UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

SÉLECTION D'ATTRIBUTS PAR DIMENSION FRACTALE

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR

ABDÉLILAH BALAMANE

DÉCEMBRE 2007

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS
Département d'informatique et d'ingénierie

Ce mémoire intitulé

SÉLECTION D'ATTRIBUTS PAR DIMENSION FRACTALE

Présenté par
Abdélilah Balamane
pour l'obtention du grade de maître ès science (M.Sc.)

a été évalué par un jury composé des
personnes suivantes :

Dr. Rokia Missaoui ..... Directrice de recherche
Dr. Karim Guemhioui ..... Président du jury
Dr. Marek Zaremba ..... Membre du jury

Mémoire accepté le : 04 Décembre 2007

À la mémoire de mon épouse
À mes enfants Saad et Ali

Au professeur Rokia Missaoui pour son aide précieuse à l'élaboration de ce mémoire
qu'elle trouve dans ce travail l'expression de ma profonde gratitude
et à mon ami le professeur Boukendour Said.

Table des matières

Table des figures

Liste des tableaux

LISTE DES ABRÉVIATIONS

ACP	Analyse en composantes principales.
ACI	Analyse en composantes indépendantes.
ALG	Algorithme génétique.
SOM	*Self Organizing Map.*
SVM	Séparateurs à vaste marge.
LLE	*Locally Linear Embedding.*
RDD	Réduction de la dimensionnalité des données.
HDDA	Analyse discriminante de haute dimension.
ISOMAP	*Isometric Feature Mapping.*
ALGRST	Algorithme basé sur la méthode RST.
ALGSVM	Algorithme basé sur la méthode des SVM.
ALGFAL	Algorithme de sélection par la méthode de Faloustos.
ALGSENT	Algorithme de sélection version entropie.
ALGSDF1	Algorithme de sélection version dimension fractale (1).
ALGSDF2	Algorithme de sélection version dimension fractale (2).
ALGSGINI	Algorithme de sélection version coefficient de Gini.

Résumé

La réduction de la dimensionnalité des données (RDD) consiste à retenir les variables les plus représentatives des données observées. Elle peut être utile comme étape préliminaire à tout processus d'analyse et de traitement de données afin de se concentrer sur les variables les plus importantes et réduire le coût d'exécution d'un tel processus. Deux approches principales sont utilisées pour la RDD : l'approche par extraction (i.e., création de nouvelles variables par combinaison de celles existantes) et celle basée sur la sélection d'attributs pertinents.

Après une analyse critique et comparative d'un certain nombre de méthodes de RDD connues dans la littérature et la mise en relief de leurs forces et limites, nous exposons notre propre méthode fondée sur la sélection d'attributs pertinents et inspirée de la théorie de la dimension fractale. L'algorithme à la base de notre méthode permet non seulement de réduire la dimensionnalité des données mais également de détecter diverses corrélations dans les données observées. En outre, il permet de trouver avec une grande probabilité la réduction optimale évaluée par le rapport entre la taille de la réduction et la perte d'information liée à cette réduction. À l'opposé de la plupart des algorithmes de RDD lesquels ont une complexité algorithmique quadratique par rapport au nombre d'observations, notre procédure s'exécute en un temps quadratique par rapport au nombre de variables observées.

Finalement, nous avons validé notre approche en comparant sa capacité à produire des réductions relatives (i.e., un ensemble réduit de variables en présence d'une variable de classification) ou absolues avec celle d'autres méthodes de RDD sur une dizaine de bases de données provenant majoritairement du répertoire UCI *(Machine learning repository)*.

Chapitre 1

Introduction

Les scientifiques sont constamment confrontés à des données à très haute dimensionnalité. La recherche médicale, la finance, la vision par ordinateur, les patrons climatiques globaux et la bioinformatique en sont des exemples. L'analyse de ces données complexes nécessite l'utilisation d'outils spécialisés. Souvent, on fera appel à des techniques d'analyse statistique ou de fouille des données (*data mining*). La fouille des données a pour objet l'extraction d'un savoir ou d'une connaissance basée sur la détection des liens, motifs et regroupements ou règles d'association dans de grands volumes de données. Cependant, la performance de ces techniques dépend de la qualité et du volume des données analysées. La présence de variables redondantes et superflues complique l'analyse et engendre des résultats peu satisfaisants. Écarter de telles variables va permettre d'améliorer les temps d'exécution et possiblement les résultats, comme cela sera illustré plus tard dans ce document.

L'objectif de ce mémoire est d'élaborer une approche permettant de trouver des structures pertinentes de plus faible dimensionnalité, cachées au sein des observations dont nous disposons et qui seront capables de représenter fidèlement les données observées. Pour extraire de telles structures, on fera appel à des techniques de réduction de la dimensionnalité des données (RDD).

Cependant, réduire la dimensionnalité des données peut engendrer plusieurs difficultés dues essentiellement au lien étroit qui existe entre l'approche qu'on va adopter pour réduire la dimensionnalité et la nature des données à réduire. La revue de la littérature dans le domaine de la RDD fait état de plusieurs travaux. Cependant, plusieurs méthodes présentent une ou plusieurs lacunes : incapacité de déceler les relations de type non linéaire qui peuvent exister entre les variables représentant les données, complexité algorithmique élevée, surcharge due à plusieurs réductions obtenues et au travail supplémentaire nécessaire pour le choix de la meilleure réduction pour les données analysées et défaillance de certaines méthodes pourtant réputées performantes face à des données où le nombre d'observations est largement inférieur au nombre de variables observées (problème de la malédiction de la dimensionnalité). Enfin, la méthode de réduction peut utiliser un ou plusieurs paramètres dont le choix dépend de la nature des données étudiées. Donc, une analyse préalable de ces données est nécessaire afin de connaître le ou les paramètres à adopter.

Dans ce mémoire, nous proposons d'étudier le problème de la réduction de la dimensionnalité des données avec application à la fouille des données et particulièrement à la classification. Précisons que la classification des données est un processus qui permet de placer un objet dans la classe la plus appropriée. Plusieurs techniques basées sur les mathématiques, statistiques ou d'intelligence artificielle peuvent être employées pour faire de la classification. L'une de ces techniques se base sur la construction des arbres de décision.

Nous allons d'abord analyser la problématique qui nous conduit à procéder à cette réduction et les difficultés inhérentes à la démarche. Nous poursuivons notre étude par une comparaison des différentes techniques présentées et nous proposons un algorithme efficace de réduction de la dimensionnalité des données basé sur la dimension fractale, capable de remédier à certaines de ces carences. Sa complexité algorithmique est quadratique par rapport au nombre de variables observées et non par rapport au nombre d'observations comme c'est le cas pour la plupart des méthodes de réduction. Cependant, et afin de rendre plus clairs certaines notions et concepts qui seront utilisés dans l'élaboration de cet algorithme, le lecteur pourra consulter en annexe les quelques définitions et notions élémentaires de topologie que nous avons jugé pertinentes.

Chapitre 2

Réduction de la dimensionnalité des données

2.1 Contexte

Plusieurs situations peuvent nous conduire à faire une réduction de la dimensionnalité des données. À titre d'exemple, nous pouvons citer : l'espace de recherche trop grand dans les tableaux statistiques multidimensionnels (cubes de données), la complexité des algorithmes qui augmente avec la taille de l'espace et lors de l'emploi des méthodes de régression multiple. Notons que les méthodes de régression sont fréquemment utilisées par la communauté scientifique comme moyen d'analyse de données. Cependant, on a souvent besoin de procéder à une réduction de la dimensionnalité des données avant de pouvoir les appliquer efficacement. Le chapitre 2 de ce mémoire sera consacré à l'étude des méthodes de régression et des approches de réduction de la dimensionnalité généralement employées avec ces méthodes. Cependant, réduire la dimensionnalité des données consiste à développer une approche capable de déterminer parmi l'ensemble des variables observées notées n celles réellement nécessaires notées m pour préserver l'information pertinente, la mettre en évidence en la dissociant du bruit et possiblement révéler une structure sous-jacente qui ne serait pas immédiatement apparente dans les données d'origine en haute dimension. La variable n est appelée la dimension d'observation des données et m la dimension intrinsèque. Notons que la dimension intrinsèque correspond

au nombre de variables nécessaires et suffisantes pour représenter efficacement les données. L'exemple classique d'une telle approche est l'algorithme d'analyse en composantes principales (ACP) [10].

Différentes méthodes de diverses disciplines ont été utilisées pour la réduction de la dimensionnalité des données. L'objet de cette section est de passer en revue les deux principales approches utilisées pour la réduction de la dimensionnalité des données, à savoir : les méthodes d'extraction qui créent de nouvelles variables à partir des anciennes et les méthodes de sélection qui cherchent seulement un sous-ensemble optimal d'attributs pour un critère donné. Nous développerons davantage les méthodes de sélection d'attributs pertinents pour la réduction de la dimensionnalité qui font l'objet de notre recherche.

2.2 Méthodes d'extraction

Une méthode d'extraction consiste en la recherche de m paramètres en fonction des n paramètres initiaux ($m << n$) . Ces m paramètres seront en général calculés à partir de combinaisons linéaires des n paramètres initiaux. Les points sont alors projetés dans un sous-espace R^m. Cependant, Le nombre de paramètres à calculer pour caractériser chaque observation sera toujours n mais l'interprétation des observations se fera alors dans ce sous-espace R^m. Par ailleurs, le sens des paramètres peut être perdu. Ce type de méthodes regroupe l'analyse factorielle en statistiques, les réseaux de neurones en intelligence artificielle et les méthodes algébriques en mathématiques. À titre d'exemples, on peut citer : PCA [10] *(Principal component analysis)*, ICA [21] *(Independent component analysis)*, SOM [13] *(Self-organizing maps)*, HDDA [4] *(High dimensional discriminant analysis)*, SVM [9] *(Support vector machine)*, LLE [39] *(Locally linear embedding)* et ISOMAP [45] *(Isometric mapping of data manifolds)*. Notons cependant que toutes ces méthodes ne permettent pas pour autant de réduire le nombre de variables à considérer pour chaque observation à la différence des méthodes de sélection que nous allons présenter.

2.3 Méthodes de sélection

Le problème de la sélection d'attributs pertinents est connu depuis les années 70. C'est un domaine très actif depuis quelques années, en particulier dans le cadre de la fouille de données, l'analyse des données, la reconnaissance des formes et le traitement de données complexes (ex. le multimédia). La sélection d'attributs consiste à réduire le nombre de variables en ne sélectionnant que les plus pertinentes. Cette sélection opère par le biais soit de l'élimination d'attributs indépendants de la variable (ou des variables) de classification(s) ou par l'élimination d'attributs redondants. La sélection d'attributs pertinents offre plusieurs avantages : réduire le temps d'extraction des données, faire baisser la complexité temporelle et spatiale (temps d'exécution et espace mémoire occupé par l'algorithme) et augmenter la performance des algorithmes d'analyse. En effet, la sélection d'attributs joue un rôle important dans la fouille des données, en particulier dans la préparation des données avant leur traitement.

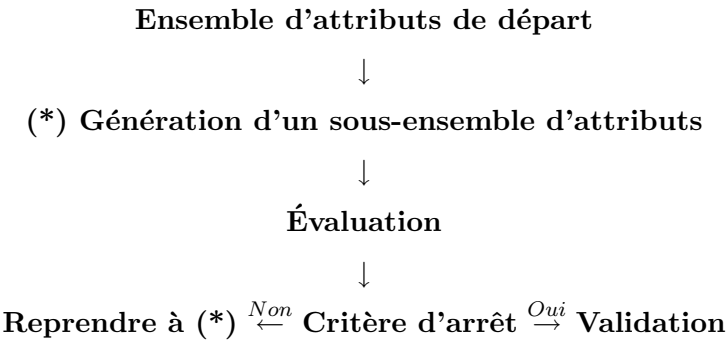

Figure 1. Procédure générale de sélection d'attributs pertinents

La figure 1 décrit la procédure générale de sélection d'attributs pertinents. Cependant, la génération de sous-ensembles est une procédure de recherche dans l'espace des sous-ensembles de cardinal 2^n, où n est le nombre d'attributs initiaux. Donc, il est nécessaire de faire appel à une stratégie efficace de recherche d'attributs. Plusieurs méthodes de parcours sont utilisables. Nous limitons notre étude aux trois principales familles : le parcours exhaustif, le parcours heuristique et le parcours non déterministe.

1. **Parcours exhaustif :** On génère et on teste tous les sous-ensembles. Ceci est prohibitif dès que le nombre d'attributs est supérieur à 10. Un tel parcours nous garantirait l'optimalité de la solution, mais le coût est excessif : $O(2^n)$.
2. **Parcours heuristique :** On utilise une heuristique pour guider la recherche et le coût devient $O(n^2)$. Dans cette catégorie, on peut commencer par un ensemble vide et ajouter les attributs un à un (*forward addition*), ou bien commencer avec l'ensemble de tous les attributs et supprimer certains (*backward elimination*), suivant un critère qu'on aurait défini et qui affecte un poids à chaque attribut. Néanmoins, une étude [19] montre que l'approche par ajout sélectionne moins d'attributs et est plus performante que l'approche par retrait.
3. **Parcours non déterministe :** Les sous-ensembles sont générés suivant un processus basé sur les algorithmes évolutionnistes (*Evolutionary Algorithms*) dont les principes de l'évolution constituent la clé de leur fonctionnement.

Nous avons décrit ci-dessus les principaux types de parcours utilisés dans l'étape 1 de la procédure générale de la sélection d'attributs pertinents. Les sous-ensembles générés à cette étape devront être évalués. Il existe pour cela deux grandes classes d'algorithmes : les algorithmes basés sur des méthodes enveloppantes (*wrapper*) [15] et les algorithmes basés sur des approches filtrantes (*filter*) [15].

Méthodes enveloppantes : Introduites par John, Kohavi et Pfleger en 1994 [15], ces méthodes ont pour principe de générer des sous-ensembles candidats et de les évaluer grâce à un algorithme de classification. Ces méthodes génèrent des sous-ensembles bien adaptés à l'algorithme de classification. Un autre avantage pour ces types d'algorithmes est leur simplicité conceptuelle. Cependant, ce type de méthodes ne constitue pas une solution parfaite car la procédure de sélection est spécifique à un algorithme de classification particulier. De plus, il n y'a pas de justification théorique à la sélection et les calculs deviennent fastidieux, voire irréalisables lorsque le nombre d'attributs croît (entre autre par l'appel à l'algorithme de classification à chaque évaluation).

L'approche filtrante : L'approche filtrante repose sur l'idée d'attribuer un score à chaque sous-ensemble. Aussi, le sous-ensemble avec le plus grand score représente le sous-ensemble d'attributs pertinents. Pour cela, plusieurs solutions sont possibles. Une première solution consiste à donner un score à chaque attribut indépendamment des autres, et faire la somme des scores. Dans le cas d'un problème de classification, on peut retenir le coefficient de corrélation comme le score de l'attribut avec la classe. Cette approche nommée *feature ranking* pose des problèmes dans le cas général car elle n'élimine pas les attributs redondants. De plus, il est possible qu'un attribut peu corrélé avec la classe devient utile lorsqu'on le considère dans le contexte des autres attributs. Une autre solution consiste à évaluer le sous-ensemble dans sa globalité (*subset ranking*).

Ghiselli [14] propose une idée intermédiaire entre *feature ranking* et *subset ranking* où le score d'un sous-ensemble est construit en fonction des corrélations attribut-classe et des corrélations attribut-attribut, selon la formule suivante :

$$R_{s\Theta} = K \times R_{\theta i} / \sqrt{K + K(K-1) \times R_{ij}}$$

$R_{s\theta}$: Score du sous-ensemble de cardinal K.
$R_{\theta i}$: Moyenne arithmétique des corrélations entre la classe θ et les attributs a_i.
R_{ij} : La moyenne des K^2 inter corrélations entre attributs.

Cette équation exprime que le score d'un sous-ensemble augmente si les attributs sont fortement corrélés avec la classe θ, et diminue s'ils sont fortement corrélés entre eux.

Chapitre 3

Régression linéaire simple et multiple

3.1 Contexte

L'analyse de la régression peut être définie comme la recherche de la relation stochastique qui lie deux ou plusieurs variables. On dit de cette relation qu'elle est non déterministe car elle n'est généralement pas exacte. Le champ d'application de la régression recouvre plusieurs domaines. Dans ce chapitre, nous étudierons uniquement la régression multiple pour laquelle la régression simple en est un cas particulier. Notons que dans tout modèle de régression il existe deux types de variables : les variables explicatives et la variable expliquée. Cependant, lorsqu' on cherche à établir un modèle de régression on se trouve souvent face à un vaste ensemble de données composé de multiples variables explicatives dont les effets sur la variable expliquée sont difficiles à quantifier et à interpréter, d'où la nécessité de supprimer certaines variables superflues.

3.2 Fondements théoriques

3.2.1 Régression linéaire multiple

La régression multiple [41] modélise une relation entre plusieurs variables $X_1, X_2, ..., X_p$ et Y. $X_1, X_2, ..., X_p$ étant les variables explicatives et Y la variable expliquée. La régres-

sion multiple cherche à étudier la relation qui lie la variable expliquée aux variables explicatives. Notons que la régression simple est un cas particulier de la régression multiple où il ya une seule variable explicative et une variable expliquée. C'est le cas par exemple où on a une variable X représentant l'âge d'une personne et Y son poids. Pour cet exemple, étudier la relation qui lie X à Y revient à constituer un échantillon composé de plusieurs mesures prélevées sur X et Y et l'observer dans un graphe à deux dimensions afin de déterminer la nature de la relation liant ces deux variables. Dans le cas où cette relation est exacte, alors il existe une fonction f connue tel que $Y = f(X)$ et si en plus elle est linéaire alors, $f(X) = aX + b$, avec a et b deux nombres réels fixés. Il arrive aussi que la relation soit exactement linéaire sans pour autant connaître les coefficients a et b. Il s'agit alors de déterminer ces coefficients à partir d'un échantillon $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. Toutefois, dans la plupart des cas, le modèle linéaire entre les variables explicatives et la variable expliquée n'est pas exact. À un ensemble de deux p-uplets $(x_{i1}, x_{i2}, ..., x_{ip})$ et $(x_{j1}, x_{j2}, ..., x_{jp})$ identiques peuvent correspondre deux valeurs y_i et y_j différentes. Comme c'est le cas de la relation entre l'âge d'une personne et son poids. Deux personnes d'un même âge n'ont pas nécessairement le même poids. Alors, une façon de décrire cette relation dans le cas général de la régression multiple en supposant qu'elle soit de nature linéaire est : $Y = a_0 + a_1X_1 + a_2X_2 + ... + a_pX_p + \epsilon$. Cette équation schématise le modèle d'une relation linéaire qui existe entre les variables explicatives $X_1, X_2, ..., X_p$ et la variable expliquée Y. Notons que dans cette relation la variable ϵ représente le comportement individuel permettant d'ajuster le modèle. Dans certains cas, l'équation du modèle de la régression multiple peut être formulée sous la forme : $\mu_y(x_1, x_2, ..., x_p) = a_0 + a_1X_1 + a_2X_2 + ... + a_pX_p$, où $\mu_y(x_1, x_2, ..., x_p)$ représente la moyenne de toutes les valeurs de la variable expliquée pour laquelle la valeur des variables explicatives valent $(x_1, x_2, ..., x_p)$. Dans ce cas, le problème de la régression multiple consiste à estimer les $p + 1$ paramètres $a_0, a_1, ..., a_p$ à partir d'un échantillon prélevé sur l'ensemble des objets utilisés pour bâtir le modèle de la régression. On note $\widehat{a}_0, \widehat{a}_1, ..., \widehat{a}_p$ les valeurs estimées des paramètres $a_0, a_1, ..., a_p$ et $\widehat{y}(x_1, x_2, ..., x_p)$ une estimation de $\mu_y(x_1, x_2, ..., x_p)$. Donc, $\widehat{y}(x_1, x_2, ..., x_p) = \widehat{a}_0 + \widehat{a}_1X_1 + \widehat{a}_2X_2 + ... + \widehat{a}_pX_p$.

3.2.2 L'hyperplan de la régression

Dans le cas d'un modèle de régression simple de type linéaire, la relation liant la variable explicative X à la variable expliquée Y est donnée par $Y = aX + b$. Cette relation symbolise l'équation de ce qu'on appelle une droite de régression. Cependant, dans le cadre de la régression multiple de type linéaire, l'équation symbolisant le modèle est $Y = a_0 + a_1X_1 + a_2X_2 + ... + a_pX_p$. Dans ce cas, on parle d'hyperplan de régression. Le problème de la modélisation revient donc à trouver les paramètres $a_0, a_1, ..., a_p$ de l'hyperplan. Soit y_i la ième valeur observée de la variable Y lors d'un processus d'échantillonnage et $\widehat{y_i}$ la valeur estimée pour la même observation à l'aide de l'équation du modèle. Alors, $y_i = a_0 + a_1X_{i1} + a_2X_{i2} + ... + a_pX_{ip}$ et $\widehat{y_i} = \widehat{a}_0 + \widehat{a}_1X_i1 + \widehat{a}_2X_i2 + ... + \widehat{a}_pX_ip$. Les quantités $e_i = y_i - \widehat{y_i}$ sont appelées les résidus du modèle.

Soit $f(a_0, a_1, ..., a_p) = \sum e_i^2 = \sum(y_i - \widehat{y_i})^2$, le calcul de ces sommes se fait sur l'ensemble des observations utilisées pour évaluer les paramètres du modèle. Notons que la plupart des méthodes d'estimation vont chercher les paramètres $a_0, a_1, ..., a_p$ qui minimisent la fonction f. La plus connue est la méthode des moindres carrés.

3.2.3 La méthode des moindres carrés

Pour estimer les paramètres du modèle et dans un but de simplification et de clarté de la démarche présentée, les calculs vont être faits dans le cadre de la régression simple. Le passage au cas général de la régression multiple se fait en utilisant la théorie du calcul matriciel. Dans ce cadre et pour toute observation i, on a la relation : $y_i = a_0 + a_1x_i$ et $\widehat{y_i} = \widehat{a}_0 + \widehat{a}_1X_i$. Il s'agit alors d'estimer les paramètres $\widehat{a}_0, \widehat{a}_1$ de a_0, a_1 tels que la quantité $\sum e_i^2 = \sum(y_i - \widehat{y_i})^2$ soit minimale. On doit donc calculer les dérivées partielles de la fonction $f(a_0, a_1) = \sum(y_i - a_0 - a_1x_i)^2$ par rapport à a_0 et a_1 telle que $\frac{\delta f}{\delta a_0} = -2\sum(y_i - a_0 - a_1x_i)$ et $\frac{\delta f}{\delta a_1} = -2\sum x_i(y_i - a_0 - a_1x_i)$. Les paramètres a_0 et a_1 sont obtenus en annulant ces dérivées partielles. Ce qui donne après transformations algébriques et en utilisant les notations $\overline{x} = \frac{\sum x_i}{n}$ et $\overline{y} = \frac{\sum y_i}{n}$,
$a_0 = \overline{y} - \widehat{a}_1\overline{x}$ et $a_1 = \frac{\sum x_iy_i - n\overline{xy}}{\sum x_i^2 - n\overline{x}^2}$.

3.3 Choix du modèle

Nous avons vu que le but d'un modèle de régression est de décrire un phénomène à l'aide d'une ou plusieurs variables explicatives. Cependant, lorsque plusieurs de ces variables sont fortement corrélées avec un coefficient de corrélation $\geq 70\%$, l'estimation des paramètres du modèle devient difficile voir imprécise. Il est alors impératif de déceler l'ensemble des variables corrélées et de les écarter du modèle. D'où, la nécessité de faire une réduction de la dimensionnalité des données. Plusieurs approches peuvent être adoptées [40, 46]. Nous allons présenter quatre critères parmi les plus utilisés pour le choix du modèle parcimonieux, c-à-d le modèle avec le minimum de variables explicatives et le plus pertinent pour représenter les données.

3.3.1 Critère du R^2 et du R^2_{aj} ajusté

Nous avons mentionné lors de la présentation des modèles de régression qu'une partie de la variance de la variable Y peut être expliquée par la variance des variables explicatives $(X_1, X_2, ..., X_p)$. Il s'agit de la variance expliquée par le modèle. Cependant, on peut avoir une série d'observations identiques générant des valeurs différentes pour la variable expliquée Y. Il s'agit de la variance inexpliquée par le modèle. L'erreur commise peut être représentée par la relation :

$$(y_i - \overline{y}) = (\widehat{y}_i - \overline{y}) + (y_i - \widehat{y}_i), \text{ où}$$

$(y_i - \overline{y})$: erreur totale sur Y
$(\widehat{y}_i - \overline{y})$: erreur expliquée par le modèle
$(y_i - \widehat{y}_i)$: erreur inexpliquée par le modèle.

Si on note $SC_{tot} = \sum(y_i - \overline{y})^2$, $SC_{reg} = \sum(\widehat{y}_i - \overline{y})^2$ et $SC_{res} = \sum(y_i - \widehat{y}_i)^2$, alors on peut montrer que $SC_{tot} = SC_{reg} + SC_{res}$. Le coefficient de détermination $R^2 = \frac{SC_{reg}}{SC_{tot}}$ prend ses valeurs dans $[0, 1]$ et permet d'évaluer le degré d'adéquation du modèle. Une valeur proche de 1 indique que le modèle est parfaitement adéquat alors qu'une valeur proche de 0 correspond au cas où le modèle est inadéquat. Ce critère pourra donc être utilisé afin d'évaluer la pertinence d'une variable. Pour comparer deux sous-ensembles

ayant le même nombre de variables explicatives, on peut ainsi comparer les R^2 obtenus après les deux analyses de régression et choisir le sous-ensemble pour lequel le R^2 est le plus grand. Cependant, il a été montré [23] que le coefficient R^2 croît systématiquement quand une nouvelle variable explicative est rajoutée au sous-ensemble, que cette dernière soit pertinente ou non pour la variable expliquée. Pour contourner cette difficulté, un nouveau critère a été créé, c'est le R^2 ajusté : $R^2_{aj} = 1 - \frac{SC_{res}}{SCtot} \cdot \frac{n-1}{n-p}$, où n représente le nombre des variables explicatives et p le nombre de celles qui sont sélectionnées parmi les n variables. Le R^2_{aj} n'augmente pas forcément lors de l'introduction de variables supplémentaires dans le modèle. Donc, il est possible d'utiliser ce critère pour comparer deux sous-ensembles de variables n'ayant pas nécessairement le même nombre de variables explicatives.

3.3.2 Critère du C_p de Mallows

Le critère C_p de Mallows [26] est basé sur l'erreur quadratique moyenne (EQM) qui est-elle même basée sur la notion d'espérance mathématique. On définit l'espérance mathématique d'une variable X par l'expression $E(X) = \sum xP(x)$, avec x une valeur quelconque de la variable X et $P(x)$ la probabilité qui lui est associée.

L'idée derrière l'utilisation du critère C_p est le choix du modèle qui minimiserait cette erreur. Dans le cadre de la régression multiple, si le modèle est correct, alors $E(\widehat{y_i}) = E(y_i)$ où les y_i sont les observations relatives à la variable expliquée et les $\widehat{y_i}$ sont les valeurs estimées par le modèle.

$$\text{On a, } EQM(\widehat{y_i}) = [E(\widehat{y_i}) - E(y_i)]^2 + V(\widehat{y_i}).$$

Dans cette expression, $V(\widehat{y_i}) = E(\widehat{y_i}^2)$ - $[E(\widehat{y_i})]^2$ représente la variance de la variable $\widehat{y_i}$. Donc, la somme des erreurs quadratiques moyenne est donnée par :

$$E(SC_{res}) \text{ - } \sigma^2(n - 2p).$$

Dans cette expression, n représente le nombre total de variables explicatives et p celles constituant le modèle. Le modèle retenu sera celui qui minimiserait cette somme. Toutefois, la variance σ^2 est inconnue. On pourra alors utiliser une valeur estimée pour ce

paramètre $\widehat{\sigma^2} = \frac{SC_{res}(x_1,x_2,...,x_p)}{n-p}$, ce qui donne $C_p = \frac{SC_{res}}{\widehat{\sigma^2}}$ - $n + 2p$ qu'on peut aussi exprimer sous la forme $C_p = (n-p).\ \frac{\widehat{\sigma_p}^2}{\widehat{\sigma_n}^2} + 2p - n$.

Les éléments $\widehat{\sigma_p}^2$ et $\widehat{\sigma_n}^2$ représentent respectivement la variance estimée pour les p variables sélectionnées par le modèle et la variance estimée pour la population de référence. Le modèle retenu sera celui qui contient le minimum de variables explicatives p et dont la valeur de C_p est très voisine de p. Notons que si $p = n$, alors $C_p = p$.

3.3.3 Critère AIC

Le dernier critère présenté est le AIC (*Akaike information criterion*). En régression multiple, la répartition des erreurs est supposée suivre une loi multinormale. Alors, la densité des y_i est donnée par l'équation $f_i(y) = \frac{1}{\sqrt{2\pi\sigma^2}}.EXP(-\frac{1}{2}.\frac{(y-a_0-a1x_{i1}-...-apx_{ip})^2}{\sigma^2})$ et la vraisemblance L par le produit de ces densités. Généralement, on parle de vraisemblance associée à une famille de lois de probabilité $P\{\alpha\}$ définies sur un ensemble $\{x_1, x_2, ..., x_n\}$ et donnée par l'expression $L(x_1, x_2, ..., x_n, \alpha) = \prod P_\alpha(x_i)$. Cette expression exprime la probabilité que l'échantillon théorique $\{X_1, X_2, ..., X_n\}$ ait pour réalisation $\{x_1, x_2, ..., x_n\}$, c-à-d que la loi de probabilité P_α ajuste correctement la répartition de l'échantillon $\{x_1, x_2, ..., x_n\}$. Comme avec le critère R^2, la vraisemblance augmente systématiquement lors de l'introduction de nouvelles variables dans le modèle. En 1971, le staticien japonais Hirotugu Akaike [1, 5] a mis au point un critère nommé AIC basé sur le maximum de vraisemblance pour choisir le modèle parcimonieux. Il a introduit le paramètre p dans l'expression de ce critère afin de pénaliser les sous-ensembles contenant le plus de variables explicatives, soit $AIC(p) = -2.Ln(L) + 2p$. Dans cette expression, p représente le nombre de variables du modèle et L la vraisemblance maximisée pour ce modèle. Le modèle parcimonieux pour ce critère est celui qui minimise la valeur de AIC(p). Toutefois, dans le cadre de la régression multiple et sous l'hypothèse que les erreurs de régression suivent une loi multinormale, $AIC(p) = \frac{SC_{res}}{\sigma^2} + 2p$. Dans cette expression, le terme $\frac{SC_{res}}{\sigma^2}$ correspond à la vraisemblance maximisée du modèle sélectionné. Sa valeur diminue à chaque fois qu'une variable est ajoutée au modèle. Afin d'éviter de choisir le modèle complet comme modèle de sélection finale, le terme $2p$ est introduit dans l'expression de AIC(p) pour tenir compte de la taille du modèle. Le modèle retenu

sera celui qui génère le AIC(p) le plus faible.

3.4 Principales limites

Nous avons présenté dans ce chapitre quatre critères fréquemment utilisés pour la sélection des modèles parcimonieux. Afin de maximiser le rendement de ces critères et pouvoir obtenir le modèle quasi-équivalent au modèle complet, nous devons examiner avec l'aide de l'un de ces critères, les 2^n modèles possibles, où n représente le nombre total de variables explicatives. Cela peut s'avérer très coûteux en temps de calcul [28].

Chapitre 4

État de l'art

Tel qu'indiqué précédemment, on regroupe sous le thème RDD des méthodes visant à résumer l'information présente dans un espace de haute dimension sur un espace de dimension plus petite, avec la contrainte de préserver l'information pertinente et de la mettre en évidence. On fera ici la distinction entre les méthodes d'extraction et les méthodes de sélection. L'objet de ce paragraphe est de donner un aperçu des caractéristiques essentielles des méthodes d'extraction et de passer en revue les principales méthodes de sélection, objet de notre recherche.

On rappelle que différentes méthodes dans différentes disciplines ont été élaborées pour la réduction de la dimensionnalité des données. Sous le thème méthodes d'extraction nous pouvons citer : les méthodes fondées sur les statistiques factorielles, celles basées sur les réseaux de neurones et les méthodes algébriques. Viennent ensuite les méthodes de sélection d'attributs pertinents.

4.1 Méthodes de statistique factorielle

Différentes méthodes de statistique factorielle ont été utilisées pour réduire la dimensionnalité des données. Dans cette catégorie, la réduction du nombre d'attributs ne se fait pas par une simple sélection de certains d'entre eux mais par la construction de nouvelles variables synthétiques obtenues en combinant les attributs initiaux par des

transformations linéaires. La plus célèbre de ces méthodes est l'analyse en composantes principale (ACP) linéaire. En effet, cette méthode classique est très largement utilisée notamment en analyse des données, afin de chercher des dépendances entre un grand nombre de variables et de pouvoir ainsi les représenter par un petit nombre de facteurs. Cette méthode cherche à exprimer les données observées comme résultant d'une transformation linéaire de variables permettant de trouver le plus petit sous-espace où l'erreur de reconstruction est minimale au sens des moindres carrés, ou de façon équivalente le sous-espace sur lequel les projections linéaires maximisent la variance. Cependant, il s'agit d'une méthode exclusivement linéaire et qui n'est donc pas capable de déceler des dépendances non linéaires entre les variables. Ainsi, afin de pouvoir révéler les liens non linéaires, de nombreuses méthodes ont été proposées ces dernières années. Deux voies principales se sont dégagées : la première basée sur les méthodes neuronales et connue sous le nom de cartes auto-organisatrices de Kohonen [13] et la seconde basée sur des méthodes algébriques [39, 45] et repose sur l'idée de considérer que l'ensemble de données est localement linéaire.

4.2 Méthode basée sur les réseaux de neurones

La classification d'un ensemble d'objets est l'attribution à chacun de ces objets d'une classe parmi plusieurs classes définies à l'avance. Elle peut être de type supervisée ou non supervisée. On parle de classification supervisée lorsque le processus de classification est composé de deux phases. La première est une phase d'apprentissage où on construit un modèle qui décrit un ensemble prédéterminé de classes de données suivi d'une phase de classement. C'est dans cette phase où on utilisera le modèle pour affecter une classe à un nouvel objet. Par contre, en classification non supervisée, le modèle de classification est construit sans aucune supervision. Il n'y a pas de classe définie à priori. C'est le cas par exemple du regroupement (*clustering*). Notons que les applications qui nécessitent un tel classement sont très nombreuses en reconnaissance des formes (chiffres et caractères manuscrits ou imprimés, images, parole) et en fouille de données. Dans ce type d'applications, on est confronté au problème de filtrage d'information comme par exemple trouver automatiquement dans un corpus de données, les textes qui sont

pertinents pour un thème donné. De même, dans un but de visualisation ou d'analyse de données, on dispose d'un ensemble de données représentées par des vecteurs de grande dimension et l'on souhaite trouver une représentation beaucoup plus faible tout en conservant les proximités ou ressemblances entre ces données. Cependant, la classification de telles données est un problème difficile. Les réseaux de neurones offrent un ensemble de techniques puissantes pour réaliser ce genre de tâches. Nous proposons de passer en revue une solution de plus en plus employée dans les problèmes de la RDD. Elle est basée sur un type particulier de réseau de neurones dits réseaux de Kohonen [13] ou encore cartes auto-organisatrices de Kohonen : SOM (*self-organizing maps*).

4.2.1 Le réseau de Kohonen

Les cartes auto-organisatrices de Kohonen seront utilisées dans un but descriptif. Les données à analyser sont constituées d'observations dont on cherche à comprendre la structure. Ce type de réseau est caractérisé par son aptitude à conserver la topologie des observations après projection sur un espace réduit, sa capacité à détecter les relations non linéaires entre les variables observées et une bonne résistance aux données bruitées.

Morphologiquement, le réseau SOM (Fig.4.1) est constitué d'une couche d'entrée avec un neurone pour chacune des n variables observées, d'une couche cachée dont les neurones sont disposés sous forme d'une carte généralement linéaire (appelée corde) ou rectangulaire. Une dimension plus grande de la carte peut être utilisée si cela est jugé nécessaire pour l'interprétation. Chaque neurone de la couche d'entrée est relié à l'ensemble des neurones de la couche cachée par un lien pondéré et les liaisons entre les neurones de la carte sont structurées sous forme d'un graphe non orienté. Cette structure induit une distance discrète pour l'ensemble des neurones de la carte. Elle est définie comme étant le plus court chemin entre deux neurones et permet ainsi de localiser les voisins immédiats d'un neurone sur la carte. Notons que l'algorithme SOM est une variante de l'algorithme des K plus proches voisins (*K-means*) qui lors d'une itération modifie non

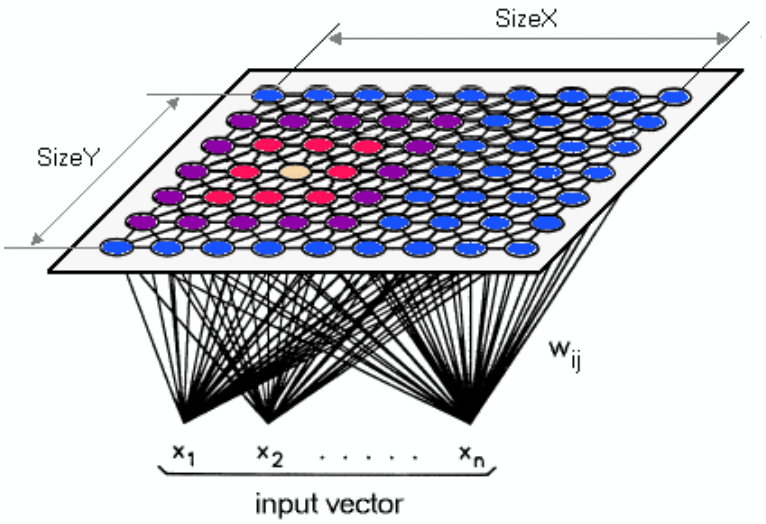

FIG. 4.1 – Réseau de Kohonen

seulement un centre sélectionné comme étant le plus proche d'une donnée, mais aussi les centres voisins pour un graphe de voisinage fixé. Aussi, c'est cette notion de voisinage qui va être exploitée pleinement dans l'algorithme de Kohonen afin de modéliser finement sur la carte la topologie des données observées. Notons que les cartes auto-organisatrices s'organisent par rapport aux exemples d'entrée présentés en respectant les contraintes topologiques de l'espace d'entrée. Il y a mise en correspondance de l'espace d'entrée avec l'espace du réseau. Les zones voisines de l'espace d'entrée sont voisines sur la carte auto-organisatrice, avec en plus une réduction de la dimensionnalité. Ainsi, ce type de réseau possède une bonne aptitude à modéliser des structures complexes, avec prise en compte des relations de type non linéaires qui peuvent exister entre les variables étudiées.

Cependant, cette méthode présente certaines limites. D'une part, le choix de la dimension de la carte de projection au début de l'algorithme n'est pas fondé puisqu'on n'est pas en mesure de prévoir que les données observées auront une dimension intrinsèque inférieure ou égale à la valeur choisie. D'autre part, on note certaines aberrations comme les neurones qui se positionnent sur la carte à des endroits où il n'existe aucun stimulus d'apprentissage. Probablement, la topologie fixée a priori des cartes de Kohonen

constitue un handicap.

4.3 Méthodes algébriques

Ces méthodes ont été développées afin de déceler les dépendances de type non linéaires qui peuvent exister entre les variables étudiées. L'idée principale utilisée par ces méthodes est de considérer que l'ensemble des données est localement linéaire. Nous allons nous limiter à la présentation des deux principales méthodes : plongement localement linéaire : LLE (*Locally Linear Embedding*) et Isomap (*Isometric feature mapping*).

4.3.1 Plongement localement linéaire

La méthode LLE (Fig.4.2) considère que les données globalement non linéaires sont localement linéaires dans de petits voisinages. L'idée véhiculée par cette méthode consiste à déterminer pour chaque vecteur x_i, point de l'espace contenant les données à projeter, ses k voisins calculés sur la base de la distance euclidienne et de considérer que dans ce voisinage, les données sont linéaires. Une fois les voisins de x_i déterminés, on calcule les poids de reconstruction w_{ij} représentant le voisinage de chaque vecteur x_i associé à chaque couple (x_i, x_j), où x_j appartient au voisinage de x_i. Les w_{ij} permettront par la suite de reconstruire la topologie de voisinage de chaque vecteur x_i dans l'espace de projection.

4.3.2 Isomap

La méthode de réduction de dimensionnalité Isomap (*Isometric Feature Mapping*) a été introduite par Tenenbaum [45] et s'inspire de la méthode MDS (*Multidimensional Scaling*) en lui donnant comme métrique la distance curviligne. Cette méthode consiste à calculer le voisinage de chacun des points de l'espace de départ, suivi par la construction d'un graphe reliant tous les points voisins. Chaque arête du graphe est ensuite pondérée par la distance euclidienne entre les deux extrémités de l'arête. Enfin, la distance curviligne entre deux points est estimée par la somme des longueurs des arêtes le long du plus court chemin entre ces points et évaluée en général à l'aide de l'algorithme de Dijkstra. Précisons les étapes de l'algorithme de projection d'Isomap.

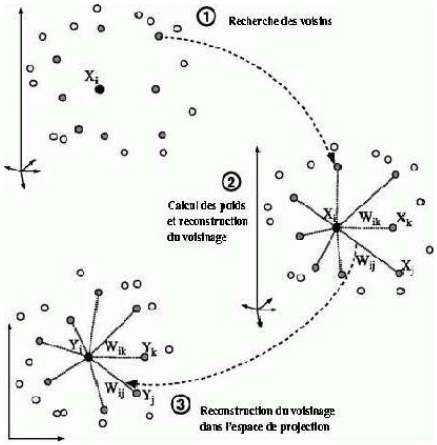

FIG. 4.2 – Étapes de la méthode LLE

1. Construire un graphe de voisinage : on définit un graphe G pour l'ensemble des points de l'espace. Deux points x_i et x_j font partie du graphe si la distance entre ces deux points est inférieure ou égale à un paramètre ϵ (paramètre fixé de Isomap), ou si x_i est un k (paramètre fixé de Isomap) plus proche voisin de x_j.
2. Calculer les plus proches voisins : appliquer l'algorithme de Dijkstra afin d'obtenir la matrice des distances D. Elle contient les distances curvilignes entre chaque point et ses voisins.
3. Déterminer les axes de projection : calculer les vecteurs propres de D et retenir ceux associés aux plus grandes valeurs propres. L'espace de projection utilisé par Isomap est défini par l'espace dont les axes sont ces vecteurs propres retenus.

Notons que la principale limite de ces deux méthodes est leur coût algorithmique élevé lequel est quadratique par rapport au nombre d'observations, ce qui les rend difficilement applicables à de grands volumes de données.

4.4 Synthèse

Le tableau suivant résume les principales caractéristiques des méthodes d'extraction. Pour réduire la dimension de l'espace d'observation, ces méthodes se basent sur des projections. La dimension de cet espace est fixée au début ou à la fin de l'algorithme suivant des heuristiques et non des considérations théoriques capables de déterminer avec précision la dimension de l'espace de projection.

Synthèse						
Méthode	Données	BASE	Nature de la réduction	Critère à optimiser	Limite	Observation
ACP	Quantitatives	Statistiques d'ordre deux	Projection orthogonale de type linéaire permettant de générer des facteurs non corrélés. Ne tient pas compte de la séparabilité des classes.	Axes maximisant la variance des projections.	Non efficace si les données sont peu corrélées	La dimension de l'espace de projection est choisie à la fin de l'algorithme
SOM	Quantitatives	Réseau de neurones type non supervisé.	Projeter les données sur une carte en général à deux ou trois dimensions.	Nombre maximum d'itérations fixé préalablement. ou stabilité des poids des arêtes de connexions durant plusieurs itérations.	Choix de la dimension de la carte au début de l'algorithme.	La dimension de l'espace de réduction est choisie au début de l'algorithme.
ISOMAP	-	Extension du MDS. Utilise la distance euclidienne ou une métrique spécifique au domaine étudié.	Projection sur des hyperplans non linéaires. Moins rapide que LLE, mais plus précis.	Minimiser une fonction de stress.	Coût algorithmique élevé. Ne tient pas compte de la séparabilité des classes.	La dimension de l'espace de réduction est choisie à la fin de l'algorithme.
LLE	-	Considère que les données sont localement linéaires. Se base sur la notion de voisinage. Utilise la distance euclidienne.	Calcul des poids de reconstruction wij associés au couple (xi, xj). Les points projetés doivent conserver au mieux les poids wij.	Minimiser l'erreur de reconstruction.	Coût algorithmique élevé.	La dimension de l'espace de réduction est choisie au début de l'algorithme.

Tab. 4.1 – Comparaison des méthodes d'extraction

4.5 Méthodes basées sur la sélection d'attributs pertinents

Dans un problème d'extraction de connaissances où un attribut représente un élément descriptif d'un objet, la sélection d'attributs pertinents consiste à réduire l'ensemble des attributs considérés. Ceci permet parfois d'augmenter la précision de la prédiction et de réduire le temps de traitement des données.

Nous avons vu au chapitre 2 que l'étude de ce problème se justifie par le fait qu'une recherche exacte a un coût prohibitif $O(2^n)$ en temps de calcul, où n est le nombre d'attributs initiaux. La suite de ce chapitre présente, critique et compare quatre méthodes de réduction de la dimensionnalité basées respectivement sur la théorie des ensembles approximatifs (*Rough set theory* RST), sur l'utilisation des algorithmes génétiques, sur les séparateurs à vaste marge et finalement sur la dimension fractale. On a voulu par ce choix limiter notre exposé aux principales méthodes utilisées pour la réduction de la dimensionnalité des données par sélection d'attributs. Par ailleurs, un ensemble de dix collections variées de données dont la plupart proviennent du répertoire UCI [38] est utilisé pour tester la qualité de la réduction obtenue par ces différentes méthodes. Les principaux facteurs ayant influencé ce choix sont la taille de l'échantillon et le nombre d'attributs. Le tableau suivant récapitule les principales caractéristiques de ces bases.

Caractéristiques des neuf bases du répertoire UCI utilisées pour les différents tests						
Base	**Types de données**	**Tâche**	**Types d'attributs**	**Nb. d'observations**	**Nb. d'attributs**	**valeurs manquantes**
Abalone	Variables multiples	Classification	Qualitative Entier Réel	4177	8	Non
Ecoli	Variables multiples	Classification	Réel	336	8	Non
Magic	Variables multiples	Classification	Réel	19020	11	Non
Cmc	Variables multiples	Classification	Qualitative Entier	1473	9	Non
Diabete	Variables multiples	Classification	Entier, Réel	768	8	Non
Heart	Variables multiples	Classification	Qualitative, Entier, Réel	303	13	Oui
Segment	Variables multiples	Classification	Réel	2310	19	Oui
Yeast	Variables multiples	Classification	Réel	1484	8	Non
Zoo-data	Variables multiples	Classification	Qualitative Entier	101	17	Non

L'analyse comparative des réductions obtenues par chacune de ces méthodes sera exposée à la suite de la présentation de la méthode étudiée. Lors de cette analyse, la meilleure réduction sera celle qui génère le plus petit rapport entre la taille de la réduction sur la taille initiale de l'espace d'observation et la plus petite erreur de classification. Les erreurs de classification dans l'ensemble des expérimentations qui vont suivre seront mesurées à l'aide de l'algorithme C4.5 [36]. Ces erreurs correspondent au pourcentage d'exemples mal classés qui sont obtenus avant et après réduction. Notons que l'algorithme C4.5 est une version améliorée de l'algorithme de classification ID3. Cet algorithme permet de construire un arbre de décision sur la base d'un échantillon c-à-d une structure informatique qui permet de déduire un résultat de test à partir de décisions successives. Chaque noeud de l'arbre est soit une feuille dénotant une décision soit une branche spécifiant un test sur une valeur d'un attribut. L'arbre ainsi construit servira à classer de futures observations.

4.5.1 Méthode basée sur la théorie des ensembles approximatifs

La théorie des ensembles approximatifs a été introduite par Pawlak au début des années 80 [33]. Elle offre un cadre théorique approprié pour quantifier la dépendance entre attributs et pour calculer les sous-ensembles d'attributs, appelés réductions, qui préservent la qualité des données du système d'information de départ.

La RST nous permet de déterminer deux types de réduction : les réductions absolues et les réductions relatives. Les réductions absolues sont indépendantes de toutes classifications prédéfinies, alors que les réductions relatives en dépendent. Cette dépendance est usuellement définie selon un attribut de décision.

Fondements théoriques de la RST

La RST permet de calculer les sous-ensembles d'attributs appelés réductions en se basant sur le degré de similitude entre les objets de l'univers, dont le principe repose sur la notion de l'indiscernabilité. Dans ce qui suit, nous allons passer en revue les notions fondamentales sur lesquelles repose cette théorie et dont nous aurons besoin pour notre étude. Pour plus de détail, le lecteur pourra consulter [33]. Notre démarche pour

la réduction des attributs basée sur la RST repose sur les notions de système d'information, relation d'indiscernabilité, matrice de discernabilité, fonction de discernabilité et processus de calcul des réductions.

Le système d'information

La théorie des ensembles approximatifs suppose la présence d'une information préalable sur les différents objets de l'univers. Chaque objet est décrit par un ensemble d'attributs. L'ensemble de ces objets est regroupé dans un système d'information. Tout système d'information est décrit par un quadruplet S = (U, Q, V, F) où :

- U est un ensemble fini d'objets appelé univers.
- Q est un ensemble fini d'attributs. Il est généralement composé de deux sous-ensembles d'attributs disjoints $Q = D \bigcup C$, où D représente l'ensemble des attributs décrivant les objets et C un attribut (ou plusieurs attributs) de classification.
- $V = \bigcup Vq$, où Vq est le domaine de valeurs de l'attribut q.
- F une fonction informative qui affecte une valeur à chaque attribut de chaque objet. $F : U \times Q \longrightarrow V$ tel que $\forall x \in U$ et $\forall q \in Q, F(x, q) \in V_q$.

La relation d'indiscernabilité

Deux objets de l'univers sont dits indiscernables par rapport à un ensemble d'attributs s'ils possèdent des valeurs identiques pour chacun des attributs de cet ensemble. Notons que la relation d'indiscernabilité est une relation d'équivalence, ce qui nous donne la possibilité de partitionner l'univers U sur lequel est définie cette relation en classes d'équivalence. Cette notion est essentielle pour définir la réduction des attributs dans la théorie des ensembles approximatifs.

Une réduction d'attributs au sens de la RST est un sous-ensemble minimal d'attributs tel que tout sur ensemble de ce sous-ensemble définit la même relation d'indiscernabilité sur l'univers U. Donc, ces sous-ensembles définissent le même partitionnement de l'univers au sens de la relation d'indiscernabilité. Skowron et Rauszer [42, 43] ont proposé un algorithme de calcul de réductions basé sur la matrice de discernabilité. Ils ont montré que le problème de calcul des réductions en RST est transformable en un

problème de recherche d'une fonction de discernabilité.

La matrice de discernabilité

Étant donné un système d'information S de dimension $n \times m$ tel que n représente la cardinalité de l'univers U et m le nombre d'attributs caractérisant ces objets, nous notons D la matrice de taille $n \times n$, appelée matrice de discernabilté telle que chaque élément de la matrice D[i,j] représente un sous-ensemble d'attributs qui discerne les deux objets de l'univers $x_i, x_j : D[i,j] = \{q \in P | F(xi,q) \neq F(xj,q)\}\ avec\ 1 < i < j \leq n$.

La fonction de discernabilité

Soit D une matrice de discernabilité d'un système d'information de n objets et m attributs. La fonction de discernabilité est une fonction booléenne de m variables $(a_1, a_2, , a_m)$ qui correspondent respectivement aux attributs $(q_1, q_2, , q_m)$. Elle est définie comme suit : $F(a_1, a_2, , a_m) = \wedge(\vee D(x,y)\ |1 \leq i < j \leq n,\ D(x,y) \neq \oslash)$. L'ensemble des réductions consiste à trouver la forme normale disjonctive de cette fonction. Rappelons qu'une disjonction est une proposition de la forme $P \vee Q$ et une conjonction est une proposition de la forme $P \wedge Q$. On dit d'une fonction exprimée sous forme d'une proposition qu'elle est en forme normale disjonctive si elle est composée de disjonctions de conjonctions. L'exemple qui suit illustre une fonction F de discernabilité et sa transformée en forme normale disjonctive.

Processus de calcul des réductions : Le processus de calcul des réductions se résume en deux étapes (voir l'exemple qui suit). On commence par calculer la matrice de discernabilité intitulée D. Chaque élément $D[i,j]$ de cette matrice contient un ensemble d'attributs discernant les objets x_i et x_j. Nous poursuivons par le calcul de la fonction de discernabilité F. Le processus de définition de cette fonction produit une forme conjonctive de termes. Chaque terme représente une forme normale disjonctive des attributs correspondant à chaque élément de D. Dans la deuxième étape, on transforme la fonction de discernabilité de sa forme normale conjonctive en une forme normale disjonctive. A chaque itération, on applique les règles d'absorption (chaque terme absorbe

ses sur-termes) entre les termes produits. Par conséquent, chaque terme final de cette forme disjonctive représente une réduction. Notons que si tous les éléments de la matrice D sont pris en considération, nous obtenons les réductions absolues. Les réductions relatives par rapport à un attribut de décision sont obtenues en ne prenant en compte que les éléments de la matrice D contenant cet attribut.

L'exemple suivant illustre la procédure de réduction de la dimensionnalité par la méthode des ensembles approximatifs. Dans cet exemple, on considère une base composée de six attributs $\{W, X, Y, Z, C\}$ dont un attribut de classification $\{C\}$ et de huit observations $\{E1, E2, E3, E4, E5, E6, E7, E8\}$.

Exemple illustrant la réduction de la dimensionnalité par la méthode RST

Univers					
	W	X	Y	Z	C
E1	5	1	3	2	2
E2	4	4	4	2	0
E3	3	2	2	0	1
E4	5	3	4	1	1
E5	3	4	3	1	2
E6	4	1	3	0	0
E7	3	3	4	1	1
E8	5	3	4	2	2

Fonction F de discernabilité
$F(W,X,Y,Z,C) = (X \vee Y) \wedge (W \vee X) \wedge (Z)$
La fonction normale disjonctive
$F(W,X,Y,Z,C) = (X \wedge Z) \vee (W \wedge Y \wedge Z)$
Ensemble des réductions
$Red = \{\{X,Z\},\{W,Y,Z\}\}$

Matrice de discernabilité								
	E1	E2	E3	E4	E5	E6	E7	E8
E1		$\{W,X,Y,C\}$	$\{W,X,Y,Z,C\}$	$\{X,Y,Z,C\}$	$\{W,X,Z\}$	$\{W,Z,C\}$	$\{W,X,Y,Z,C\}$	$\{X,Y\}$
E2			$\{W,X,Y,Z,C\}$	$\{W,X,Z,C\}$	$\{W,Y,Z,C\}$	$\{X,Y,Z\}$	$\{W,X,Z,C\}$	$\{W,X,C\}$
E3				$\{W,X,Y,Z\}$	$\{X,Y,Z,C\}$	$\{W,X,Y,C\}$	$\{X,Y,Z\}$	$\{W,X,Y,Z,C\}$
E4					$\{W,X,Y,C\}$	$\{W,X,Y,Z,C\}$	$\{W\}$	$\{Z,C\}$
E5						$\{W,X,Z,C\}$	$\{X,Y,C\}$	$\{W,X,Y,Z\}$
E6							$\{W,X,Z,C\}$	$\{W,X,Y,Z,C\}$
E7								$\{W,Z,C\}$

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Analyse des réductions générées**					
Base	NB. enreg.	NB. d'attributs	NB. des réductions	Err. avant réduction en %	Err. moyenne après réduction en %
Zoo-Data	101	17	33	8	7
Ecoli	336	8	7	15.77	30.36
Cave	1225	6	10	4.90	9.14
Diabete	768	9	21	25.78	26
Abalone	4177	9	13	79.63	78
Yeast	1484	9	4	44.0027	49.7
Magic	19020	11	33	14.94	24.14
Heart	303	14	90	20.79	28.82
Segment	2310	19	95	2.86	33.77
Cmc	1473	10	Pas de réduction	46.78	46.78

TAB. 4.2 – Algorithme basé sur la méthode RST (ALGRST)

Une analyse des résultats obtenus ci-dessus met en évidence une carence au niveau de la méthode RST. Il s'agit du nombre de réductions générées par cette méthode. À titre d'exemple, on a obtenu quatre-vingt-dix réductions pour la base Heart. Notons que ce nombre peut dépasser mille réductions dans certains cas. Alors doit-on les considérer toutes ou se limiter à certaines d'entre elles ? Afin de répondre à cette question, nous avons choisi au hasard trois réductions de tailles différentes et nous leur avons appliqué l'algorithme de classification C4.5 afin de comparer la qualité de ces réductions. Nous avons constaté que les erreurs de classification obtenues étaient très différentes les unes des autres. Prenons l'exemple des deux réductions ayant généré respectivement des erreurs de classification de l'ordre de 36% et 24%. On peut se demander s'il existe parmi ces quatre-vingt-dix réductions une qui pourra générer une erreur de classification inférieure à 24% ou peut être une réduction qui permet d'améliorer le taux d'erreur de classification obtenu avant la réduction. À cette fin, on doit tester l'ensemble des réductions, ce qui nécessite dans la majorité des cas des temps de traitement parfois assez importants, sans oublier le temps algorithmique nécessaire pour générer les réductions.

Principales limites

L'algorithme proposé par Skowron et Rauszer [42, 43] utilise une approche basée sur le calcul de la discernabilité entre tous les objets de l'univers et utilise la totalité de la matrice de discernabilité, ce qui impose un temps minimum quadratique pour cette première phase de l'algorithme. Notons que l'ensemble des algorithmes de la RST souffre d'un coût algorithmique assez élevé qui les rend difficilement applicable à de grands volumes de données.

4.5.2 Méthode basée sur les algorithmes génétiques

Dans cette section, nous allons commencer par présenter les algorithmes génétiques et identifier les points importants de ces algorithmes. Nous citerons les principaux travaux réalisés dans le domaine de la sélection d'attributs pertinents basés sur les algorithmes génétiques.

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature : croisement, mutation et sélection. Ces algorithmes appartiennent à la classe des algorithmes évolutionnaires du fait que le principe de l'évolution constitue la clé de leur fonctionnement. Contrairement à un grand nombre de méthodes d'optimisation, les algorithmes génétiques travaillent sur une population de solutions potentielles, ce qui leur permet d'explorer plusieurs zones de l'espace à la fois et constitue de ce fait l'un des points forts de ce type d'algorithme.

Développés dans les années 70 par Holland [20], puis approfondis par Goldberg [16], ces algorithmes sont utilisés particulièrement dans les problèmes d'extraction de connaissances, la recherche de règles d'association, la compression d'images, la reconnaissance des formes et l'optimisation des réseaux de neurones. La particularité de ces algorithmes est le fait qu'ils font évoluer des populations d'individus codés par une chaîne binaire, et utilisent les opérateurs de mutation binaire et de recombinaison de différents types.

Objectif des algorithmes génétiques

L'objectif des algorithmes génétiques est de déterminer les extrêmes d'une fonction $f : X \longrightarrow R$ où X est un ensemble quelconque appelé espace de recherche et f est appelée fonction d'adaptation ou fonction d'évaluation ou encore fonction "fitness". Ces algorithmes sont utilisés comme alternative à l'optimisation de fonctions là où les méthodes classiques ne permettent pas d'obtenir un équilibre satisfaisant entre la performance du système d'optimisation et le coût nécessaire pour atteindre cet optimum.

Fondements théoriques des algorithmes génétiques

Les algorithmes génétiques fonctionnent avec une population regroupant un ensemble d'individus appelés chromosomes. Chaque chromosome est constitué d'un ensemble de gènes lesquels peuvent prendre différentes valeurs appelées allèles. La position d'un gène dans un chromosome est identifiée par son locus (position dans la chaîne représentant le chromosome). Les algorithmes génétiques sont constitués d'un ensemble de cycles d'opérations génétiques. Dans chacun de ces cycles, une nouvelle population plus adaptée que son prédécesseur, appelée génération est créée. Cette évolution des générations est effectuée par l'intermédiaire des opérations de reproduction, de croisement et de mutation.

Fonctionnement d'un algorithme génétique

On dispose d'une population de chaînes de caractères sur un alphabet que nous supposerons binaire. À chaque individu est associé un score "fitness". Trois opérateurs vont être utilisés :

- Un opérateur de sélection (*select*) dont le rôle est de choisir les individus les plus adaptés, ceux qui possèdent un score élevé.
- Un opérateur de croisement.
- Un opérateur de mutation.

Algorithme

Algorithm 1: Schéma général d'un algorithme génétique

Data: Population initiale : P

Résultat du croisement : R

Compteur : T

Result: Population sélectionnée : Q

Initialisation T = 0 ;

while *(critère d'arrêt non atteint)* **do**

Évaluer(P,T) ;

Q = Sélectionner(P,T) ;

R = Croisement(Q) ;

P = Mutation(R) ;

T = T + 1 ;

end

Toute utilisation d'algorithmes génétiques est précédée par un codage adapté au problème (codage des chromosomes) et la définition d'une fonction "fitness" qui va caractériser l'adéquation de la solution au problème.

Sélection d'attributs pertinents à l'aide d'algorithmes génétiques

Comme il a été mentionné ci-dessus, la sélection d'attributs consiste à rechercher un sous-ensemble d'attributs pour réduire le temps de traitement et/ou améliorer la classification. Nous avons mentionné et présenté les deux méthodes d'évaluation des sous-ensembles d'attributs sélectionnées, les *wrappers* et les *filters*.

La sélection d'attributs par algorithmes génétiques utilise principalement deux types de codage des individus. La première représentation considère que l'espace de recherche peut être décrit par tous les sous-ensembles possibles d'attributs et code donc une solution par une chaîne de bits d'une longueur fixe de m bits où m représente le nombre d'attributs disponibles. Le i-ème bit indique si le i-ème attribut est sélectionné

(valeur du bit à un) ou non (valeur du bit à zéro). L'avantage de ce codage réside dans sa simplicité et la possibilité d'utiliser des opérateurs de croisement et de mutation non dépendants du problème [22]. Certains auteurs proposent des opérateurs adaptés au problème de sélection d'attributs et notamment un opérateur de croisement, le "non-standard crossover" [17]. En effet, les auteurs considèrent que dans les opérateurs de croisement standards, les zéros et les uns sont considérés de la même façon alors qu'ils n'apportent pas la même information. Chen [7] considère que les "1" qui représentent les attributs potentiellement intéressants apportent plus d'information que les "0". Un opérateur de croisement particulier est utilisé et permet à partir des attributs sélectionnés communs, de générer de nouveaux individus.

La seconde représentation a été proposée par Cherkauer [8] dans laquelle un individu est de taille fixe égale au nombre d'attributs et est codé par l'index des attributs sélectionnés et des zéros. L'auteur autorise les attributs à être présents plusieurs fois car il considère que cette redondance peut ralentir la perte de diversité. Pour lui permettre de travailler avec cette représentation, il utilise une variante du "crossover" uniforme et un opérateur de mutation. En plus de ces deux opérateurs, il introduit un opérateur "Delete Feature" qui, à partir d'un individu, génère un nouvel individu en lui enlevant un attribut sélectionné de façon aléatoire. Si l'individu possède cet attribut en plusieurs exemplaires, tous sont alors supprimés. Notons que l'évaluation des sous-ensembles générés se fait dans la majorité de cas en utilisant les méthodes enveloppantes avec C4.5 [36, 8], des tables de décision euclidiennes [18], des arbres de décision [49, 2], les réseaux de neurones [31] et les k plus proches voisins [34].

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Analyse des réductions générées**					
Base	NB. enreg.	NB. d'attributs	Taille des réductions	Err. avant réduction en %	Err. moyenne après réduction en %
Zoo-Data	101	17	[5; 6]	8	5.94
Ecoli	336	8	3	15.77	22.62
Cave	1225	6	2	4.90	9.14
Diabete	768	9	3	25.78	25
Abalone	4177	9	[3; 4]	79.63	77.90
Yeast	1484	9	[4; 5]	44.0027	59.77
Magic	19020	11	2	14.94	30.49
Heart	303	14	[3; 4; 5; 6]	20.79	25.35
Segment	2310	19	[4; 5]	2.86	30.25
Cmc	1473	10	Pas de réduction	46.78	46.78

TAB. 4.3 – Algorithmes génétiques (ALG)

L'analyse des résultats obtenus ci-dessus montrent que la précision des réductions générées par cette méthode reste sensiblement la même que celle obtenue avec la méthode RST. Cependant, on peut faire varier le nombre de réductions générées ainsi que leur taille en modifiant certains paramètres de l'algorithme. Cela ne garantit pas l'obtention de réductions de meilleure qualité mais nécessite la recherche, parmi l'ensemble des réductions obtenues de celles qui génèrent des taux d'erreurs acceptables comparativement à ceux obtenus avant la réduction. Précisons que la qualité des réductions va être mesurée à l'aide d'un algorithme de classification et donc le résultat obtenu pourra dépendre de cet algorithme.

Principales limites

1. Difficulté à choisir les bons paramètres due au fait que ces paramètres dépendent du problème à résoudre.
2. Choix du critère d'arrêt. Comment peut on savoir que la solution optimale du problème a été trouvée ?

4.5.3 Méthode basée sur le séparateur à vaste marge (SVM)

Le séparateur à vaste marge est une méthode de classification par apprentissage supervisé. Elle fut introduite par Vapnik [12, 29] et elle est basée sur l'utilisation de fonctions dites noyau (*Kernel)* [32] qui permettent une séparation optimale des données.

Objectifs de la méthode

Le séparateur à vaste marge (SVM) permet de trouver un hyperplan qui va séparer les données et maximiser la distance entre deux classes. Les points les plus proches qui sont utilisés pour déterminer l'hyperplan sont appelés vecteur de support et la distance minimale qui sépare l'hyperplan des exemples d'apprentissage s'appelle la marge. Trois cas de figures peuvent se présenter. Le cas où les données sont linéairement séparables, le cas où elles ne sont pas séparables linéairement et le cas où elles ne peuvent pas être séparées. Notons cependant que le cas où les données ne sont pas séparables linéairement peut être ramené à un problème linéairement séparable. L'idée consiste à projeter les données dans un espace de dimension plus grande ($Fig.4.3$) [29], ce qui aura pour effet de permettre une séparation linéaire des données observées. Cette transformation est réalisée par l'intermédiaire d'une fonction noyau dont voici quelques exemples.

Exemples de Fonction Noyau	
Type de Noyau	**Fonction k(x,x')**
Linéaire	x.x'
Polynomial	$(x.x')^d$ ou $(C + x.x')^d$
Gaussien	$e^{-\|\|x-x'\|\|^{2/}\sigma}$
Laplacien	$e^{-\|\|x-x'\|\|/\sigma}$

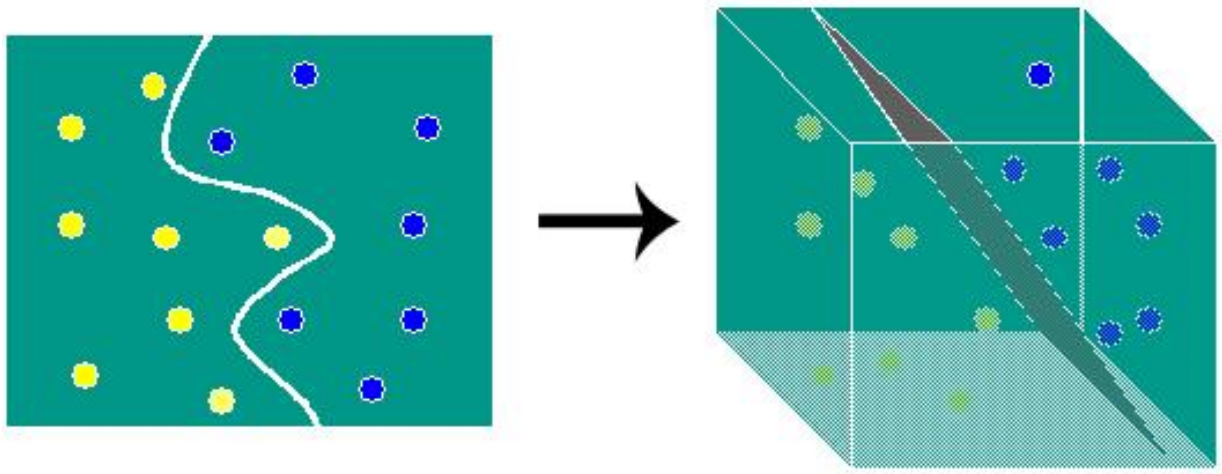

FIG. 4.3 – Séparation de données non linéairement séparables par projection.

Fondements théoriques

Nous allons décrire brièvement le principe des séparateurs à vaste marge. Pour de plus amples informations, le lecteur pourra consulter [12]. Dans ce descriptif, nous allons nous limiter au cas linéairement séparable.

Étant donné un ensemble de points d'apprentissage (x_i, y_i) avec $1 \leq i \leq l$ *où* chaque $x_i \in R^d$ et $y_i \in (-1, +1)$. Dans le cas présent *où* le modèle traité est de type linéaire, nous pouvons écrire : $y_i = w.x_i + \mathrm{b}$, et donc l'hyperplan séparateur a pour équation $\mathrm{w.x} + \mathrm{b} = 0$. Aussi, la distance d'un point au plan séparateur est donnée par : $d(x) = |z.x+b|/||w||$. Donc si x_1 et x_2 sont deux points situés de part et d'autre du plan optimal, alors la distance entre ces deux points est de la forme : $w.(x_1 - x_2)/||w|| = 2/||w||$. Maximiser cette distance revient alors à minimiser $||w||$ sous certaines contraintes.

Sélection des variables par SVM

Comme nous l'avons mentionné précédemment, les algorithmes de sélection de variables se composent de trois phases : une pour le parcours de l'ensemble des variables, la suivante pour évaluer ces variables et la dernière pour arrêter l'algorithme. Aussi, l'algorithme de sélection de variables basé sur le modèle SVM [37, 3] utilise dans la phase de parcours la méthode consistant à effectuer une suppression séquentielle des variables. Quant à l'évaluation de la pertinence des attributs, l'algorithme préconise l'utilisation de la marge comme moyen d'évaluer l'importance d'un attribut. Ainsi, un attribut peu informatif sera celui auquel la marge $2/||w||$ est peu sensible. L'algorithme s'arrête quand il n'ya plus d'attribut à tester.

Algorithm 2: Algorithme de sélection d'attributs basé sur SVM (ALGSVM)

Data: Ensemble des attributs à traiter : var

Paramètre de la fonction de décision : w

Un attribut de var : v

Result: Ensemble des attributs sélectionnés classés par ordre d'importance : trie

Initialisation trie = {} ; var = $\{a_1, .., a_d\}$;

while *(tous les attributs ne sont pas triés)* **do**

Évaluer la marge $2/||w||$ de la fonction de décision utilisant les variables $\in$ var;

$\forall v \in$ var, calculer la sensibilité de la marge par rapport à chaque variable;

Sélectionner la variable v $\in$ var qui minimise la sensibilité;

Classer cette variable : trie = $\{v\} \bigcup trie$;

Supprimer v de l'ensemble var;

end

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Analyse des réductions générées**					
Base	NB. enreg.	NB. d'attributs	Taille réductions	Err. avant réduction en %	Err. après réduction en %
Zoo-Data	101	16	8	8	2.97
Ecoli	336	7	7	15.77	15.77
Cave	1225	5	4	4.90	4.57
Diabete	768	8	6	25.78	24.35
Abalone	4177	8	7	79.63	79.22
Yeast	1484	8	6	44.0027	44.81
Magic	19020	10	7	14.94	14.45
Heart	303	13	6	20.79	20.79
Segment	2310	18	6	2.86	4.11
Cmc	1478	10	Pas de réduction	46.78	46.78

TAB. 4.4 – Algorithme basé sur la méthode SVM (ALGSVM)

Parmi les trois algorithmes analysés jusqu'a présent, c'est l'algorithme ALGSVM relatif à la méthode SVM qui fournit les réductions donnant les meilleurs taux d'erreurs de classification. Cependant, à l'exception de quelques bases, le pourcentage moyen de réduction reste faible de l'ordre de 30 %.

Principales limites :

1. Dans un problème où les données ne sont pas linéairement séparables, la méthode SVM nécessite la réalisation d'un ensemble de tests afin de déterminer la fonction noyau qui convient le mieux aux données traitées, sans oublier l'opération qui consiste à s'assurer que les données qu'on traite ne sont pas linéairement séparables.
2. La performance de l'algorithme de sélection des variables basé sur SVM dépend du nombre d'exemples d'apprentissage dans la base. Plus ce nombre est important,

meilleure est la performance.

3. La complexité de cet algorithme dépend du nombre d'entrées à classer et du nombre d'exemples d'apprentissage n, et ce selon la relation :
   $n^2d \leq$ complexité $\leq n^3d$. Donc, ce type d'algorithme sera difficilement applicable en haute dimensionnalité.

4.5.4 Méthode basée sur la dimension fractale

Des études menées par Faloustos et al. [48, 47] ont permis de mettre en évidence le lien qui existe entre la dimension de corrélation et la pertinence des attributs au sein des données. Si on note FD la dimension de corrélation de l'ensemble des attributs, FD_p la dimension de corrélation partielle (i.e, la dimension de corrélation de l'ensemble des attributs à l'exception de l'attribut évalué), alors plus l'attribut considéré est important pour les données, plus grande sera la différence $FD - FD_p$. Notons que dans une telle approche, on est amené à fixer un seuil de pertinence d'un attribut pour les données observées. Cela nous amène à poser la question suivante : Sur quelle base allons-nous fixer ce seuil ?

Sélection d'attributs sur la base de la dimension de corrélation

Nous avons défini en annexe la notion de dimension de boîte (*Box-counting dimension*), nous allons montrer comment on la calcule en pratique. Soit l'expression : $C(r) = [2/(N*(N-1)]*\sum I*(||x_j - x_i||) \leq r, 1 \leq i \leq j \leq N$, où r représente une suite géométrique de premier terme $1/2$, N le nombre d'observations et I une fonction définie comme suit :

$$I : R \longrightarrow \{0,1\}\, tel \text{ que} : \lambda \longrightarrow I(\lambda) = 1 \text{ si } ||x_j - x_i|| \leq r \text{ et } 0 \text{ autrement.}$$

alors la quantité FD définie par : lim de $[Ln(C(r))]/[Ln(r)]$ quand r tend vers zéro, avec Ln le logarithme népérien, représente la dimension de corrélation de l'ensemble des données observées.

Algorithm 3: Algorithme de Faloustos / ALGFAL

Data: L'ensemble S des attributs à traiter

Result: Liste des attributs dans l'ordre inverse de leur importance

1. Calculer la dimension de corrélation D de l'ensemble de données ;
2. Initialisation : S = Ensemble des attributs ;

while *(Il y a un attribut pertinent)* **do**

3. Pour chaque attribut, calculer la dimension de corrélation partielle PD_i, obtenue en utilisant l'ensemble des attributs à l'exception de l'attribut a_i;
4. Trier les dimensions PD_i obtenues à l'étape 3, et sélectionner l'attribut a_i qui minimise la différence $(D - PD_i)$;
5. $D = PD_i$;
6. Afficher l'attribut a_i et supprimer-le de l'ensemble S;

end

NB. Un attribut non pertinent sera celui qui minimise la différence $(D - PD_i)$

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Analyse des réductions générées**					
Base	NB. enreg.	NB. d'attributs	Taille réductions	Err. avant réduction en %	Err. après réduction en %
Zoo-Data	101	17	8	8	21.78
Ecoli	336	8	5	15.77	17.56
Cave	1225	6	5	4.90	4.90
Diabete	768	9	6	25.78	25.52
Abalone	4177	9	7	79.63	78.21
Yeast	1484	9	4	44.0027	53.17
Magic	19020	11	7	14.94	16.79
Heart	303	14	4	20.79	31.35
Segment	2310	19	8	2.86	4.72
Cmc	1473	10	6	46.78	44.94

TAB. 4.5 – Algorithme basé sur la méthode de Faloustos (ALGFAL)

L'analyse du tableau précédent, montre que toutes les réductions obtenues à l'aide de ALGFAL ont une taille supérieure à celle obtenue par ALGRST et ALG. Cependant, la performance des résultats reste sensiblement la même que celle obtenue avec les deux algorithmes cités précédemment, à l'exception de la base Magic, où ALGFAL donne un meilleur taux d'erreur, sans pour autant le réduire à une valeur inférieure à celle obtenue avant la réduction. En revanche, la complexité algorithmique de ALGFAL est meilleure puisque elle est quadratique par rapport au nombre d'attributs et non par rapport au nombre d'observations. Notons cependant que cela pourrait être problématique dans le cas d'une base où le nombre d'attributs est largement supérieur au nombre d'observations. Dans ce cas, nous devons faire appel à d'autres algorithmes pour faire face au problème de la malédiction de la dimensionnalité. Ceci dit, cet algorithme a certains mérites par rapport aux méthodes citées précédemment. Il est en mesure de détecter les corrélations linéaires, non linéaires ainsi que les relations de type non polynomiales qui peuvent exister entre les attributs. Il peut être également appliqué aux ensembles de données à haute dimensionnalité. Notons que l'algorithme utilisé pour évaluer la dimension de corrélation a une complexité temporelle linéaire. Donc, ALGFAL a une complexité temporelle quadratique par rapport au nombre d'attributs et non par rapport au nombre d'observations, comme c'est le cas pour ALGSVM ou les algorithmes basés sur les ensembles approximatifs.

Cependant, ALGFAL présente certaines limites : (*i*) la pertinence d'un attribut dépend d'un seuil qu'on doit fixer au début de l'algorithme, (*ii*) les attributs sélectionnés par cette approche sont considérés comme ayant tous la même importance au sein des données, ce qui en général n'est pas le cas, et pour finir, (*iii*) un attribut écarté car jugé peu pertinent pour les données ne sera plus jamais reconsidéré suite à la suppression d'un autre attribut. Or, la pertinence d'un attribut peut être cachée par un autre attribut.

Chapitre 5

Méthodologie

5.1 Objectifs

Ce projet de recherche traite l'aspect de réduction de la dimensionnalité par sélection d'attributs pertinents. Dans le chapitre précédent, nous avons passé en revue les deux principales approches permettant de réduire la dimensionnalité des données : l'approche par extraction et celle basée sur la sélection d'attributs pertinents. On a discuté aussi des avantages et des inconvénients de chacune de ces approches et des raisons qui nous ont conduit à choisir l'approche basée sur la sélection d'attributs décrivant le mieux possible les données analysées.

Cependant, la méthode adoptée soulève de nombreux problèmes et nous amène à émettre des réserves et observations. Dans ce chapitre, nous allons décrire ces différents problèmes et les solutions que nous leur avons apportées. Au cours de notre démarche et quand cela sera jugé nécessaire, chaque problème et sa solution seront illustrés par un exemple. Ceci nous permettra de préciser l'approche qui nous a permis de concevoir un algorithme efficace permettant de réaliser nos objectifs. Nous terminerons par une série de tests permettant de valider notre démarche.

5.2 Problèmes posés et solutions apportées

Rappelons que toute méthode de RDD basée sur la sélection d'attributs pertinents comporte trois phases. La première consiste à trouver un moyen adéquat pour explorer l'ensemble des variables étudiées. Durant la seconde phase, on choisit un critère pour mesurer la pertinence de ces variables et finalement, on évalue les sous-ensembles générés.

Notre démarche de RDD utilise une méthode de sélection de type ajout. Donc, nous étions amenés à trouver ou à élaborer un critère adéquat pour évaluer la pertinence des attributs au sein des données. À cette fin, différentes techniques peuvent être utilisées. Au mieux de notre connaissance, le plus souvent, on fait appel à la théorie de l'information [49, 24] pour faire cette évaluation et plus précisément on utilisera l'entropie, le coefficient de Gini ou l'information mutuelle pour déterminer la pertinence d'un attribut. Notons que dans un problème de classification, l'entropie sera utilisée pour mesurer la quantité moyenne d'information nécessaire pour identifier la classe d'un objet. Si S représente un échantillon et $S_1, S_2, ..., S_k$ sa partition suivant les classes de l'attribut du test, alors l'entropie de S est donnée par $E(S) = -\sum \frac{|S_i|}{|S|}.Log_2(\frac{|S_i|}{|S|})$ et le coefficient de Gini de S par $Gini(S) = \sum \frac{|S_i|}{|S|}.(1 - \frac{|S_i|}{|S|})$. Ce dernier sera employé pour connaître le taux de disparité et d'hétérogénéité dans un échantillon. Plus la valeur de ce coefficient est proche de 1, plus grande est la disparité dans l'échantillon. Dans l'expression de l'entropie et du coefficient de Gini, les sommes portent sur un indice i prenant ses valeurs dans $\{1, 2, ..., k\}$.

Considérons maintenant l'entropie et l'information mutuelle [49, 24] comme critères pour évaluer les attributs et notons que l'information mutuelle de deux variables aléatoires est une quantité mesurant la dépendance statistique de ces variables. La valeur de cette dernière donne une idée sur la quantité d'information apportée en moyenne par une réalisation de la première variable sur les probabilités de réalisation de la seconde variable. Dans le cas où on emploie l'entropie comme critère d'évaluation, l'approche consiste à sélectionner les attributs capables de faire chuter l'entropie globale du système analysé. Le but étant, dans un problème de classification (cf. la sous-section 4.2), de faire

tendre l'entropie du système vers une valeur égale ou très proche de zéro afin de réduire ou d'annuler les erreurs de classification. Si par contre on utilise l'information mutuelle comme moyen d'évaluation, alors il s'agit dans ce cas de trouver le sous-ensemble d'attributs permettant de générer une quantité d'information optimale. Toutefois, les approches utilisant ces méthodes sont réputées ne pas résister au bruit et le critère d'évaluation peut favoriser certains attributs au détriment d'autres. D'autres approches ont été élaborées afin de contrer certaines de ces limites et fournir généralement de meilleurs résultats. On peut citer la méthode SVM discutée précédemment. Nous avons pu constater la qualité des réductions qu'elle génère. Plus récemment, des méthodes basées sur la dimension fractale ont vu le jour. Comme nous l'avons déjà mentionné, la dimension fractale est sensible à la pertinence d'un attribut. Notre algorithme de réduction l'utilisera comme critère d'évaluation. On discutera des raisons de ce choix ultérieurement dans ce chapitre.

Penchons nous maintenant sur le problème qui consiste à déterminer à quel moment et sur quelle base doit-on arrêter le processus qui génère les sous-ensembles. Or, comme nous l'avons déjà abordé, la génération des sous-ensembles se fait soit par ajout, par retrait ou en combinant les deux méthodes. Dans le cas où l'approche par retrait est utilisée, la réponse est simple. On arrête le processus quand il n'y a plus d'attributs à tester. Mais, la question reste entière dans les deux autres cas. Notons que ce même problème existe aussi dans les méthodes par extraction, car en général ce sont des méthodes basées sur des projections. Dans un tel cas, l'estimation de la dimension de l'espace de projection est faite sur la base de heuristiques. Notons que les méthodes de sélection d'attributs de type ajout (*forward*) peuvent employer différents critères d'arrêt. On peut utiliser la quantité d'information apportée par les sous-ensembles générés. Dans ce cas, on arrête le processus qui génère les sous-ensembles quand le gain d'information mesuré par exemple à l'aide de l'information mutuelle devient stable. On peut aussi fixer un seuil d'erreur à ne pas dépasser et employer des algorithmes de type ID3 [35], CART [6] ou C4.5 [36] afin d'évaluer la pertinence des sous-ensembles générés. Cependant, cette façon de faire soulève plusieurs questions : existe t-il une règle permettant de fixer judicieusement le seuil d'erreur ? Le gain apporté par les attributs ne change-t-il plus à cause d'un mauvais choix des sous-ensembles ? L'utilisation des algorithmes ID3, CART

ou C4.5 pour valider les réductions n'influence t-elle pas le résultat obtenu ? Autrement dit, l'utilisation d'un autre algorithme donnerait-il un résultat différent ?

Une solution à ce dernier problème consiste à trouver un moyen adéquat pour évaluer la dimension réelle des données appelée dimension intrinsèque. C'est le nombre d'attributs nécessaire et suffisant pour représenter efficacement les données analysées. Dans notre approche de RDD, nous avons tenté d'utiliser cette solution. Signalons qu'il existe plusieurs techniques pour évaluer adéquatement cette dimension. On trouve une approche basée sur l'entropie [49, 24], une autre utilise la dimension fractale et plus précisément la dimension de corrélation [48] et finalement une approche basée sur l'algorithme des K-plus proches voisins [11]. Cependant et sur la base de ce qui a été dit en annexe, seule la méthode basée sur la dimension fractale apporte une réponse appuyée par un fondement théorique adéquat permettant de faire une telle estimation. Nous avons donc choisi d'utiliser la dimension fractale pour calculer la dimension intrinsèque d'un ensemble de données. Pourtant, là encore, on fait face à un autre type de problème dans l'estimation de cette dimension. Il a été montré [44] qu'afin d'estimer convenablement la dimension intrinsèque d'un ensemble sur la base de la dimension fractale, le nombre d'observations N présentes dans cet espace doit vérifier la relation : $D < 2 * Log(N)$, où D est le nombre de variables observées.

A titre d'exemple, considérons une base avec 20 attributs, ce qui est loin d'être un cas rare. Pour avoir une bonne estimation de la dimension intrinsèque des données de cette base, le nombre d'observations dont nous devons disposer devra être supérieur ou égal à 10^{10} observations !

Après avoir passé en revue certains problèmes rencontrés dans notre démarche, voyons maintenant les solutions suggérées. Afin d'évaluer la pertinence d'un attribut, nous avons procédé comme suit : on part d'un ensemble vide S, ensemble des éléments sélectionnés et on lui additionne un premier élément suivant une procédure qui sera précisée plus tard. Soit a_i un attribut et FD_{ai} la dimension fractale de $S \bigcup \{a_i\}$. Notons FD la dimension fractale de l'ensemble de données observées et $\overline{FD}_{ai}$, la dimension fractale

de ce même ensemble à l'exception de $S \bigcup \{a_i\}$. Alors, l'attribut considéré sera celui qui maximisera la quantité : $D = |(FD - \overline{FD}_{ai}) - FD_{ai}|$. Il sera ajouté à la sélection s'il permet d'accroître la quantité FD_{ai}. Sinon, il sera ajouté à $\overline{S}$, l'ensemble des attributs écartés. Notons que les éléments de $\overline{S}$ seront reconsidérés dès que l'ensemble S aura changé. On arrête la procédure de recherche quand tous les attributs auront été testés.

Afin d'illustrer ce mécanisme de sélection, considérons l'exemple synthétique suivant : on crée une base constituée de 2000 observations avec 4 variables observées X,Y,Z,T et une variable de classification nommée Classe. Notons que la relation $D < 2 * Log(N)$ est respectée pour cet exemple. Le nombre d'attributs D vaut cinq et $2 * Log(2000) > 6$. Par ailleurs, les cinq attributs ont été construits pour qu'ils vérifient les relations : $Y = 25 * X + 12$, $Z = Y + 150$,$T = Y$ et $Classe = Z - T + Y$. Notre procédure de sélection nous a retournée une dimension fractale égale à 0.9897105. En arrondissant à l'entier supérieur, on obtient 1. Cela signifie que cet ensemble de données nécessite un seul attribut représentatif. L'algorithme nous a retourné la valeur 1 (Select = 1), donc seul la variable X est nécessaire pour représenter cet ensemble. Cela est tout à fait valable, puisque tous les autres attributs lui sont corrélés linéairement. L'exemple suivant illustre ce processus de sélection.

$$> Selection(data, 5, 5)$$

$$Dimension fractale - -- > \mathbf{0.9897105}$$

Début de la phase de sélection du premier attribut

$$1 - -- > 0.001919470 - - - - -- > 0.9897105 - - - - - - > \mathbf{0.987791}$$

$$2 - -- > 0 - - - - -- > 0.99163 - - - - - - > 0.99163$$

$$3 - -- > 0 - - - - -- > 0.99163 - - - - - - > 0.99163$$

$$4 - -- > 0 - - - - -- > 0.99163 - - - - - - > 0.99163$$

$$5 - -- > 0 - - - - -- > 0.99163 - - - - - - > 0.99163$$

$$Select = \mathbf{1} - -- > Cumul = 0.9897105$$

Fin de la phase de sélection du premier attribut

$$2 - -- > 0.9897105 - - - - - - > 0.001919470 - - - - - -- > 0.987791$$

$$3 - -- > 0.9897105 - - - - - - > 0.001919470 - - - - - -- > 0.987791$$

$$4 - -- > 0.9897105 - - - - - - > 0.001919470 - - - - - -- > 0.987791$$

élement supprimé : 4

$2 - -- > 0.9897105 - - - - - - > 0.001919470 - - - - - - -- > 0.987791$

$3 - -- > 0.9897105 - - - - - - > 0.001919470 - - - - - - -- > 0.987791$

élement supprimé : 3

$2 - -- > 0.9897105 - - - - - - > 0.001919470 - - - - - - -- > 0.987791$

élement supprimé : 2

$Select = \mathbf{1}$

Sur la base d'une procédure qui sera illustrée plus tard dans ce chapitre, le premier attribut sélectionné dans l'exemple ci-dessus est celui qui a généré entre le début de la phase de sélection du premier attribut et la fin de cette phase, la plus petite valeur parmi la liste des valeurs obtenues dans la colonne quatre de cet exemple. Comme nous pouvons le constater, c'est l'attribut 1 qui répond à cette condition. Le reste de la procédure de sélection se poursuit en essayant dans chaque étape de l'algorithme d'ajouter un attribut à la sélection parmi l'ensemble des attributs qui restent à tester. Cependant, aucun d'eux n'a permis d'augmenter la valeur de la variable Cumul représentant la dimension fractale de l'ensemble des attributs sélectionnés. D'où la suppression des attributs 4,3 et 2 dans la suite du programme.

Dans ce même contexte, on a créé une deuxième base contenant le même nombre d'observations que celles utilisées précédemment, mais avec six attributs X,Y,Z,T,U au lieu de cinq, dont un attribut de classification. Cependant, on a modifié le type de corrélations entre les attributs : $Y = 25{*}X{+}12$, $Z = Log(Y){+}35{*}X$, $U = 25{*}X^2{+}0.5{*}Y{+}17$, $Classe = Y - T + X$. On peut constater que la relation entre le nombre d'observations et le nombre d'attributs reste vérifiée. Pour cet exemple, la dimension fractale est de 1.767979. En arrondissant à l'entier supérieur, on obtient deux. Cela signifie qu'on a besoin de deux attributs pour représenter ces données et non pas cinq. De plus, notre algorithme de sélection nous a retourné $\{T, Z\}$ comme attributs sélectionnés. Donc la dimension fractale est capable de détecter les relations de types linéaires, polynomiales et non polynomiales qui peuvent exister entre les attributs. L'exemple qui suit illustre cette procédure de sélection.

$$Selection(data, 6, 6)$$

$$Dimension fractale - -- > \mathbf{1.767979}$$

$$1 - -- > 0 - - - - -- > 0.9910232 - - - - - - > 0.9910232$$

$$2 - -- > 0.06712857 - - - - -- > 0.9910232 - - - - - - > 0.9238946$$

$$3 - -- > 0.1216356 - - - - -- > 0.9867158 - - - - - - > 0.8650801$$

$$4 - -- > 0.2234666 - - - - -- > 0.9926992 - - - - - - > \mathbf{0.7692326}$$

$$5 - -- > 0 - - - - -- > 0.9603952 - - - - - - > 0.9603952$$

$$6 - -- > 0.1167089 - - - - -- > 0.9544414 - - - - - - > 0.8377325$$

$$Select = \mathbf{4} - -- > Cumul = 1.640985$$

$$1 - -- > 1.411336 - - - - - - > 0.2234666 - - - - - -- > 1.187869$$

$$2 - -- > 1.835333 - - - - - - > 0.2495706 - - - - - -- > 1.585763$$

$$3 - -- > 1.923007 - - - - - - > 0.1516130 - - - - - -- > \mathbf{1.771394}$$

$$5 - -- > 1.374993 - - - - - - > 0.2234666 - - - - - -- > 1.151527$$

$$Select = \mathbf{4,3} - -- > Cumul = \mathbf{1.923007}$$

$$1 - -- > 1.923007 - - - - - - > 0.1516130 - - - - - -- > 1.771394$$

$$2 - -- > 1.884688 - - - - - - > 0.6974534 - - - - - -- > 1.187234$$

$$5 - -- > 1.923007 - - - - - - > 0.1542099 - - - - - -- > 1.768797$$

élement supprimé : 1

$$2 - -- > 1.884688 - - - - - - > 0.6974534 - - - - - -- > 1.187234$$

$$5 - -- > 1.923007 - - - - - - > 0.1542099 - - - - - -- > 1.768797$$

élement supprimé : 5

$$2 - -- > 1.884688 - - - - - - > 0.6974534 - - - - - -- > 1.187234$$

élement supprimé : 2

$$Select = \mathbf{4,3}$$

Suivant le même principe que celui utilisé dans l'exemple précédent, c'est l'attribut 4 qui est sélectionné en premier. Dans les étapes suivantes de l'algorithme et en se basant sur une procédure que nous décrirons plus tard dans ce chapitre, c'est l'attribut 3 qui va être ajouté à la sélection, alors que les attributs 1,5 et 2 vont être écartés.

Poursuivons notre démarche et rappelons que l'approche utilisée se base sur une méthode de sélection de type ajout. Il a été montré que cette dernière sélectionne moins d'attributs et est parfois plus performante que l'approche par retrait [19]. Cependant, au cours de la procédure de sélection, on peut être amené à écarter un attribut peu pertinent dans l'association analysée. Toutefois, rien ne prouve qu'il le sera encore lorsque la sélection aura changé. Nous avons alors décidé de réévaluer les attributs précédemment écartés et ce dès que la sélection aura évolué afin de voir si cela pourrait accroître l'efficacité de notre algorithme sans pour autant accroître considérablement sa complexité temporelle. L'exemple suivant est présenté pour illustrer cette procédure (Tab. 5.1). Il traite le cas d'une base nommée Cmc avec 1473 observations et 10 variables observées dont une variable de classification.

Exemple qui illustre la sélection des deux premiers attributs N_i **: Ordre de sélection** N_s **: Attribut non sélectionné**				
Attribut	FD_{ai}	$FD_{ai} - \overline{FD_{ai}}$	D	Ordre / État
1	0.01342594	0.4864523000	**0.4730264000**	N_2
2	0.10264060	2.652239e-17	0.1026406000	Ns
3	0.06803028	6.594231e-18	0.0680302800	Ns
4	0.29243850	0.3224895000	**0.0300509600**	N_1
5	0.00000000	7.006371e-18	7.006371e-18	Ns
6	0.00000000	2.351619e-17	2.351619e-17	Ns
7	0.05241344	4.58687e-17	0.0524134400	Ns
8	0.14224930	4.480198e-17	0.1422493000	Ns
9	0.00000000	4.788091e-19	4.788091e-19	Ns
10	0.06670342	3.011042e-17	0.0667034200	Classe

TAB. 5.1 – Exemple de récupération d'un attribut

$> Selection(data, 10, 10, 4) : 2.321290$

$Select = 4 - -- > Cumul = 0.3224895$

$Select = 4, 1 - -- > Cumul = 0.8369699$

élement supprimé : 9

élement supprimé : 5

élement supprimé : 3

$Select = 4, 1, 7 - -- > Cumul = 0.973059$

élement supprimé : 9

élement supprimé : 5

$Select = 4, 1, 7, 3 - -- > Cumul = 1.637429$

élement supprimé : 9

Select = 4,1,7,3,5 : Cumul = 1.822438

élement supprimé : 9

élement supprimé : 6

$Select = 4, 1, 7, 3, 5, 2 - -- > Cumul = 2.080975$

élement supprimé : 9

élement supprimé : 6

élement supprimé : 8

$Select = 4, 1, 7, 3, 5, 2 - -- > Cumul = 2.254586$

Exemple de récupération d'un attribut jugé non pertinent.

Dans cette exemple, ce sont les attributs 4 et 1 qui sont sélectionnés en premier. Ensuite l'attribut 5 est porté candidat pour être ajouté à la sélection. Seulement, ce dernier ne permet pas d'accroître la valeur de la variable *Cumul* qui représente la dimension fractale de l'ensemble des attributs sélectionnés. Alors, on supprime l'attribut 5. On poursuit la procédure de sélection et c'est au tour de l'attribut 7 d'être ajouté au attributs 4 et 1. Dans un tel cas, on doit réinjecter dans l'ensemble des attributs à considérer ceux précédemment écartés. Donc, l'attribut 5 va être porté pour la deuxième fois candidat à la sélection et comme on peut le constater il sera encore écarté. Cependant, cet attribut sera ajouté à l'ensemble des attributs pertinents une fois associé aux attributs 4,1,7 et 3. Ceci illustre l'idée selon laquelle un attribut peut paraître non pertinent dans une association d'attributs et le devenir dans une autre.

Cependant, cette façon de faire va t-elle améliorer la qualité des réductions ? Pour apporter une réponse à cette question, nous avons comparé les deux réductions, c-à-d celle obtenue sans et avec récupération. Quand aucune récupération n'est appliquée, la réduction obtenue est $\{4, 1, 7, 6, 8, 2\}$, mais avec récupération on obtient $\{4, 1, 7, 3, 5, 2\}$. Afin de juger de la pertinence de ces réductions, on a utilisé l'algorithme C4.5 de la plate-forme Weka [50]. À l'inverse de ce à quoi on pouvait s'attendre, la précision est légèrement meilleure sans récupération. Nous avons relevé les erreurs de classification avant réduction, avec récupération et sans récupération : 46.78%, 46.71% et 45.96%. Notons cependant que les deux réductions obtenues ont la même taille. Alors, doit-on récupérer les attributs écartés au cours de la procédure de sélection ? Pour mieux cerner la nature de ce problème, nous avons appliqué ce même protocole aux dix bases ayant servi dans la comparaison des méthodes de sélection, (cf. chapitre 4). Les résultats de la comparaison se trouvent résumés dans Tab 5.2. L'analyse de ces résultats montre que la procédure de récupération des attributs écartés est inefficace. Comme on peut le constater, dans le pire des cas, on obtient le même taux d'erreur que lorsqu'on n'applique pas de récupération et même dans certains cas un taux légérement meilleur. De plus, comme on doit retester les attributs précedemment écartés, cela induit un temps algorithmique supplémentaire. Notons cependant que les tailles des réductions générées sont les mêmes. Nous avons donc décidé de ne pas reconsidérer les attributs écartés dans notre approche de RDD.

Base	Err. avant réduction en %	Sans récupération		Avec récupération	
		Taille réduction	Err. classification en %	Taille réduction	Err. classification en %
Zoo-Data	8	5/17	**12.87**	5/17	**13.86**
Ecoli	15.77	5/8	**16.96**	5/8	**16.96**
Cave	4.90	4/6	**5.38**	4/6	**5.38**
Diabete	25.78	3/9	**32.84**	3/9	**32.84**
Abalone	79.63	4/9	**77.62**	4/9	**77.62**
Yeast	44.0027	6/9	**43.67**	6/9	**43.67**
Magic	14.94	5/11	**17.65**	5/11	**17.65**
Heart	20.79	3/14	**30.69**	3/14	**29.80**
Segment	2.86	5/19	**12.86**	5/19	**12.86**
Cmc	46.78	6/10	**45.96**	6/10	**46.71**

TAB. 5.2 – Étude comparative - Procédure de récupération.

Nous avons mis en gras les erreurs de classification obtenues sans et avec récupération. On constate qu'elles sont pratiquement les mêmes. Donc, la procédure de récupération est inutile.

Le dernier point qu'on va aborder pour finaliser notre approche consiste à déterminer l'impact du premier attribut sélectionné sur la qualité de la réduction obtenue. Notons que dans une approche par ajout, on débute avec un ensemble vide et on ajoute au fur et à mesure les attributs qu'on juge pertinents. Nous avons alors remarqué, lors de nos tests, que le choix du premier attribut influence considérablement la suite de la procédure de sélection, en ce sens que le sous-ensemble d'attributs obtenu après réduction sera fonction de ce choix. D'où la nécessité de faire le bon choix. La question est sur quelle base doit-on choisir cet attribut ? À cette fin, nous avons retenu quatre critères d'évaluation et nous avons développé quatre variantes de notre algorithme basées sur ces critères. La première sélectionne le premier attribut sur la base de l'entropie, la seconde utilise le coefficient de Gini [25] et les deux autres sont basées sur la dimension fractale. Pour déterminer le critère le mieux adapté à notre approche de RDD, nous avons réalisé une série de tests utilisant les dix bases ayant servi dans la comparaison des méthodes de sélection, (cf. chapitre 4). Nous donnerons la version complète de la première variante de notre algorithme, suivie d'un tableau comparatif des résultats obtenus sur l'échantillon de ces dix bases. Pour les autres variantes, on précisera la nature de la modification à apporter à la variante 1 de notre algorithme, suivie là aussi, de la même série de tests. Nous concluons nos investigations par une analyse permettant de choisir le critère à retenir.

Nous allons maintenant présenter notre algorithme version entropie. Avant de l'énoncer, précisons la signification de quelques unes des variables manipulées par cet algorithme. Dans cette version et les suivantes, P : représente l'ensemble des attributs à tester et DC(P) : la dimension de corrélation de cet ensemble (cf. annexe). La variable *Classe* représente l'attribut de classification, ST : la liste des attributs qui restent à tester, SP : la liste des attributs à écarter, S : l'ensemble des attributs sélectionnés classés par ordre de leur importance et dim : la variable contenant la dimension de corrélation du sous-ensemble testé.

Algorithm 4: Algorithme de Sélection - Version Entropie -

Data: P : ensemble des attributs à traiter et des données d'entrée

ST : la liste des attributs qui restent à tester

SP : la liste des attributs à écarter.

Result: Liste des attributs sélectionnés classés par ordre de leur importance

1. Calculer la dimension de corrélation D de l'ensemble de données (cf. annexe) ;

2. Initialisation : S = {} ; $ST = P$; SP = {} ; dim = 0;

foreach $a_i \in P$ **do**
  X_i = ENT($\{a_i, Classe\}$) /* ENT(Y) = entropie de l'ensemble Y */ ;
end

3. Classer les X_i par ordre décroissant;

4. Soit a_i l'attribut correspondant à la plus petite valeur de X_i;

5. $S = \{a_i\}$;

6. dim = DC(S) /* Dimension de corrélation de l'ensemble S (cf. annexe) */ ;

7. $ST = P - S$;

while *(*$ST \neq \{\}$*)* **do**
  foreach $a_i \in ST$ **do**
    $X_i = DC(S \cup \{a_i\})$;
    $Y_i = DC(P - S \cup \{a_i\})$;
    $D_i = |(D - Y_i) - X_i|$;
    $D = Max(D_i)$;
  end
  Soit a_i l'attribut qui maximise la quantité D_i;
  if $(dim \geq DC(S \cup \{a_i\})$ **then**
    8. $SP = SP \cup \{a_i\}$;
    9. $ST = P - S - SP$;
  else
    10. $S = S \cup \{a_i\}$;
    11. $dim = DC(S)$;
    12. $ST = P - S - SP$;

end

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Analyse des réductions générées**				
Base	Premier attribut sélectionné	Réduction	Erreur avant réduction	Erreur après réduction
Zoo-Data	9	$\{9, 16, 15, 6, 13\}$	8	17
Ecoli	4	$\{4, 6, 2, 1, 5\}$	15.77	18.15
Cave	5	$\{5, 1, 2, 4\}$	4.90	4.57
Diabete	5	$\{5, 3, 8\}$	25.78	31.12
Abalone	1	$\{1, 6, 2, 8\}$	79.63	77.62
Yeast	5	$\{5, 1, 4, 3, 8, 2\}$	44.0027	43.67
Magic	5	$\{5, 3\}$	14.94	57.98
Heart	6	$\{6, 5, 8, 4\}$	20.79	30.37
Segment	4	$\{4, 5, 7\}$	2.86	58.27
Cmc	9	$\{9, 1, 4, 3, 7, 6, 8\}$	46.78	48.47

TAB. 5.3 – Sélection du premier attribut sur la base de l'entropie (ALGSENT)

Notons que la seule différence entre les 4 versions de notre algorithme réside dans la façon de choisir le premier attribut. Donc, la méthode employée pour calculer la variable X_i. Dans ce qui va suivre, on précisera pour chacune des versions de notre algorithme uniquement ce calcul suivi des résultats des tests.

Débutons avec la version basée sur le coefficient de Gini. Alors, $X_i = \text{Gini}(\{a_i, Classe\})$. Cette valeur représente le coefficient de Gini pour l'ensemble réduit aux attributs a_i et $Classe$. De plus, par comparaison avec la version précédente de notre algorithme, les valeurs X_i devront être classées par ordre croissant au lieu de décroissant.

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Analyse des réductions générées**				
Base	Premier attribut sélectionné	Réduction	Erreur avant réduction	Erreur après réduction
Zoo-Data	13	$\{13, 16, 5, 11, 15\}$	8	25.74
Ecoli	6	$\{6, 2, 1, 5\}$	15.77	18.15
Cave	2	$\{2, 5, 3, 4\}$	4.90	5.39
Diabete	2	$\{2, 6, 3, 4, 8\}$	25.78	24.48
Abalone	8	$\{8, 7, 2\}$	79.63	79.53
Yeast	3	$\{3, 1, 4, 8, 2\}$	44.0027	43.94
Magic	9	$\{9, 6, 7, 8, 1\}$	14.94	17.82
Heart	13	$\{13, 5, 8, 4\}$	20.79	24.42
Segment	10	$\{10, 1, 14\}$	2.86	14.50
Cmc	4	$\{4, 1, 7, 6, 8, 2\}$	46.78	45.96

TAB. 5.4 – Sélection du premier attribut sur la base du coefficient de Gini (ALGSGINI)

Poursuivons maintenant avec l'algorithme basé sur la dimension fractale version 1. Dans ce cas, les changements à apporter à la version précédente sont : $X_i = \text{DC}(\{a_i\})$. Cette valeur représente la dimension de corrélation de l'ensemble réduit au singleton a_i et ensuite le classement des valeurs de la variable X_i ainsi calculées par ordre décroissant.

Les résultats obtenus sont les suivants :

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Analyse des réductions générées**				
Base	Premier attribut sélectionné	Réduction	Erreur avant réduction	Erreur après réduction
Zoo-Data	16	$\{16, 7, 6, 14, 1, 13\}$	8	11.88
Ecoli	2	$\{2, 7, 1, 5\}$	15.77	17.56
Cave	1	$\{1, 4, 2, 3\}$	4.90	5.63
Diabete	6	$\{6, 1, 4, 8, 3\}$	25.78	33.20
Abalone	7	$\{7, 8, 2\}$	79.63	79.53
Yeast	1	$\{1, 4, 3, 8, 2\}$	44.0027	43.94
Magic	10	$\{10, 6, 1, 8, 9\}$	14.94	17.66
Heart	8	$\{8, 5, 4\}$	20.79	30.69
Segment	1	$\{1, 9, 14\}$	2.86	15.80
Cmc	1	$\{1, 4, 7, 6, 8, 2\}$	46.78	45.96

TAB. 5.5 – Sélection du premier attribut sur la base de la dimension fractale ver.1 (ALGSDF1)

Poursuivons et précisons les modifications à apporter à la version précédente afin d'obtenir la version deux de notre algorithme dont la sélection du premier attribut est basée sur la dimension fractale. Dans ce cas, en plus de la variable X_i, on introduit deux autres variables : Y_i et D_i. Alors, les changements sont les suivants : X_i = D - DC(P - $\{a_i\}$), Y_i = DC($\{a_i\}$) et $D_i = X_i - Y_i$. Dans ce cas, le premier attribut sélectionné sera celui qui génère la plus petite valeur pour la variable D_i. Les résultats obtenus sont :

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Analyse des réductions générées**				
Base	Premier attribut sélectionné	Réduction	Erreur avant réduction	Erreur après réduction
Zoo-Data	1	$\{1, 7, 15, 13\}$	8	12.87
Ecoli	3	$\{3, 6, 2, 1, 5\}$	15.77	16.96
Cave	2	$\{2, 5, 3, 4\}$	4.90	5.39
Diabete	4	$\{4, 8, 6\}$	25.78	32.16
Abalone	1	$\{1, 6, 2, 8\}$	79.63	77.62
Yeast	5	$\{5, 1, 4, 3, 8, 2\}$	44.0027	43.67
Magic	6	$\{6, 10, 1, 8, 9\}$	14.94	17.66
Heart	4	$\{4, 5, 8\}$	20.79	30.69
Segment	6	$\{6, 14, 11, 1, 15\}$	2.86	12.86
Cmc	4	$\{4, 1, 7, 6, 8, 2\}$	46.78	45.96

TAB. 5.6 – Sélection du premier attribut sur la base de la dimension fractale ver.2 (ALGSDF2)

Afin de mieux synthétiser ces informations, on a procédé à un groupement des principaux résultats obtenus à l'aide des quatre versions de notre algorithme.

Synthèse **Étude réalisée sur un échantillon de dix bases de données réelles** **Lien entre méthode employée et premier attribut sélectionné**				
Base	Premier attribut basé entropie Err. après réduction (1)	Premier attribut basé Coef. Gini Err. après réduction	Premier attribut Dim. fractale ver.1 Err. après réduction	Premier attribut Dim. fractale ver. 2 Err. après réduction
Zoo-Data	9/17	13/25.74	16/11.88	1/12.87
Ecoli	4/18.15	6/18.15	2/17.56	3/16.96
Cave	5/4.75	2/5.39	1/5.63	2/5.39
Diabete	5/31.12	2/24.48	6/33.20	4/32.16
Abalone	1/77.62	8/79.53	1/79.53	1/77.62
Yeast	5/43.67	3/43.94	10/43.94	5/43.67
Magic	5/57.98	9/17.82	8/17.66	6/17.66
Heart	6/30.37	13/24.42	1/30.69	4/30.69
Segment	4/58.27	10/14.50	1/15.80	6/12.86
Cmc	9/48.47	4/45.96	1/45.96	4/45.96

TAB. 5.7 – Comparaison des méthodes de sélection du premier attribut

Nous avons utilisé une notation fractionnaire de la forme A/B pour indiquer les résultats de cette comparaison. A représente le premier attribut sélectionné et B le taux d'erreur de classification correspondant exprimé en (%) quand on utilise cette réduction comme élément représentatif des données analysées. Le tableau (Tab.5.7) ci-dessus met bien en évidence le lien qui existe entre le premier attribut sélectionné et la méthode employée pour cette sélection. On poursuit notre analyse par la comparaison des quatre versions de notre algorithme sur la base (Taille réduction) / (Err. de classification). Le résultat de cette analyse se trouve dans le tableau (Tab. 5.8).

Base Dimension Initiale	ALGSENT	ALGSGINI	ALGSDF1	ALGSDF2	ALGSVM	Dimension de départ / Erreur en %
Zoo-data (17)	5/17	5/25.74	6/11.88	4/12.87	8/2.97	17/8
Ecoli (8)	5/18.15	4/18.15	4/17.56	5/16.96	7/15.77	8/15.77
Cave (6)	4/4.57	4/5.39	4/5.63	4/5.39	4./4.57	6/4.90
Diabete (9)	3/31.12	5/24.48	5/33.20	3/32.16	6/24.35	9/25.78
Abalone (9)	4/77.62	3/79.53	3/79.53	4/77.62	7/79.22	9/79.63
Yeast (9)	6/43.67	5/43.94	5/43.94	6/43.67	6/44.81	9/44.0027
Magic (11)	2/57.98	5/17.82	5/17.66	5/17.66	7/14.45	11/14.94
Heart (14)	4/30.37	4/24.42	3/30.69	3/30.69	6/20.79	14/20.79
Segment (19)	3/58.27	3/14.50	3/15.80	5/12.86	6/4.11	19/2.86
Cmc (10)	7/48.47	6/45.96	6/45.96	6/45.96	5/44.74	10/46.78

TAB. 5.8 – Étude comparative de ALGSVM et des quatre variantes de notre algorithme

Nous avons utilisé une notation fractionnaire de la forme A/B pour indiquer les résultats de cette comparaison. A représente le nombre d'attributs sélectionnés et B le taux d'erreur de classification correspondant exprimé en (%) quand on utilise cette réduction comme élément représentatif des données analysées. On constate que c'est l'algorithme ALGSVM qui fournit les meilleures réductions du point de vue qualité de la précision, mais nécessite un plus grand nombre d'attributs. Nous allons donc l'utiliser comme référence pour analyser et comparer les résultats obtenus par les différentes versions de notre algorithme. Une première remarque nous amène à considérer la base Zoo-Data qui est formée de 17 attributs dont un attribut de classification. L'algorithme C4.5 de la plate-forme Weka nous révèle une erreur de classification de l'ordre de 8% avant réduction et de 2.97 % après réduction, d'où une nette amélioration du taux de classification. Ceci confirme l'idée selon laquelle, dans certains cas, non seulement la réduction réduit le temps de traitement des données, mais peut aussi améliorer la précision.

Une autre remarque nous parvient de la base Abalone. L'algorithme de référence indique une erreur de classification de l'ordre de 79% avec 7 attributs retenus sur 8 pour ALGSVM, alors qu'elle est de 77% avec seulement 4 attributs gardés sur les 8 pour la première et la quatrième variantes de notre algorithme. Donc, la présence de certains attributs au sein des donnéees non seulement n'apporte aucun supplément d'information, mais bien au contraire amène du bruit. Notons la capacité de l'algorithme ALGSDF2 à écarter de tels attributs. Nous pouvons constater aussi le lien étroit entre les réductions obtenues et le premier attribut choisi (cf. Tab.5.3 à 5.6). Comme on peut le voir, les réductions générées ont des tailles différentes et produisent des erreurs de classification différentes. Notons enfin que, parmi l'ensemble des quatre versions d'algorithme présentées, seul ALGSDF2 possède la capacité de générer des réductions donnant le meilleur compromis entre la taille des réductions et les taux d'erreurs de classifications sous-jacents. C'est donc l'algorithme ALGSDF2 qui sera retenu pour notre approche de réduction de la dimensionnalité des données.

5.3 Approche retenue

Notre méthode de réduction de la dimensionnalité des données est basée sur la sélection d'attributs pertinents de type ajout. Elle est fondée essentiellement sur la dimension fractale. Cette dernière permet d'obtenir aussi bien des réductions de type relatif[1] que de type absolu.[2]. On a mentionné précédemment le fait que la dimension fractale était sensible à l'importance qu'a un attribut au sein des données. Notons que plus un attribut est pertinent pour la représentation des données, plus est grande la variation que subira la dimension fractale si on décide d'écarter ce dernier des données analysées.

Nous débutons notre algorithme de réduction par le calcul de la dimension fractale de l'ensemble des données nommée FD. La valeur ainsi obtenue servira de critère d'évaluation de la pertinence d'un attribut quand ce dernier est considéré seul et quand il est associé à d'autres attributs.

[1] Au sein des données, il existe un attribut de classification.

[2] Sans lien avec un attribut de classification.

Après initialisation de l'ensemble S (ensemble des attributs sélectionnés) à l'ensemble vide, on calcule pour chaque attribut de l'ensemble P, représentant l'ensemble des attributs, la quantité FD_{ai} égale à la dimension fractale de $S \bigcup \{a_i\}$ et $\overline{FD}_{ai}$, la dimension fractale de l'ensemble $P - S \bigcup \{a_i\}$. Ainsi, au sein des attributs, celui qui minimise la quantité $\left|FD_{ai} - \overline{FD}_{ai}\right|$, sera le premier attribut sélectionné. Soit *dim*, la variable représentant la dimension fractale de l'ensemble S réduit au singleton $\{a_i\}$. On poursuit la procédure en calculant pour tous les attributs a_i qui restent à évaluer, les dimensions fractales X_i de $S \bigcup \{a_i\}$ et par Y_i celle de l'ensemble $P - S \bigcup \{a_i\}$. L'attribut a_i qui sera considéré pour la sélection, sera celui qui maximisera la quantité $|(FD - Y_i) - X_i|$. L'ajout de ce dernier deviendra effectif si la nouvelle valeur de la variable *dim* égale à la dimension fractale de $S \bigcup \{a_i\}$ est supérieure à l'ancienne valeur. Sinon, l'attribut sera retiré de la liste des attributs à considérer. La procédure de sélection se poursuit jusqu'à ce qu'il n'y ait plus d'attribut à tester.

Après avoir présenté notre algorithme de réduction de la dimensionnalité, nous allons le situer par rapport à l'approche de Faloustos. La comparaison se fera sur la base des critères suivants : type de parcours, critère permettant d'évaluer l'importance d'un attribut, condition d'arrêt de l'algorithme, complexité temporelle et limites relevées dans les deux approches. Cependant, avant d'entamer cette comparaison, calculons la complexité de notre algorithme.

Soit N le nombre d'observations et D le nombre d'attributs observés. La première phase de notre algorithme où on calcule la dimension fractale de l'ensemble des données nécessite un temps de traitement linéaire *O(N)* [48]. Comme il y a D attributs et que le temps nécessaire pour choisir un attribut est $2 * (D - 1) * N$ (première boucle où on teste l'ensemble des attributs, afin de choisir le premier), la complexité est aussi *O(N)*. Cependant, après chaque étape de l'algorithme, il va rester $(D - 1)$ attributs à évaluer (la boucle Tant que). Alors, le temps nécessaire pour évaluer l'ensemble des attributs est donc : $\frac{D*(D-1)*N}{2}$. D'où une complexité temporelle quadratique par rapport à l'ensemble des attributs observés.

Tableau comparatif					
Méthode utilisée	Type de parcours	Critère utilisé pour évaluer l'importance d'un attribut	Critère d'arrêt	Complexité temporelle	Limites
Notre approche	Ajout	La dimension fractale	Tous les attributs ont été testés	$O(D^2 * N)$	- Les résultats se dégradent dans des problèmes où on dispose de peu d'observations comparativement au nombre de variables observées - Dégradation de la précision des réductions en présence du bruit. [30]
Approche Faloustos	Retrait	La dimension fractale	Tous les attributs ont été testés	$O(D^2 * N)$	- Seuil fixé pour évaluer les attributs - Ne prend pas en considération les associations d'attributs. - Tous les attributs de la réduction sont considérés comme ayant la même importance. - Performances médiocres quand le nombre d'observations ou d'attributs est important.

Tab. 5.9 – Étude comparative - ALGSDF2 vs ALGFAL

Notons que les deux méthodes utilisent la dimension fractale pour évaluer la pertinence d'un attribut au sein des données. Cependant, une différence majeure les distingue. Faloustos évalue les attributs individuellement. Dans son approche de réduction, on ne connaît pas les répercutions sur les données causée par la suppression d'attributs associés. Un attribut qu'on vient d'écarter peut devenir important après la suppression d'un autre puisque son importance peut être camouflée par ce dernier. C'est justement à ce niveau que nous avons apporté le changement à son approche. Dans notre démarche, si S représente l'ensemble des attributs sélectionnées, alors on suggère d'ajouter à S l'attribut a_i qui une fois supprimé des données en association avec S produira la plus grande variation de la dimension fractale par rapport à sa valeur initiale, lorsque tous les attributs étaient réunis. Cependant, cet ajout ne prendra effet que si l'attribut a_i une fois associé à l'ensemble S permettra d'obtenir une dimension fractale supérieure à celle obtenue quand l'attribut n'était pas considéré dans cette association.

Chapitre 6

Implémentation

6.1 Validation de l'approche proposée

Pour valider notre approche tant au niveau de la qualité de sélection des attributs qu'au niveau du coût d'exécution, nous allons examiner les résultats obtenus par l'algorithme basé sur la dimension fractale ver 2 (ALGSDF2) et les comparer à ceux générés par les algorithmes de Faloustos (ALGFAL) et l'algorithme basé sur SVM (ALGSVM). À cette fin, nous avons procédé à une douzaine d'expériences sur des bases de données dont la majorité d'entre elles proviennent du répertoire UCI [38]. Nous avons tenu à avoir des bases dont les caractéristiques nous permettent de tester l'algorithme ALGSDF2 dans des cas limites. Voici les cas considérés.

Premier cas : peu d'observations par rapport au nombre d'attributs observés. Cela nous permet de tester la fiabilité de notre algorithme dans le cas où la relation $D < 2 * Log(N)$, n'est pas vérifiée où D est la dimension de l'espace d'observations et N le nombre d'observations. Rappelons que si le nombre d'observations par rapport au nombre d'attributs observés vérifie cette relation, la valeur de la dimension fractale obtenue ajuste mieux la dimension intrinsèque des données.

Deuxième cas : Nous allons nous servir de deux bases de données synthétiques constituées d'attributs dont certains sont indépendants et d'autres non. Dans ce cas, nous cherchons à montrer la capacité de notre algorithme à déceler ce type de relations, contrairement à l'algorithme ALGFAL.

Dans l'ensemble de ces expériences, nous avons cherché à connaître les réductions obtenues par l'application des trois méthodes citées précédemment. Ensuite, pour juger et comparer la qualité des réductions obtenues, nous avons utilisé l'algorithme C4.5 afin d'évaluer le taux de mauvais classement obtenu suite à l'emploi des réductions générées par chacune des trois méthodes (Tab.6.2). Rappelons que le but principal pour ce type d'algorithme est de modéliser la relation qui existe entre les attributs observés et l'attribut à expliquer, c-à-d que l'algorithme se base sur des exemples déjà classés pour déterminer un modèle de classification. Notons que l'algorithme C4.5 est une extension et une amélioration de l'algorithme ID3 lequel produit un arbre de décision qui servira à classer les nouvelles observations.

Base	Condition $D < 2 * Log(N)$ satisfaite à	ALGSDF2 (1) % Err.	ALGFAL (2) % Err.	ALGSVM (3) % Err.	Err. avant Réduction en %	Meilleur taux de réduction	Observations
Zoo-Data	23.5%	*12.87*	21.78	**2.97**	8	75%(1)	5 %
Ecoli	62.5%	*16.96*	17.56	**15.77**	15.77	38%(1)	1%
Cave	100%	5.39	*4.90*	**4.57**	4.90	34%(1)	< 1%
Diabete	55.55%	32.16	*25.52*	**24.35**	25.78	67%(1)	7 %
Abalone	77.77%	**77.62**	*78.21*	79.22	79.63	56%(1)	- 2 %
Yeast	66.66%	**43.67**	53.17	*44.81*	44.0027	56%(2)	- 1 %
Magic	72.72%	17.66	*16.79*	**14.45**	14.94	55%(1)	3%
Heart	35.71%	*30.69*	31.35	**20.79**	20.79	79%(1)	10 %
Segment	36.84%	12.86	*4.72*	**4.11**	2.86	74%(1)	10 %
Cmc	60%	45.96	*44.94*	**44.74**	46.78	50%(3)	1%

TAB. 6.1 – Étude comparative : ALGSDF2/ALGFAL/ALGSVM

Afin de mieux faire ressortir les résultats obtenus par ces trois algorithmes, nous avons utilisé une légende dont voici la signification. Le caractère gras est utilisé pour illustrer le meilleur pourcentage d'erreur, le normal illustre le plus mauvais et le caractère italique un résultat intermédiaire.

Ces résultats mettent en évidence une prédominance de l'algorithme ALGSVM. Les plus faibles taux d'erreurs de classification ont été obtenus à l'aide de ce dernier. Toutefois, comme nous l'avons déjà mentionné, cet algorithme souffre d'une complexité algorithmique élevée, laquelle est quadratique par rapport au nombre d'observations. Il faut ajouter à cela le temps pour trouver le noyau adapté aux données analysées. Cependant, nous l'utiliserons comme référence pour évaluer la qualité des réductions générées par l'algorithme ALGSDF2. Les résultats obtenus nous amènent à faire les remarques suivantes : la première remarque concerne les écarts entre les taux d'erreurs obtenus par l'application de ALGSDF2 avec ceux de ALGSVM. Dans le pire des cas, ils sont de 10 %. Ce type de résultat peut s'expliquer probablement par l'écart important qui sépare la valeur requise pour obtenir une bonne estimation de la dimension réelle des données et le nombre d'attributs utilisés par les bases testées. Cette remarque concerne les bases *Zoo-Data*, *Diabete*, *Heart* et *Segment*. La deuxième remarque concerne la taille des réductions. Les plus petites tailles ont été obtenues dans la plupart des cas à l'aide de la variante de notre algorithme ALGSDF2 avec d'excellents taux d'erreur. Comme on peut le constater, dans le pire des cas, il est de 10 % supérieur à sa valeur initiale, c-à-d quand tous les attributs sont réunis. Notons aussi qu'à l'aide de l'algorithme ALGSDF2, nous avons pu atteindre des taux de réduction variant entre 34 % et 80 %.

Le deuxième point qu'on voulait tester concerne la capacité d'un algorithme à déceler différents types de corrélations entre les attributs, et ce pour les algorithmes ALGSDF2 et ALGFAL. Nous avons alors créé deux bases synthétiques constituées chacune de 2000 observations et respectivement de cinq et six attributs dont un attribut de classification dans chaque cas. Le but est de montrer la capacité de notre algorithme à générer des réductions là où l'algorithme de Faloustos n'étais pas en mesure de le faire.

Sur cet exemple, nous pouvons constater que la relation qui lie le nombre d'observations au nombre d'attributs est vérifiée. Pour la première base, les attributs X, Y, Z, T, $Class$ vérifient les relations suivantes : $Y = 25X + 12$, $Z = Y + 150$, $T = Y$ et $Class = Z - T + Y$. Alors que pour la deuxième, les relations entre les attributs X, Y, Z, T, V, $Class$ sont : $Y = 25X + 12$, $Z = Log(Y) + 35X$, $V = (25X^2 + 0.5Y + 17)/37$ et $Class = Z - Y + X$.

En utilisant l'algorithme ALGFAL sur ces deux bases, nous avons obtenu les résultats suivants : aucune réduction n'a pu être générée pour la première base. Quant à la deuxième base, nous avons obtenu le sous-ensemble $\{Y, Z, T\}$ comme réduction. Cependant, avec notre algorithme ALGSDF2 les résultats sont respectivement : $\{X\}$ pour la première base et $\{T, Z\}$ pour la deuxième. Or, comme on peut le constater, tous les attributs de la première base sont corrélés linéairement à l'attribut X. Donc, théoriquement, $\{X\}$ est la seule réduction possible pour cette base. Ceci vient confirmer la validité du premier résultat obtenu par ALGSDF2. Analysons maintenant la réduction obtenue par l'algorithme ALGFAL sur la deuxième base. Comme nous pouvons le voir, l'attribut X est corrélé avec l'attribut Z et Y. De même que l'attribut Y est corrélé avec Z. Donc, dans une réduction, nous ne pouvons pas avoir les attributs Y et Z ensemble. Or, ces deux attributs sont présents dans la réductions générée par ALGFAL, contrairement à ce qu'on a obtenu comme réduction en appliquant l'algorithme ALGSDF2.

Chapitre 7

Conclusion

Dans ce mémoire, on s'est fixé deux objectifs. Le premier est l'étude comparative des principales méthodes utilisées lors de la réduction de la dimensionnalité des données. Le second concerne la mise au point d'une approche permettant de s'affranchir de certaines limites bien connues rencontrées lors de l'utilisation des méthodes de réduction de la dimensionnalité des données (RDD).

Après avoir présenté les principales techniques de réduction de la dimensionnalité (régression multiple, approches par extraction et approches par sélection), nous avons poursuivi notre étude en faisant une analyse critique et comparative des principales techniques. Certaines de ces techniques souffrent d'un temps algorithmique élevé, ce qui les rend inutilisables avec des données à haute dimensionnalité. D'autres ne sont tout simplement pas en mesure de déceler certains types de relations qui peuvent exister entre les attributs.

Nous avons ensuite présenté et validé une approche de RDD basée sur la sélection d'attributs pertinents dans laquelle nous avons mis en évidence l'importance de bien choisir le premier attribut lors du processus de sélection et son influence sur le reste de ce processus. Nous avons montré aussi qu'il n'était pas pertinent de récupérer les attributs écartés lors du processus de sélection et qu'il était important d'avoir un équilibre entre le nombre d'observations et le nombre de variables observées. Finalement, nous

avons proposé un algorithme basé sur la dimension fractale. Ce dernier est en mesure de détecter des corrélations cachées de différentes natures qui peuvent exister entre les attributs. Il peut aussi générer une réduction dans un temps quadratique par rapport au nombre d'attributs avec une perte d'information ne dépassant généralement pas sa valeur initiale augmentée de 10%. Pour valider notre approche, nous avons procédé à une série d'expériences sur des données synthétiques et réelles. Nous avons confronté nos résultats avec ceux obtenus par quatre autres algorithmes parmi les plus connus dans le domaine de la RDD par sélection d'attributs. Notre approche nous a permis d'obtenir dans la majorité des cas le meilleur rapport *taille réduction / taux d'erreur de classification.* Cependant, plusieurs améliorations peuvent être envisagées :

1. Dans le cas où la relation $D < 2*Log(N)$ n'est pas vérifiée, nous pouvons considérer les valeurs prises par chaque attribut comme étant une série chronologique et l'approcher par une loi de probabilité. Le but étant de générer des valeurs pour chacun de ces attributs afin de rendre cette relation valide et permettre ainsi l'exécution de notre algorithme dans des conditions optimales.
2. Après avoir calculé les PD_i, dimensions partielles pour chaque attribut, il s'agit de trouver une relation permettant de calculer la dimension de corrélation partielle de n'importe quelle combinaison d'attributs sur la base des PD_i. Le but étant de ramener la complexité temporelle de l'algorithme, de quadratique à linéaire par rapport au nombre d'attributs.
3. Faire des études plus approfondies permettant de fixer judicieusement le premier attribut.

Annexe A

Notions de topologie

Cette annexe rappelle quelques définitions nécessaires à la compréhension de la méthode de réduction de la dimensionnalité des données proposée dans ce mémoire, basée sur la dimension de corrélation et plus généralement la dimension fractale.

La topologie est l'étude mathématique des propriétés des objets préservées malgré les déformations, les torsions ou les étirements. Plus formellement, la topologie est l'étude des espaces topologiques.

Définition 1 : Soit X un ensemble, on appelle topologie sur X, une famille T de parties de X appelées ouverts, telles que :

- Toute union d'ouverts est un ouvert.
- Toute intersection d'ouverts est un ouvert.
- $\emptyset$ et X sont des ouverts.

Définition 2 : Une topologie T est dite séparée si : $a \neq b \Rightarrow \exists$ U , V $\in T$; $a \in U$, $b \in V$, $U \bigcap V = \emptyset$

Définition 3 : Une distance (ou métrique) sur X est une application
$d : X \to R^+$ tel que :
– d(x,y) = d(y,x) $\forall x, y \in X$
– d(x,y) = 0 $\Leftrightarrow x = y$
– d(x,z) $\leq d(x, y) + d(y, z)$ $\forall x, y, z \in X$

Définition 4 : Un espace métrique est un ensemble au sein duquel une notion de distance entre les éléments de l'ensemble est définie. C'est un cas particulier d'espace topologique.

Définition 5 : Soit (X,d) un espace métrique, $a \in X$, $r > 0$. Une boule fermée, centrée en a est de rayon r, est définie par :
$\bar{A}(a, r) = \{x; d(x, a) \leq r\}$, et une boule ouverte de centre a et de rayon r par : $A(a, r) = \{x; d(x, a) < r\}$.

Définition 6 : Une partie U de E (espace métrique) est un ouvert si : $U = \emptyset$ ou si $\forall x \in U$, $\exists r > 0$ tel A(x,r) $\subset U$.

Définition 7 : Soit (E,d) un espace métrique. On appelle recouvrement de E, un ensemble de partie de E tel que tout point de E appartient à au moins l'une de ces parties. Notons qu'un sous-recouvrement d'un recouvrement est celui formé de parties appartenant au premier recouvrement. Un recouvrement est considéré fini s'il contient un nombre fini de parties de E, et est ouvert si toutes les parties de ce recouvrement sont des ouverts de E.

Définition 8 : Un espace métrique (E,d) est compact si de tout recouvrement ouvert de E, on peut extraire un sous-recouvrement ouvert fini.

Définition 9 : Soit E un espace métrique. E est dit précompact si $\forall \epsilon > 0$, on peut recouvrir E par un nombre fini de boules ouvertes de rayon ϵ.

Définition 10 : Soit A une partie non vide d'un espace (E, d). Le nombre $\lambda(A) = max\ d(x, y); (x, y) \in A^2$ représente le diamètre de A. L'ensemble A sera dit borné si son diamètre est fini.

Nous somme maintenant en mesure d'introduire la notion de dimension de boîte (ou dimension métrique). Cette dimension sera utilisée par la suite pour évaluer la pertinence d'un attribut au sein d'un ensemble de données.

Pour chaque $\epsilon > 0$, E peut être recouvert par un nombre fini de boules fermées de rayon ϵ centrées ou non de E. On s'intéresse au plus petit nombre $N_E(\epsilon)$ de telles boules recouvrant E quand ϵ tend vers 0. Ce nombre nous donne une *idée* de la dimension de E qui peut être enfermé dans des petites boîtes de rayon ϵ, en nombre $N_E(\epsilon)$, à l'échelle ϵ. On montre [27] qu'en général, $N_E(\epsilon)$ est une puissance de $(1/\epsilon)$.

Si on a pour $\epsilon < 1$: $C_1\epsilon^{-\alpha} \leq \mathrm{N}_E(\epsilon) \leq \mathrm{C}_2\epsilon^{-\alpha}$ *où* C_1 et C_2 sont des constantes supérieures à zéro, alors α s'obtient à partir de $N_E(\epsilon)$ par la relation :
α= lim (log $\mathrm{N}_E(\epsilon)$)/log$(1/\epsilon)$ quand ϵ tend vers zéro avec $\epsilon > 0$.

Définition 11 : On appelle dimension de boîte de E et on note $dim_B(E)$ la valeur de α, définie précédemment, qui peut être éventuellement infinie.

Définition 12 : On appelle ϵ nombre de packing de E et on note $P_E(\epsilon)$, le plus grand nombre n de points x_1, ... , x_n de E, deux à deux distants de plus ϵ tel que d(x_i,x_j) $> \epsilon$ si i $\neq$ j, alors $P_E(\epsilon)$ vérifie la relation : $N_E(\epsilon) \leq P_E(\epsilon) \leq N_E(\epsilon/2)$.

Bibliographie

[1] Akaike, H. A new look at the statistical. *IEEE Transaction on automatic control AC-19* (1974), 716–723.

[2] bala, J., Huang, J., vafaie, H., Dejong, K., and Wechsler, H. Hybrid learning using genetic algorithms and decision tress for pattern classification. *IN Proccedings of the 14 th International Joint conference on artificial Intelligenecy* (1995), 719–724. Montreal, Canada.

[3] ben Ishak, A. *Sélection de variables par les machines à vecteurs supports pour la descrimination binaire et multiclasse en grande dimension.* PhD thesis, Université de Tunis. Institut supérieur de gestion de Tunis, 2007.

[4] Bouveyon, C., Girard, S., and Schmid, C. Analyse discriminante de haute dimension. *INRIA* (2005), 43.

[5] Bozdogan, H. Model selection and akaike's information criterion (aic) : The general theory and its analytical extensions, psychometrica. . *52* (1987), 345–370.

[6] Breiman, L., Freidman, J.-H., and Stone, R.-A. Classification and regression trees. *New York : Chapman Hall* (1984).

[7] Chen, S., guerra Salcedo, C., and Smith, S. Non-standard crossover for a standard representation commonality-based feature subset selection. *In W. banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, Proceedings of the Genetic and Evolutionary Computation Conference 1* (1999), 129–134. Orlando, Florida, USA, 13-17.

[8] Cherkauer, K., and Shavlik, J. Growing simpler decision trees to facilitate knowledge discovry. *In AAAI Press, editor, Proccedings of the second International*

Conference on Knowledge Discovry and Data Mining (KDD'96) (1996), 315–318. Portland, OR, USA.

[9] Cornuéjols, A. Une nouvelle méthode d'apprentissage : Les svm. séparateurs à vaste marge. Tech. Rep. 51, L'AFIA France, Juin 2002.

[10] Corouge, I. *Modélisation statistique de formes en imagerie cérébrale*. PhD thesis, Univérsité de Renne 1, 2003.

[11] Costa, J., and Girotra. Estimating local intrinsic dimension with k-nearest neighbor graphs. *Statistical signal processing IEEE/SP 13 th workshop July 17-20* (2005), 417–422.

[12] Cristianini, N., and Shave-Taylor, J. An introduction to support vector machines. *Cambridge University Press. cambridge, UK.* (2000).

[13] Dreyfus, Martinez, and Samuelides. *Réseaux de neurones : méthodologie et applications*, 2 ième ed. Eyrolles, 2004.

[14] E.Edwin, and Ghiselli. *Theory of Psychological Measurement*. Book Company, 1964.

[15] George, H., Kohavi, R., and Karl, P. Irrelevant features and the subset selection problem. *International conference on machine learning* (1994), 121–129.

[16] Goldberg, D. Algorithmes génétiques. *Adisson Wesley France* (1994).

[17] Guerra-Salcedo, C., Chen, S., Whitley, D., and Smith, S. Fast and accurate feature selection using hybrid genetic strategies. *In P.J. Angeline, Z. Michalewicz, M. Choenauer, X. Yao, and A. Zalzala, editors, Proccedings of the Congress on Evolutionnary Computation 1* (1999), 177–184. Mayflower Hotel, Washington D.C. USA, 6-9. IEEE Press.

[18] Guerra-Salcedo, C., and Whitley, D. Genetic search for feature subset selection : A comparison between chcand genesis. *In J.R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M.H. garzon, D. E. Goldberg, H. Iba and R. Riolo, edi tors, Genetic Programming : Proccedings of the third annual conference* (1998), 504–509. University of Wisconsin, Madison Wisconsin, USA.

[19] Guon, I., and Elisseff, A. An introduction to variable and feature selection. *Journal of machine learning research 3* (2003), 1157–1182.

[20] Holland, J. Adaptation in natural and artificial systems. *University of Michigan Press.* (1975).

[21] Hyvarinen, A., and Oja, E. Independent component analysis : algorithms and applications. *Neuronal networks 13(4-5)* (2000), 411–430.

[22] IEEE Computer Society Press, e., Ed. *Robust feature selection algorithms* (1993), In Proceedings of the Fifth Conference on Tools for Artificial Intelligence. 356-363 Boston.

[23] Lebart, L., Piron, M., and Morineau, A. *Statistique exploratoire multidimensionnelle.*, 4 ième ed. Dunod, 2006.

[24] Levina, L. Maximum likelihood estimation of intrinsic dimension. *Department of statistics University of Michigan* (2003).

[25] libre Wikipédia, E. http ://www.geog.umontreal.ca/geotrans/fr/ch4fr/meth4fr/ch4m1fr.html. *Intelligence artificielle* (2003).

[26] Mallows., C. Some comments on cp. technometrics. *15 :661-675* (1973).

[27] Mandelbrot, and Benoit, B. *Les objets fractales : Forme,Hasard et Dimension*, paris ed. Flammarion, 1989. 268p. ill.

[28] Maria, S. *Modélisation parcimonieuse : application à la sélection de variables et aux données STAP.* PhD thesis, Université de Renne 1, 2006.

[29] Mohamadally, H., and Fomani, B. Machines à vecteurs de support ou séparateurs à vastes marges. Versailles St Quentin, France, 2003.

[30] Muller, A. Reconnaissance dynamique d'environnement par un robot mobile de type khepera. application à un problème de localisation non métrique. Master's thesis, Laboratoire CEMIF, 2002.

[31] Opitz, D., and Shavlik, J. Using genetic search to refine knowledge-based neural networks. *In Morgan Kaufman, editor, Machine learning : Proccedings of the Eleenth International Conference* (1994), 208–216.

[32] Organization. Machine learning. *http ://www.kernel-machines.org* (2007).

[33] Pawlak, Z. Rough sets. *International journal of computer and information sciences 11* (2000), 341–356.

[34] Pei, M., Goodman, E., Punch, W., and Ding, Y. Genetic algorithms for classification and feature extraction. *In Annual Meeting : Classification Society of North America* (June 1995).

[35] Quinlan, J.-R. Discovring rules by induction from large collections of example in d. michie (ed). *Expert systems in the micro electronic Age, Edinburgh University Press.* (1979).

[36] Quinlan, J.-R. Simplifying decision trees. *International journal of man-machine studies 27* (1987), 221–234.

[37] rakotomamonjy, A., and frédéric Suard. Sélection de variables par svm : Application à la détection des pietons. *INSA de Rouen* (2005).

[38] repository Of Machine Learning Databases, T. U., and theries., D. . ftp ://ftp.ics.uci.edu/pub/machine-learning-databases, 2005.

[39] Roweis, T., and Saul, L. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (2000), 2323–2326.

[40] Schwarz, G. Estimating the dimension of a model. The Annals of statistics, 6(2), 1978.

[41] Seber, G.-A.-F., and Lee., A.-J. Linear regression analysis, second edition. John Wiley and Sons, 2003.

[42] Skowron, A., and Rauszer., C. The discernability matrices and functions in information systems. *In intelligent decision support : Handbook of applications and advances of Rough Sets theory.* (1992), 331–362. R. Slowinski ´ed. Boston.

[43] Slowinski, R. *Handbook of Applications and Advances of the Rough Sets Theory*, intelligent decision support. ed. Kluwer Academic Publishers, Dordrecht, 1992.

[44] Smith, L. Intrinsic limits on dimension calculations. *Pys lett. A 133* (1988), 283–288.

[45] Tenenbaum, B., Silva, V. D., and Langford, J. A global geometric framework for non linear dimensionality reduction. *Science 290* (2000), 2319–2323.

[46] Thompson, M. Selection of variables in multiple regression, part 1 and 2. International Statistical Review, 46 :1-19, 129-146, 1978.

[47] Traina, C., Traina, A., and Faloustos, C. Fast feature selection using fractal dimension. Department of computer science-Carnegie Mellon university. USA, 2000.

[48] Traina, C., traina, A., Wu, L., and Faloustos, C. Fast feature selection using fractal dimension. *In XV Brazilian symposium on Database (SGBD)* (2000). Department of cumputer science and statistics. University of Sao Paulo at Sao Carlos - Brazil.

[49] Wehenkel., L. Théorie de l'information et du codage. Faculté des sciences appliquées, Université de Liège, 2003.

[50] Zealand, W. N. http ://www.cs.waikato.ac.nz/ml/weka. Machine Learning Project, 2007.