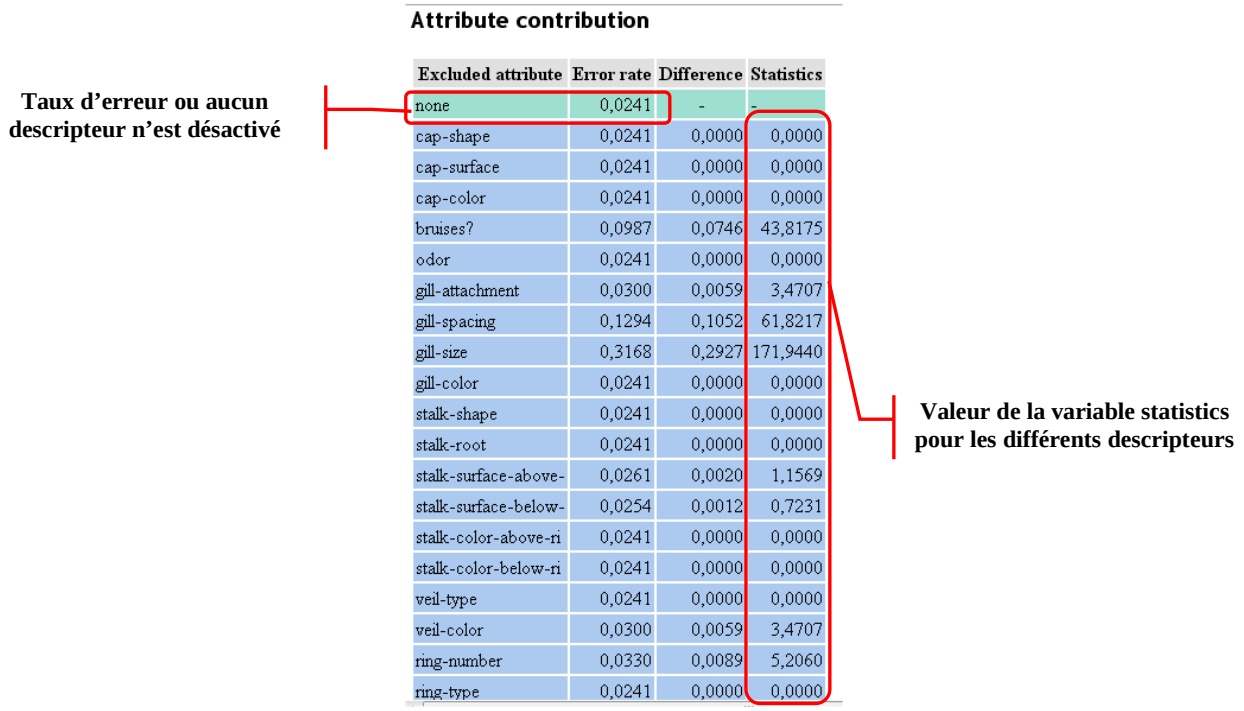

Figure 4.7 : Représentation de la variable statistics dans Tanagra

4.4 Application et Résultats

Après avoir éliminer les contraintes liées aux données et aux paramètres des algorithmes, un autre problème survenu lors des différents tests d'apprentissage qui nous a rendu perplexe, car avec toute les précaution prises pour limités les autres influences, il y avait une petite variation des résultats obtenues pour la même méthode d'apprentissage (avec les mêmes paramètres) et avec la même base de données. Ce problème s'est avéré lié aux performances du matériel utilisé lors de l'apprentissage surtout pour les méthodes de rééchantillonnage ou, d'une machine à l'autre, les résultats peuvent être relativement différents puisque ces méthodes s'appuient sur un générateur de nombres aléatoires indexé sur l'horloge de la machine, il est initialisé de manière différente à chaque fois. Notre souci étant d'éviter à tout prix ce genre de problèmes pour ne se focaliser que sur notre objectif, nous avons fait tout les tests sur le même PC sans avoir à chargé sa mémoire avec d'autre application, et ainsi éliminer l'influence du matériel sur les résultats. L'apprentissage avec l'algorithme du réseau de neurones, surtout lorsque le volume de la base de données

augmente et la nécessité de réitérer les estimations conduit souvent à des temps de traitement importants se chiffrant en heures voire en jours, même avec des serveurs ou micro-ordinateurs puissants, pour cela, les traitements et les résultats obtenues ont été sauvegardé au format HTML. Le choix du format HTML a une seconde conséquence, l'exportation des résultats pour une lecture en dehors du logiciel est simplifiée et il en est de même pour les impressions. Pour ce qui est de la deuxième étape de notre démarche hybride, nous avons utilisé le modèle bayésien naïf.

Vu qu'il n'existe pas d'études qui montrent la technique d'évaluation la plus performante pour le taux d'erreur en classification, notre étude a été faite avec les trois méthodes bootstrap, validation croisée et apprentissage-test (Figure 4.8).

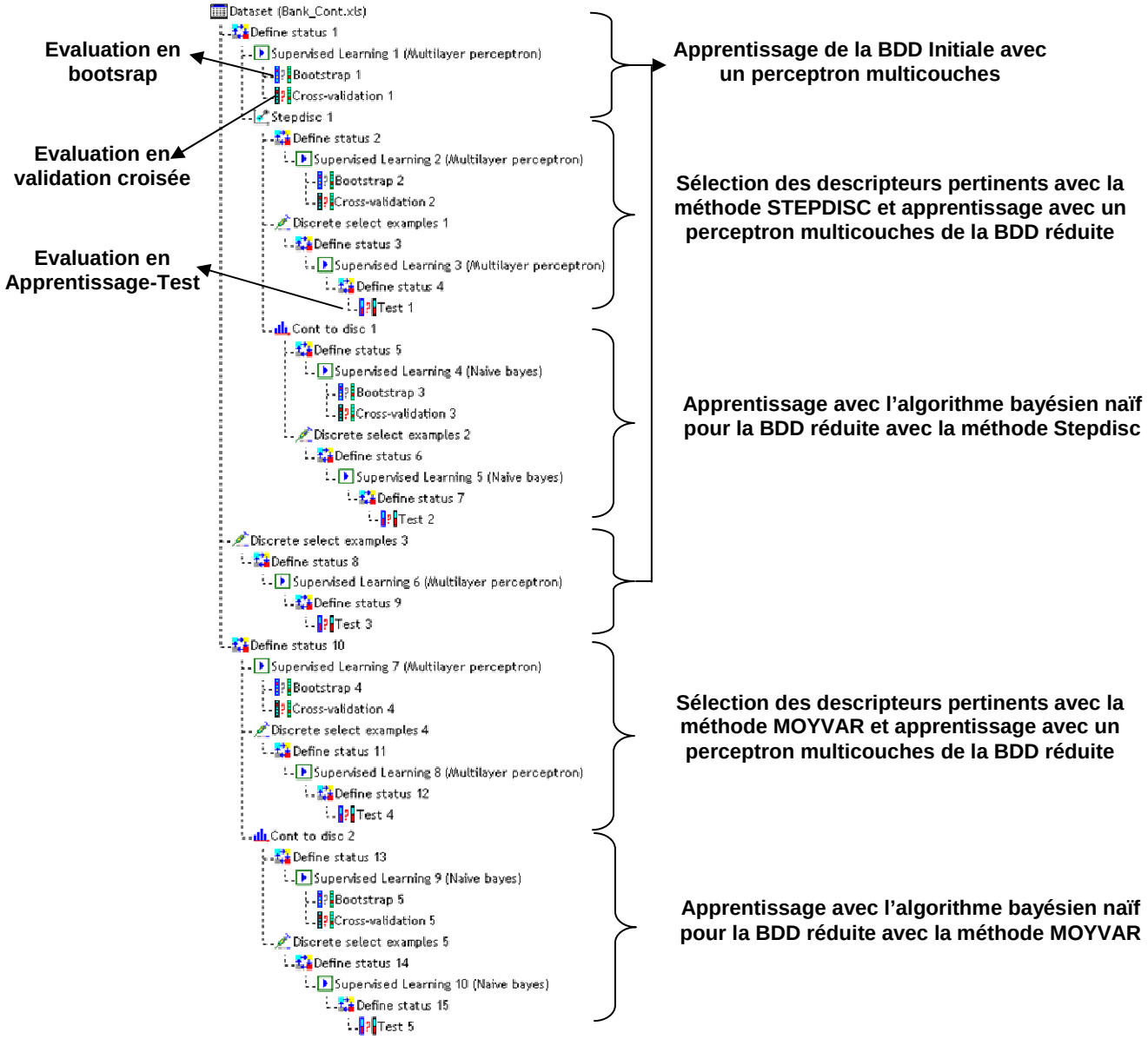

Figure 4.8 : Diagramme de traitement avec les différentes méthodes d'évaluation

Ce diagramme représente toutes les étapes suivies avec la démarche amont pour la première base de données. Le même diagramme est appliqué pour les deux autres bases de données. La démarche amont est déployée totalement avec le logiciel Tanagra et les résultats de cette démarche pour les trois bases de données sont résumés dans les trois tableaux 4.2, 4.3 et 4.4 présentés en dessus.

La première base de données ($BDD_{Scoring}$) : Le tableau 4.2 représente les résultats obtenus en appliquant la classification hybride en amont sur la première base de données concernant le ciblage marketing avec les deux méthodes de sélection de variables STEPDISC et MOYVAR.

Tableau 4.2 : Résultats de la méthode STEPDISC et MOYVAR sur la première BDD

	Réseau de neurones				**Réseau bayésien**
	BDD initiale		**BDD réduite**		
Méthode d'évaluation	**Nombre de descripteurs**	**Taux d'erreur T_1**	**Nombre de descripteurs**	**Taux d'erreur T_2**	**Taux d'erreur Final T_F**
METHODE STEPDISC					
Apprentissage -test	**177**	0.3400	**25**	0.3150	**0.3020**
Bootstrap 0.632+		0.2729		0.2730	**0.2694**
Validation Croisée		0.3219		0.3084	**0.2940**
METHODE MOYVAR					
Apprentissage -test	**177**	0.3400	**167**	0.2900	**0.3300**
Bootstrap 0.632+		0.2729		0.2727	**0.2744**
Validation Croisée		0.3219		0.3033	**0.2935**

L'apprentissage initial avec les réseaux de neurones intégrant la totalité descripteurs a donnée un taux d'erreur de T_1=34,00 % en apprentissage-test, T_1=27,29 % en Bootstrap et T_1=32,19 % en Validation Croisée. Avec la méthode STEPDISC, le nombre de descripteurs a été optimisé considérablement passant de 177 variables à 25 variables contre 167 variables avec la méthode MOYVAR, et malgré cette diminution du volume de la BDD, le taux d'erreur T_2 est resté stable ce qui indique qu'il y avait beaucoup de données bruitées qui n'apportait pas d'amélioration à la classification de la BDD. L'application du réseau bayésien à la BDD réduite pour la première méthode a apporté une amélioration du taux d'erreur T_F avoisinant les 2 % par rapport au taux initial, ceci minimisera certainement le nombre d'envois du publipostage en gardant le même nombre de ventes réalisés, et de ce fait réduire le coût du publipostage des tiers avec un nombre minime de descripteurs qui engendra un gain considérable à la société.

La deuxième base de données ($BDD_{Champignon}$) : Le tableau 4.3 représente les résultats obtenus en appliquant la classification hybride en amont sur la deuxième base de données concernant la classification des champignons avec les deux méthodes de sélection de variables STEPDISC et MOYVAR.

Tableau 4.3 : Résultats de la méthode STEPDISC et MOYVAR sur la deuxième BDD

	Réseau de neurones				**Réseau bayésien**
	BDD initiale		**BDD réduite**		
Méthode d'évaluation	**Nombre de descripteurs**	**Taux d'erreur T_1**	**Nombre de descripteurs**	**Taux d'erreur T_2**	**Taux d'erreur Final T_F**
METHODE STEPDISC					
Apprentissage -test	**22**	0.0255	**19**	0.0193	**0.0042**
Bootstrap 0.632+		0.0179		0.0278	**0.0021**
Validation Croisée		0.0198		0.0262	**0.0014**
METHODE MOYVAR					
Apprentissage -test	**22**	0.0255	**9**	0.0323	**0.0872**
Bootstrap 0.632+		0.0179		0.0272	**0.0929**
Validation Croisée		0.0198		0.0257	**0.0947**

Malgré que la base de données comporte des données manquantes, le taux T_1 est de qualité et il ne dépasse pas les 2.6 % pour les trois méthode d'évaluation, ceci est du principalement aux données qui sont bien structurées et au nombre d'enregistrements relativement grand qui dépasse les 8000 individus. En terme d'optimisation des descripteurs, la méthode MOYVAR a donné meilleurs résultats en portant le nombre de descripteurs a 9 contre 19 pour la méthode STEPDISC, alors qu'il était de 22 initialement. Cette optimisation exagérée a pesé sur la qualité de classification ou les taux T_2 et surtout T_F se sont détériorés considérablement. Nous pouvons expliqué ceci par le principe de la méthode MOYVAR, basé sur une moyenne de toutes les valeurs du descripteur lors de son élimination, donnant sûrement de faux résultats avec des données manquantes ou altérées. Par contre, après une légère détérioration du taux T_2 avec la méthode STEPDISC, l'application du réseau bayésien, a donnée une nette amélioration du taux T_F qui est passé sous la barre de 0.5 %, due au comportement de la méthode de sélection de variable qui a pris ses précaution devant les données utilisées et surtout au réseau bayésiens connu pour son savoir faire avec les données altérées. Ainsi, le taux final obtenu représente presque 88% d'amélioration dans la prédiction par rapport au taux initial obtenant une meilleure précision dans la prédiction avec un minimum de caractéristiques des champignons connues.

La troisième base de données (BDD_{Credit}) : Le tableau 4.4 représente les résultats obtenus en appliquant la classification hybride en amont sur la troisième base de données concernant la classification des clients douteux avec les deux méthodes de sélection de variables STEPDISC et MOYVAR.

Tableau 4.4 : Résultats de la méthode STEPDISC et MOYVAR sur la troisième BDD

	Réseau de neurones				**Réseau bayésien**
	BDD initiale		**BDD réduite**		
Méthode d'évaluation	**Nombre de descripteurs**	**Taux d'erreur T_1**	**Nombre de descripteurs**	**Taux d'erreur T_2**	**Taux d'erreur Final T_F**
METHODE STEPDISC					
Apprentissage -test	**16**	0.2909	**7**	0.2909	**0.2667**
Bootstrap 0.632+		0.2948		0.2828	**0.2675**
Validation Croisée		0.2970		0.3080	**0.2740**
METHODE MOYVAR					
Apprentissage -test	**16**	0.2909	**15**	0.2788	**0.2455**
Bootstrap 0.632+		0.2948		0.2889	**0.2567**
Validation Croisée		0.2970		0.3005	**0.2640**

Ces résultats montrent une stabilité positive du taux d'erreur T_1 après l'application des méthodes de sélection des descripteurs avec élimination de 9 descripteurs pour la première méthode et seulement 1 descripteur pour la deuxième méthode expliqués par le petit volume de la base de données et la corrélation élevés entre les données influençant l'élimination de plusieurs descripteurs par la méthode MOYVAR.

Avec l'application du modèle bayésien naïf, nous remarquons une amélioration du taux d'erreur T_F pour les deux méthodes. Cependant, cette amélioration (avoisinant les 3%) est significative pour la méthode MOYVAR car elle s'adapte mieux pour les base de données de petite taille. Cette amélioration est très bénéfique pour la banque, en terme de questions posées pour le prétendu au crédit par l'optimisation du nombre de descripteurs, et en terme de prédiction du comportement du client par la réduction du taux d'erreur.

Vu que la majorité des études ne trouvent pas une grande différence entre les trois méthodes d'évaluation du taux d'erreur utilisées, et pour une analyse adéquate et juste des résultats obtenues, nous avons utilisé quelques termes que nous expliquons ci-dessus :

- Le taux moyen : représente la moyenne des taux concernant les trois méthodes d'évaluation : Apprentissage-test, Bootstrap 0632+ et validation croisée pour la même base de données et la même méthode de sélection de variables.

T moyen = (T_Aprentissage-Test + T_Bootstrap 0.632+ + T_Validation Croisée) /3 ....(18)

- Le taux d'optimisation : c'est le taux en nombre de descripteurs non pertinents éliminés par la méthode de sélection de variables par rapport au nombre de descripteurs initial.

Taux d'optimisation = (Nb descripteurs éliminés / Nb descripteurs initial) * 100 ….(19)

- Le taux d'amélioration : c'est le gain ou la perte en taux du taux moyen en classification hybride par rapport au taux moyen de la classification initiale sans optimisation de descripteurs.

Taux d'amélioration = ((T_moyen initial – T_moyen final) / T_moyen initial)*100 ….(20)

L'analyse des résultats de la démarche amont pour les trois bases de données est résumée dans le tableau 4.5 :

Tableau 4.5 : Résultats de l'hybridation avec les méthodes STEPDISC et MOYVAR

			$BDD_{Scoring}$	**$BDD_{Champignon}$**	**BDD_{Credit}**
Caractéristiques BDD		Nombre enregistrements	2 158	8 124	1 000
		Nombre descripteurs	177	22	16
		Corrélation donnée	Normale	Très Faible	Haute
		Qualité BDD	Bien remplie	Manquante	Bien remplie
Classification initiale RN		**T_{RN} moyen**	**0.3116**	**0.0211**	**0.2942**
Classification hybride Amont RN → RB	**STEPDISC**	Descripteurs	25	19	7
		Taux d'optimisation	**85,88 %**	**13,64 %**	**56,25 %**
		T_{SD} moyen	**0.2885**	**0.0026**	**0.2697**
		Taux d'amélioration	**07.41 %**	**87,68 %**	**08,33 %**
	MOYVAR	Descripteurs	167	9	15
		Taux d'optimisation	**05,65 %**	**59,09 %**	**06,25 %**
		T_{MV} moyen	**0.2993**	**0.0916**	**0.2554**
		Taux d'amélioration	**03,95 %**	**- 334.12 %**	**13.19 %**

Ce tableau représente le récapitulatif des résultats obtenus en appliquant la classification initiale avec le réseau de neurones et la classification hybride en amont avec les deux méthodes de sélection de variables STEPDISC et MOYVAR sur les trois bases de données.

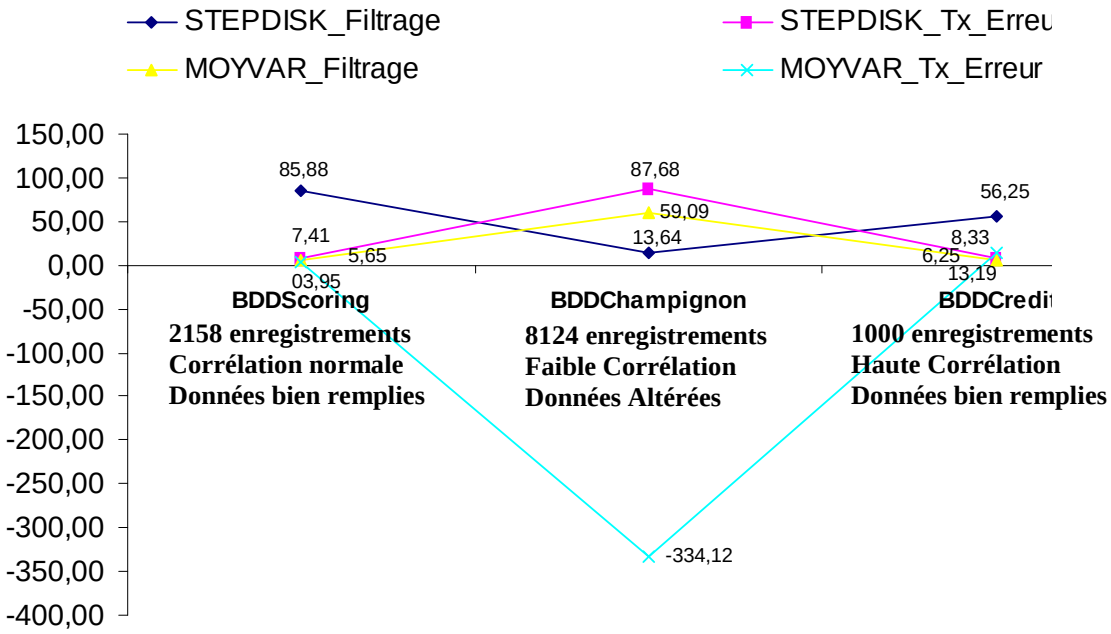

Figure 4.9 : Taux d'optimisation et taux d'amélioration du taux d'erreur avec les deux méthodes (STEPDISC et MOYVAR) sur les trois BDD

La figure 4.9 montre clairement les résultats positifs après l'application de l'hybridation sur les trois BDD dans l'élimination du bruit et l'amélioration du taux de prédiction. Nous constatons que l'hybridation avec la méthode STEPDISC a été très performante en terme de sélection des descripteurs pertinents avec un taux d'optimisation largement positif par rapport au nombre initial des descripteurs dépassant même les 85 % pour la base de données du ciblage marketing, et aussi en terme d'amélioration du taux de prédiction surtout pour la classification des champignons avec un taux d'amélioration de plus de 87 % par rapport au taux initial ou la puissance des réseaux bayésiens s'est bien montré devant les données altérées. D'un autre coté, l'hybridation avec la méthode MOYVAR a montré aussi une grande performance aussi bien pour le filtrage des données que pour l'amélioration du taux de prédiction surtout pour la base de donnée du crédit bancaire avec ses données hautement corrélées. Cependant, nous avons remarqué une grande détérioration du taux de prédiction après l'hybridation pour la base de donnée des champignons ou l'optimisation exagérée a pesé sur la qualité de la classification expliquée par le principe de la méthode MOYVAR, basé sur une moyenne de toutes les valeurs du descripteur lors de son élimination.

5. Réseau de neurones en aval d'un Réseau Bayésien

5.1 Présentation

Motivé par les résultats obtenue par la méthode amont nous avons décidé d'étudier une autre démarche de classification hybride, c'est la démarche réseau de neurones en aval d'un réseau bayésien. Dans cette démarche, nous avons mis en œuvre la méthode WRAPPER **[24]** pour la sélection des descripteurs pertinents avec le modèle bayésien naïf comme algorithme d'apprentissage. Le réseau de neurones choisi pour la deuxième étape est toujours un perceptron multicouches avec l'algorithme de rétropropagation d'erreur. La figure 4.10 représente la démarche suivie détaillée.

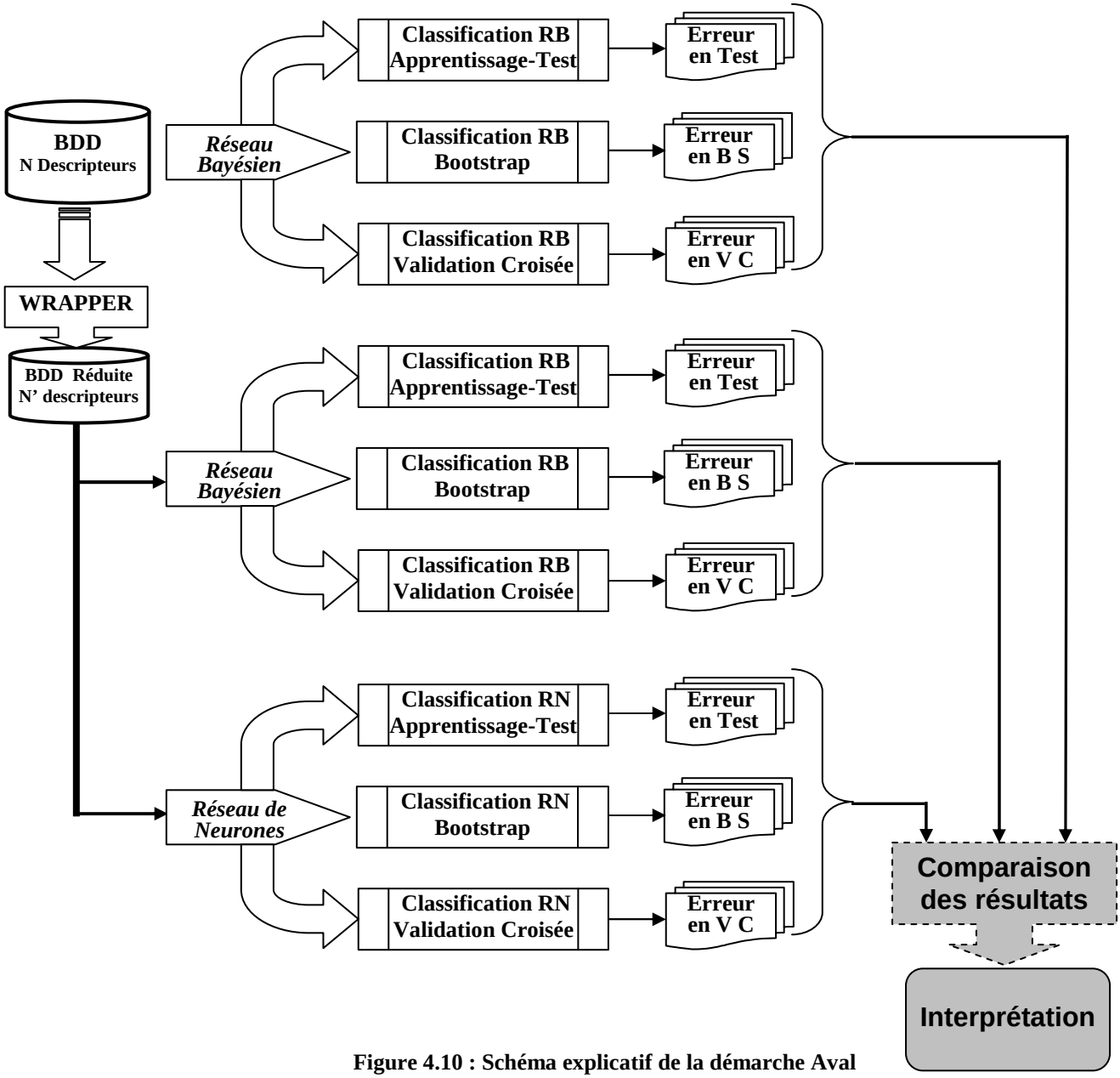

Figure 4.10 : Schéma explicatif de la démarche Aval

5.2 WRAPPER

La méthode WRAPPER a été introduite récemment par JOHN.G, KOHAVI.R et PEGER.K en 1994. De par son principe même, cette méthode génère des sous-ensembles bien adaptés à l'algorithme de classification qui est appelé plusieurs fois à chaque évaluation car un mécanisme de rééchantillonnage est fréquemment utilisé. Un autre avantage est sa simplicité conceptuelle : il n'y a nul besoin de comprendre comment l'induction est affectée par la sélection de variables, il suffit de générer et de tester. Cependant, trois raisons font que cette méthode ne constitue pas une solution parfaite. D'abord, elle n'apporte pas vraiment de justification théorique à la sélection et elle ne nous permette pas comprendre les relations de dépendances conditionnelles qu'il peut y avoir entre les variables. D'autre part la procédure de sélection est spécifique à un algorithme de classification particulier et les sous-ensembles trouvés ne sont pas forcements valides si on change de méthode d'induction. Finalement, et c'est le défaut principal de la méthode, les calculs deviennent vite très longs, voir irréalisables lorsque le nombre de variable croit par contre les performances décroît rapidement lorsque le nombre de variables est très petit.

La méthode WRAPPER est parmi les meilleurs optimisateurs du critère de performance (le taux d’erreur) surtout en la couplant avec le modèle bayesien naif, ce compliment nous a poussé à l’utiliser pour l’approche aval qui est basé sur les réseaux bayésiens pour la sélection des variables. La stratégie de recherche utilisée est très simple avec l’approche d’ajouter et de retirer au fur et à mesure un descripteur à la solution courante; et à chaque fois en mesurer le taux d’erreur. A la fin, nous sélectionnons les descripteurs qui ont donnée le taux d’erreur minime.

5.3 Application et Résultats

La méthode WRAPPER a été implémentée dans le logiciel Sipina, c’est parmi les rares logiciels gratuits qui renferment cette méthode. La chose remarquée dans l’apprentissage avec cette méthode avec le modèle bayésien naïf est qu’elle prend énormément de ressources physiques pour avoir des résultats, cet inconvénient nécessite un temps énorme surtout lorsque le nombre de descripteurs est élevé. La figure 4.11 représente l’application de la méthode WRAPPER avec le modèle bayésien naïf sur la première base de données dans le logiciel Sipina.

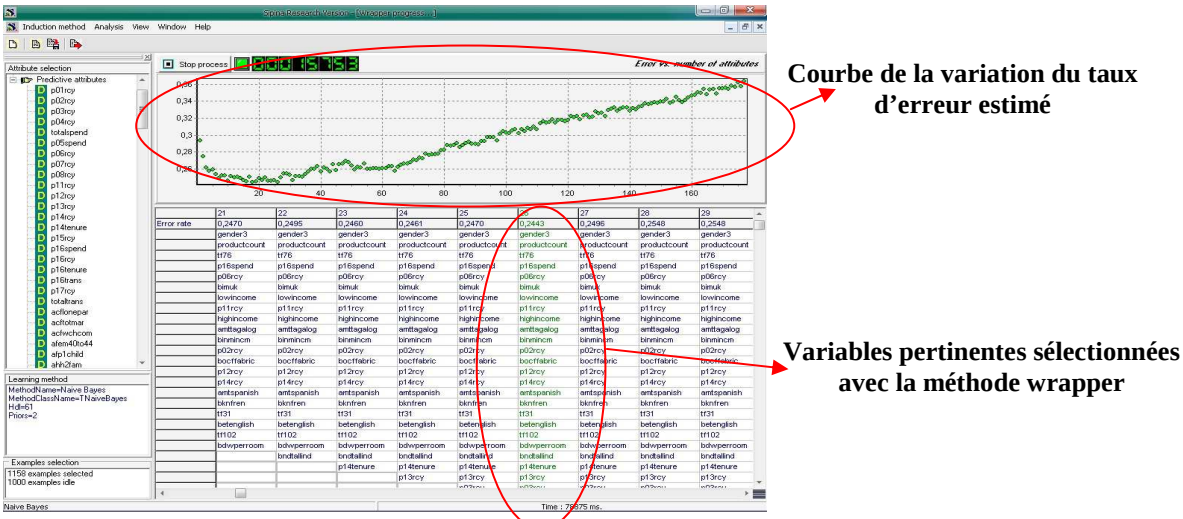

Figure 4.11 : Méthode WRAPPER couplée avec le modèle bayésien naïf

Pour une meilleure précision des résultats, l'évaluation des performances de cette méthode ainsi que l'hybridation avec les réseaux de neurones est appliquée dans le logiciel Tanagra. Les résultats d'évaluation des trois bases de données sont présentés dans les trois tableaux 4.6, 4.7 et 4.8.

La première base de données ($BDD_{Scoring}$) : Le tableau 4.6 représente les résultats obtenus en appliquant la classification hybride en aval sur la première base de données avec la méthode de sélection de variables WRAPPER.

Tableau 4.6 : Résultats de la méthode WRAPPER sur la première BDD

	Réseau bayésien				Réseau de neurones
	BDD initiale		BDD réduite		
Méthode d'évaluation	**Nombre de descripteurs**	**Taux d'erreur T_1**	**Nombre de descripteurs**	**Taux d'erreur T_2**	**Taux d'erreur Final T_F**
METHODE WRAPPER					
Apprentissage -test	**177**	0.3790	**26**	0.2960	**0.2820**
Bootstrap 0.632+		0.3400		0.2884	**0.2809**
Validation Croisée		0.3674		0.2815	**0.2816**

Le nombre de descripteurs est optimisé à 26 variables avec une amélioration de plus de 7 % du taux d'erreur T_1 avec seulement l'application du réseau bayésien couplé avec la méthode WRAPPER. Une très légère amélioration du taux T_2 est observée en appliquant l'hybridation avec les réseaux de neurones sur la base de données intégrant seulement les 26 descripteurs sélectionnés auparavant. Enfin, une amélioration moyenne en

prédiction de plus de 22 % engendre certainement un gain énorme en terme de nombre de publipostage économisé.

La deuxième base de données ($BDD_{Champignon}$) : Le tableau 4.7 représente les résultats obtenus en appliquant la classification hybride en aval sur la deuxième base de données avec la méthode de sélection de variables WRAPPER.

Tableau 4.7 : Résultats de la méthode WRAPPER sur la deuxième BDD

	Réseau bayésien				**Réseau de neurones**
	BDD initiale		**BDD réduite**		
Méthode d'évaluation	**Nombre de descripteurs**	**Taux d'erreur T_1**	**Nombre de descripteurs**	**Taux d'erreur T_2**	**Taux d'erreur Final T_F**
METHODE WRAPPER					
Apprentissage -test	**22**	0.0542	**10**	0.0118	**0.0085**
Bootstrap 0.632+		0.0442		0.0104	**0.0059**
Validation Croisée		0.0452		0.0105	**0.0079**

La première étape de la classification a diminué le nombre des descripteurs a plus de la moitié avec une amélioration concrète de plus de 4 % du taux d'erreur T_1. Encore ces résultats prouvent l'efficacité du réseau bayésien devant les données incomplètes. Une autre amélioration du taux d'erreur T_2 après l'application du réseau de neurones sur la base de données réduite est observée. Ainsi, une amélioration du taux moyen final avoisinant les 85 % par rapport au taux initial avec 12 descripteurs éliminés est considéré comme une très grande optimisation des performances de prédiction en classification des champignons pouvant sauver des vies humaines.

La troisième base de données (BDD_{Credit}) : Le tableau 4.8 représente les résultats obtenus en appliquant la classification hybride en aval sur la troisième base de données avec la méthode de sélection de variables WRAPPER.

Tableau 4.8 : Résultats de la méthode WRAPPER sur la troisième BDD

	Réseau bayésien				**Réseau de neurones**
	BDD initiale		**BDD réduite**		
Méthode d'évaluation	**Nombre de descripteurs**	**Taux d'erreur T_1**	**Nombre de descripteurs**	**Taux d'erreur T_2**	**Taux d'erreur Final T_F**
METHODE WRAPPER					
Apprentissage -test	**16**	0.2606	**9**	0.2576	**0.3000**
Bootstrap 0.632+		0.2550		0. 2454	**0.2755**
Validation Croisée		0.2620		0.2440	**0.2790**

Les résultats obtenus montrent une légère amélioration du taux d'erreur T_1 avec la sélection de 9 descripteurs. Cependant, le taux T_F s'est détérioré considérablement avec l'application du réseau de neurone sur la base de données réduite. Cette détérioration est expliquée par le comportement du réseau bayésien avec la méthode WRAPPER devant la corrélation élevée des données éliminant les descripteurs corrélés entre eux malgré que quelque uns pèsent sur la classification d'une part, et la fragilité du réseau de neurones devant les données altérées d'une autre part. Malgré ça, l'optimisation du nombre des descripteurs reste un grand avantage dans ce contexte ou le nouveau client peut s'ennuyé par un nombre élevés de questions posées ou bien de papiers demandés dans la demande de crédit bancaire ce qui l'oblige à changer la banque.

L'analyse des résultats de la démarche aval pour les trois bases de données est résumée dans le tableau 4.9 :

Tableau 4.9 : Résultats de l'hybridation avec la méthode WRAPPER sur les trois BDD

			$BDD_{Scoring}$	**$BDD_{Champignon}$**	**BDD_{Credit}**
Caractéristiques BDD		Nombre enregistrements	2 158	8 124	1 000
		Nombre descripteurs	177	22	16
		Corrélation donnée	Normale	Très Faible	Haute
		Qualité BDD	Bien remplie	Manquante	Bien remplie
Classification initiale RB		Apprentissage -test	0.3790	0.0542	0.2606
		Bootstrap 0.632+	0.3400	0.0442	0.2550
		Validation Croisée	0.3674	0.0452	0.2620
		T_{RN} moyen	**0.3621**	**0.0479**	**0.2592**
Classification hybride Aval RB → RN	**WRAPPER**	Descripteurs	**26**	**10**	**9**
		Taux d'optimisation	**85.31 %**	**54.54 %**	**43.75 %**
		Apprentissage -test	0.2820	0.0085	0.3000
		Bootstrap 0.632+	0.2809	0.0059	0.2755
		Validation Croisée	0.2816	0.0079	0.2790
		T_F moyen	**0.2815**	**0.0074**	**0.2848**
		Taux d'amélioration	**22.26 %**	**84.55 %**	**- 9.87 %**

Ce tableau représente le récapitulatif des résultats obtenus en appliquant la classification initiale avec les réseaux bayésiens et la classification hybride en aval avec la méthode de sélection de variables WRAPPER sur les trois bases de données

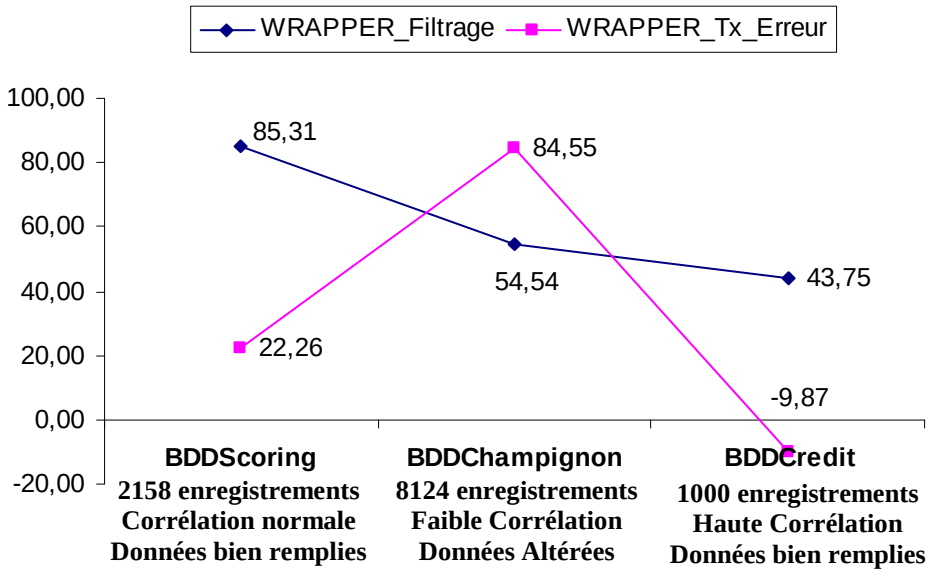

Figure 4.12 : Taux d'optimisation et taux d'amélioration du taux d'erreur avec la méthode WRAPPER sur les trois BDD

La figure 4.12, représentant les taux d'optimisation des descripteurs et les taux d'amélioration des taux d'erreur pour les trois bases de données, montre les taux élevés positifs réalisés avec les deux critères de performance. Cependant, nous constatons une légère détérioration du taux de prédiction pour la base de donnée crédit, due au comportement de la méthode WRAPPER couplée avec le modèle bayésien naïf devant la corrélation élevée des données éliminant les descripteurs dépendants entre eux d'une part, alors que quelques uns pèsent sur la classification, et la fragilité du réseau de neurones devant les données altérées d'une autre part. Mais malgré ça, l'optimisation du nombre des descripteurs reste un grand avantage dans ce contexte dans la mesure ou le gain en stabilité des données compense la perte en taux de prédiction.

6. Comparaison et interprétation

Après avoir étudier les deux variantes amont et aval chacune a part, nous passons à une comparaison des différentes variantes en combinant les résultats obtenus précédemment.

Tableau 4.10 : Récapitulatif des résultats de la démarche proposée

				$BDD_{Scoring}$	**$BDD_{Champignon}$**	**BDD_{Credit}**
Caractéristiques BDD			Nombre enregistrements	2 158	8 124	1 000
			Corrélation donnée	Normale	Très Faible	Haute
			Qualité BDD	Bien remplie	Manquante	Bien remplie
Classification initiale			T_{1RN} moyen	**0.3116**	**0.0211**	0.2942
			T_{1RB} moyen	0.3621	0.0479	**0.2592**
Classification hybride	**Amont**	**STEPDISC**	**Taux d'optimisation**	**85,88 %**	**13,64 %**	**56,25 %**
			Taux d'amélioration	**07.41 %**	**87,68 %**	**- 04,05 %**
		MOYVAR	**Taux d'optimisation**	**05,65 %**	**59,09 %**	**06,25 %**
			Taux d'amélioration	**03,95 %**	**- 334.12 %**	**1.47 %**
	Aval	**WRAPPER**	**Taux d'optimisation**	**85.31 %**	**54.54 %**	**43.75 %**
			Taux d'amélioration	**22.26 %**	**84.55 %**	**- 9.87 %**

Le tableau 4.10 représente les résultats obtenus avec la classification hybride par rapport aux meilleurs résultats de la classification initiale entre la classification avec le réseau de neurones et la classification avec le réseau bayésien naïf, ou nous remarquons que pour la première et deuxième bases de données, le réseau de neurones a donnée meilleure estimation du taux d'erreur initial que le réseau bayésien, contrairement à la troisième base de donnée ou nous observons une supériorité du réseau bayésien dans la classification avec un taux d'erreur meilleur de plus de 3.5 % par rapport au réseau de neurone due essentiellement au comportement négatif de ce dernier face au base de donnée de petit volume.

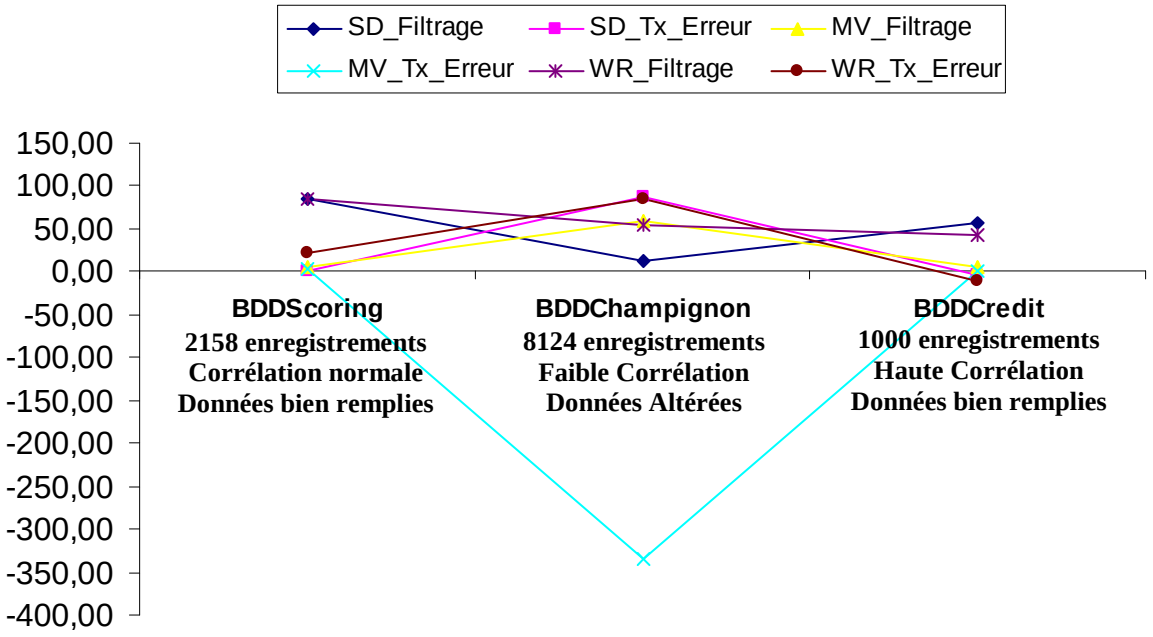

Figure 4.13 : Taux d'optimisation et taux d'amélioration du taux d'erreur avec les trois méthodes STEPDISC, MOYVAR et WRAPPER sur les trois BDD

Après hybridation, une grande variation des résultats est observée d'après la base de donnée, la méthode de sélection de descripteurs et la démarche amont ou aval appliquée représentée dans la figure 4.13 ou nous constatons globalement que :

1. Avec une base de données de volume moyen contenant des données bien remplies et une corrélation normale : l'hybridation (amont et aval) à base des réseaux de neurones et des réseaux bayésien pour les différentes méthodes de sélection de variables donne de meilleure résultats qu'une classification simple.
2. L'hybridation amont avec la méthode STEPDISC et aval avec la méthode WRAPPER montrent une meilleure optimisation du nombre de descripteurs.
3. Les données altérées influes négativement sur la méthode MOYVAR, par contre, les autres méthodes prennent leur précautions et fournissent de bonnes performances, surtout avec l'utilisation du réseau bayésien qui a un comportement positif faces à ce type de données.
4. Pour les bases de données de petit volume, et même avec une corrélation haute des données, l'hybridation amont avec la méthode MOYVAR est la meilleure à utiliser, vu son savoir faire avec ce type de base de données. Cependant, les deux autres méthodes restent applicables dans la mesure où ils fournissent un gain de stabilité meilleur compensant la petite perte en erreur de prédiction.

7. Conclusion

Nous avons implémenté dans ce chapitre une méthode hybride de classification utilisant le réseau bayésien naïf et le réseaux de neurone à perceptron multicouches, utilisant l'algorithme de rétropropagation d'erreur, qui a été appliqué pour trois base de données différentes : la première concerne la ciblage clientèle, la deuxième la classification des champignons et la dernière pour la prédiction des clients douteux. Cette méthode consiste à utiliser les deux algorithmes en amont et en aval tel que le premier sert a un optimisateur de descripteurs en collaboration avec une méthode de sélection de variables et le deuxième sert à un classifieur fin des données résultants de l'amélioration précédente.

Différentes méthodes de sélection de variables sont disponibles ou, en plus des méthodes STEPDISC et WRAPPER choisies, nous avons mis en œuvre une nouvelle méthode appelée MOYVAR pour la démarche amont. La classification hybride avec les trois méthodes est comparée sous l'angle d'amélioration du taux d'erreur et d'optimisation du nombre de descripteurs. Cependant les deux méthodes STEPDISC et WRAPPER, et malgré leur supériorité dans l'optimisation des descripteurs même avec des données manquantes, ont démontré une détérioration du taux d'erreur dans la classification pour les bases de données de petit volume surtout avec des données hautement corrélées. Par contre, la méthode MOYVAR s'applique bien lorsque la corrélation entre les données non altérées est forte et prouve son efficacité surtout avec les petites bases de données.

Pour permettre une évaluation effective et certaine de la classification hybride nous avons utilisé, en plus de la technique traditionnelle apprentissage-test, deux autres techniques de rééchantillonnage pour évaluer l'erreur de prédiction, c'est la technique de validation croisée et la technique de bootsrap 632+ censée tenir compte des spécificités de la technique d'apprentissage.

CONCLUSION

Dans ce travail, nous nous sommes placés dans un cadre plutôt défavorable ou il n'existe pas d'études similaires. Cependant, en combinant deux algorithmes relativement stables, nous avons pu proposé un modèle de classification hybride à base de réseau de neurones et de réseau bayésien. Nous avons voulu vérifier le comportement de cette hybridation en classification pour plusieurs bases de données avec des caractéristiques assez différentes. En utilisant trois méthodes de sélection de variables, les deux approches amont et aval ont données des résultats satisfaisants en terme d'amélioration du taux d'erreur et / ou de sélection de sous-ensembles de données pertinentes mettant en évidence que cette sélection permet de réduire considérablement la complexité des processus décisionnels qui a généré les données en facilitant l'interprétation des résultats de la classification et réduire le temps d'obtention des solutions, tout en garantissant de bonnes performances en classification par la réduction de l'instabilité des données de sorte que parfois l'erreur commise soit compensée par le gain de stabilité.

Notre principale contribution technique était l'idée proposée pour profiter des caractéristiques des réseaux de neurones et des réseaux bayésiens en les combinant avec les approches amont et aval. Nous avons pu offrir une nouvelle méthode de classification optimale avec le minimum de données facilitant l'analyse du data miner. Dans un autre contexte, la nouvelle méthode MOYVAR couplée avec le réseau de neurones a prouvée son efficacité dans cette classification, ceci nous donne une deuxième contribution. Toutefois, l'efficacité de cette méthode est conditionnée par quelques règles. Enfin, l'utilisation de trois techniques d'évaluation en classification a savoir : l'apprentissage-test, la validation croisée et le bootstrap donne une confiance certaines aux résultats obtenus.

Pour résumer, nous avons pu avec ce travail concevoir une stratégie de classification hybride utilisant les réseaux de neurones et les réseaux bayésiens qui a prouvée ses résultats. La suite du travail consiste en une implémentation générale destinée à la prédiction d'un nouveau cas.

Bibliographie et Références

[1] Michel Jambu : « Introduction au Data Mining », Editions Eyrolles et Fr Télécom-CENT, 1999

[2] Stéphane Tufféry : « Datamining et Statistique décisionnelle », Éditions Technip, 2007

[3] René Lefébure et Gilles Venturi : « Le Data Mining », éditions Eyrolles, Mars 2001

[4] Gilles Ballmise : « Les réseaux de neurones », rapport technique, Septembre 2002

[5] Benamar Houmadi : « Etude Exploratoire d'outils pour le Data Mining », mémoire de maîtrise en mathématiques et informatique appliquées, université de Québec à Trois-Rivières-Canada, 2007

[6] Gilles Ballmise : « Les réseaux bayésiens », rapport technique, Septembre 2002

[7] Patrick Naïm, Pierre-Henri Wuillemin, Philippe Leray, Olivier Pourret, Anna Becker : « Réseaux bayésiens », Editions Eyrolles 2007

[8] Gérard Dreyfus , Jean-Marc Martinez , Manuel Samuelides , Mirta B. Gordon , Fouad Badran , Sylvie Thiria , Laurent Hérault : « Réseaux de neurones ; Méthodologie et Applications », éditions Eyrolles, 2004

[9] AntSnio C. Roque-da-Silva-Filho : «Use of a neural field model to Derive equilibrium Values for the Weights of Recurrent Synapses», Université de Sao Paulo, Brazil, 1990

[10] E-G.Talbi : « Fouille de données (Data Mining) -Un tour d'horizon », support de cours, Laboratoire d'Informatique Fondamentale de Lille, 2007

[11] Demouche Mouloud : « Classification non linéaire par réseaux de neurones », thèse de magister, Université de Béjaia, Novembre 2005

[12] Jiawei Han and Micheline Kamber : « Data Mining: Concepts and Techniques», Morgan Kaufmann Publishers, 2000.

[13] Sylvain Barthelemy : « Les Réseaux de neurones », rapport technique, Juin 2000

[14] Léon Personnaz et Isabelle Rivals : « Réseaux de neurones formels pour la modélisation, la commande et la classification », CNRS Editions, 2003.

[15] http://www.eaa.egss.ulg.ac.be/seminaire/docs/Sem03_12_12_pmack.PDF

[16] O.Nouali : « Classification automatique des massages : une approche hybride », Conférence au RECITAL à Nancy en France, Juin 2002

[17] Ale Koerich - Yann Leydier - Robert Sabourin - Ching Y.Suen : « Système Hybride de reconnaissance de mots manuscrits sur un grand vocabulaire utilisant des Réseaux Neuronaux et des Modèles de Markov Cachés », rapport technique, 2002

[18] Philippe Leray, Patrick Gallinari : « Une architecture Neuro-Bayésienne pour le traitement spation-temporel d'Alarmes.Application au diagnostic dans le réseau téléphonique Français », rapport technique, 1999

[19] Dreyfus G., Guyon I : « Assessment Methods, in Feature Extraction, Foundations and Applications », rapport technique, 2006.

[20] Noelia Sanchez-Marono, Amparo Alonso-Betanzos, Maria Tombilla-Sanroman : « Filter Methods for Feature Selection – A Comparative Study », University of A Coruna, Department of Computer Science, Espagne 2007.

[21] Philippe Leray, Patrick Gallinari : « Report on variable selection », Neurosat project – environment and climate – science research and development, paris 1997

[22] Philippe Leray, Patrick Gallinari : « Feature Selection with Neural Networks» , spécial Issue on Analysis of Knowledge Representation in Neural Network Models, 1999

[23] Zapranis A. and Refenes A : « Principles of neural model identification, selection and adequacy», Perspectives in Neural Computing, 1999.

[24] Mikko Korpela : «Introduction to variable selection : Wrappers for feature subset selection», support de cours, September 2006

[25] Philippe Leray, Patrick Gallinari : « De l'utilisation d'OBD pour la sélection de variables » Rapport technique - INSA Rouen- Université Paris 6, Octobre 2000

[26] Stoppiglia H : « Méthodes statistiques de sélection de modèles neuronaux ; applications financières et bancaires », Thèse de Doctorat de l'Université Pierre et Marie Curie, ParisIV. Dec 1997, Disponible sur le site http://www.neurones.espci.fr.

[27] Grandvalet Y, Canu S: « Outcomes of the equivalence of adaptative ridge with the least absolute shrinkage », Neural Information Processing Systems, 1998

[28] Kittler J : « Feature Selection and Extraction » et Tzay Young, King-Sun Fu : chapitre 3 dans « Handbook of Pattern Recognition and Image Processing »,. Academic Press. 1986

[29] Goutte C : « Extracting the Relevant Decays in Time Series Modelling », Neural Networks for Signal Processing VII, Proceedings of the IEEE Workshop, 1997

[30] Moody J: « Prediction Risk and Architecture Selection for Neural Networks » et Cherkassky V, Friedman J.H, Wechsler H : « From Statistics to Neural Networks - Theory and Pattern Recognition Applications », 1994.

[31] Akaike H : « Statistical Predictor Identification » Ann. Inst. Statist. Math. 1970

[32] Moody J : « Generalization, regularization and architecture selection in non linear learning systems »,Proceedings of the first IEEE Workshop on Neural Networks for Signal Processing,1991.

[33] Larsen J, Hansen L.K : « Generalized performances of regularized neural networks models », Proceedings of the 1994 IEEE Workshop on Neural Networks for Signal Processing, 1994.

[34] Gustafson and Hajlmarsson : «21 maximum likelihood estimators for model selection», Automatica 1995

[35] LeCun Y, Denker J.S, Solla S.A : « Optimal Brain Damage ». Neural Information Processing Systems 2, 1990.

[36] Michael Schyns : « Les réseaux de neurones: principes et application à la détection financière des faillites » Facultés Universitaires Notre-Dame de la Paix, Département de Gestion, 1997

[37] http://www.jybaudot.fr/Tests/conformproport.html

[38] Lei Yu , Huan Liu : « Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution », Department of Computer Science & Engineering, Arizona State University, USA. Conference de Machine Learning, Washington, 2003.

[39] TANAGRA : un logiciel gratuit pour l'enseignement et la recherche, Ricco Rakotomalala, http://eric.univ-lyon2.fr/~ricco/tanagra/fr/tanagra.html. Dernière MAJ 2010.

[40] **Gartner Groupe**, fondée en 1979, est une firme américaine de consulting et de recherche dans le domaine de la technologie. Ayant environ 10 000 clients, elle mène des recherches, fournit des services de consultation, tient à jour différentes statistiques et maintient un service de nouvelles spécialisées.

[41] Cowan J.D, Sharp D.H : « Neural nets and artificial intelligence », Proceedings of the American Academy of Arts and Sciences, 1988.

[42] Jain A.K, Mao J, Mohiuddin M : « Neural Networks: A Tutorial », Publication dans IEEE Computer Society Press Los Alamitos, USA, March 1996

[43] G. Dreyfus : « Les réseaux de neurones, une technique opérationnelle pour le traitement des données industrielles, économiques et financières », rapport, laboratoire d'électronique - école supérieure de physique et de chimie industrielle de paris, 1997.

[44] Ricco RAKOTOMALALA : « Estimation de l'erreur de prédiction : les techniques de ré-échantionnage », Rapport technique, Laboratoire ERIC, 2006

[45] B. Efron and R.J. Tibshirani : « Cross-validation and the bootstrap: Estimating the error rate of a prediction rule ». Rapport technique, Standford Université, MAY 1995.

[46] A. Molinaro, R. Simon, R. Pfeiffer : « Prediction error estimation: a comparison of resampling methods », Rapport technique, laboratoires de recherche en Bioinformatique, USA, publication Oxford université 2005.
http://bioinformatics.oxfordjournals.org/cgi/content/full/21/15/3301

[47] S. Merler and C. Furlanello : « Selection of tree-based classifiers with the bootstrap 632+ rule ». publication dans la revue "Biometrical Journal", 1997.

[48] C.Furlanello 1 ; S.Merler 1,2 ; C.Chemini 2 and A.Rizzoli 2 : « An Application of the Bootstrap 632+ Rule to Ecological Data », Rapport technique, 2009
1 Institut Scientifique et Technologique Ricerca, Trento, Espagne
2 Centre d'ecologie Alpina, Trento, Espagne

[49] Wenyu Jiang2, Richard Simon1: « A comparison of bootstrap methods and an adjusted bootstrap approach for estimating the prediction error in microarray classification», Rapport technique, 2008

1 Biometric Research Branch, National Institutes of Health, Rockville, U.S.A.

2 Department of Mathematics and Statistics, Concordia University, Quebec, Canada

[50] Ricco Rakotomalala

- Maître de Conférences depuis 1998, enseignant depuis 1995 au département informatique et statistique de l'université de Lyon 2
- PhD Computer Science (Machine Learning), University Claude Bernard Lyon 1
- DEA Modélisation et Analyse Quantitative, University Nanterre Paris 10
- Maîtrise Econométrie, University Lumière Lyon 2
- Licence Sciences Economiques, University of Antananarivo (Madagascar)
- Membre du laboratoire ERIC - France
- Web : http://eric.univ-lyon2.fr/~ricco/ricco.html

[51] http://www.jybaudot.fr/Stats/inertie.html

[52] Lounis Hakim : « Apprentissage Bayésien, Notes de cours » Séminaire sur l'apprentissage automatique du programme de Doctorat en informatique cognitive, Université du Québec à Montréal, 2006.

[53] Mitchell Tom : « Bayesian Learning » Chapter 6 of Machine Learning, 1997.

[54] Benoit Lavoie : « Apprentissage Bayésien, Synthèse de lectures » Séminaire sur l'apprentissage automatique du programme de Doctorat en informatique cognitive, Université du Québec à Montréal, Avril 2006.

[55] http://genomics.energy.gov/

[56] http://www.si.fr.atosorigin.com/datawarehouse/

[57] http://www.csc.kth.se/~orre/snns-manual/UserManual/node149.html

[58] http://www.neurones.espci.fr/Theses_PS/Stoppiglia_H/chap5.pdf

[59] SIPINA : un logiciel gratuit pour l'induction des arbres de décision, Ricco Rakotomalala, http://eric.univ-lyon2.fr/~ricco/sipina.html. Dernière MAJ 2010.

[60] Ricco Rakotomalala : « Didacticiel : Comprendre le modèle d'indépendance conditionnelle (Classifieur Bayesien Naïf) », rapport technique, Mars 2010.