UNIVERSITE LILLE1 DES SCIENCES ET TECHNOLOGIES
UNIVERSITE DE TUNIS EL MANAR
ECOLE NATIONALE D'INGENIEURS DE TUNIS

THESE

Présentée en vue d'obtenir le grade de

DOCTEUR

en

Spécialité : Automatique, Génie Informatique, Traitement de signal et Images

par

Sabra EL FERCHICHI

Ingénieur - ENIT

Doctorat délivré conjointement par l'Ecole Nationale d'Ingénieurs de Tunis ET l'Université des Sciences et Technologies Lille 1

Titre de la thèse :

Sélection et Extraction d'attributs pour les problèmes de classification

Soutenue le 01/07/2013 devant le jury d'examen

Président	: Olivier COLOT	Pr., Université des Sciences et Technologies Lille 1
Rapporteur	: Stéphane CANU	Pr., Institut National des Sciences Appliquées de Rouen
Rapporteur	: Moncef TAGINA	Pr., Ecole Nationale des Sciences de l'Informatique
Examinateur	: Kaouther LAABIDI	MC., Ecole Sup des Sciences et Techniques de Tunis
Examinateur	: Aziz MOUKRIM	Pr., Université de Technologie de Compiègne
Examinateur	: Kamel ABDERRAHIM	MC., Ecole Nationale d'Ingénieurs de Gabés
Invité	: Salah ZIDI	Consultant de recherche à l'entreprise ARCHIMED
Directeur	: Salah MAOUCHE	Pr. Université des Sciences et Technologies Lille 1
Co-Directeur	: Moufida KSOURI	Pr. Ecole Nationale d'Ingénieurs de Tunis

Thèse préparée dans le laboratoire LAGIS de Université des Sciences et Technologies Lille 1 et au LACS de l'Ecole Nationale d'Ingénieurs de Tunis, sous la direction de Moufida Ksouri et Salah Maouche

A mon petit bout de chou, A mon bébé.

Remerciements

J'exprime mes profonds remerciements à Monsieur Salah MAOUCHE, Professeur à l'Université des Sciences et Technologies Lille 1, de m'avoir accepté dans son équipe, pour sa confiance et ses précieux conseils.

J'adresse également mes vifs remerciements à Madame Moufida KSOURI, Professeur à l'Ecole Nationale d'Ingénieurs de Tunis, pour ses conseils constructifs et l'encouragement continuel durant toute la période d'encadrement.

J'aimerais transmettre ma gratitude à Monsieur Salah ZIDI, Consultant de recherche à l'entreprise ARCHIMED, pour sa disponibilité, sa patience et ses précieuses directives qui ont permis de structurer et améliorer ce travail.

Ma gratitude s'adresse également à Madame Kaouther LAABIDI, Maître de Conférence à l'Ecole Supérieure des Sciences et Techniques de Tunis, pour sa confiance, son engagement et son encouragement.

Je tiens à exprimer mes sincères remerciements à Monsieur Olivier COLOT, Professeur à l'Université des Sciences et Technologies Lille 1, pour l'honneur qu'il m'a fait en présidant ce jury.

J'adresse ma vive reconnaissance à Monsieur Moncef TAGINA, Professeur à l'Ecole Nationale des Sciences de l'Informatique et Monsieur Stéphane CANU, Professeur à l'Institut National des Sciences Appliquées de Rouen pour l'intérêt qu'ils ont porté à mon travail en acceptant de rapporter ce manuscrit et d'avoir examiné minutieusement ce travail.

Je remercie Monsieur Kamel ABDERRAHIM, Maître de conférences à l'Ecole Nationale d'Ingénieurs de Gabès et Monsieur Aziz MOUKRIM, Professeur à l'Université de Technologie de Compiègne pour m'avoir fait l'honneur d'accepter de participer au jury en tant que examinateurs.

J'aimerais remercier touts mes amis et mes collègues à l'ACS pour leur soutiens et encouragements.
Mes remerciements s'adressent finalement à toute ma famille pour sa patience et son encouragement infaillible durant toutes les années de mes études.

Table des matières

Table des figures

Liste des tableaux

LISTE DES TABLEAUX

Introduction générale

générale

Chapitre 1

Réduction de dimension pour les problèmes de classification

Contents

1.1 Introduction

L'apprentissage et la classification automatique (Machine Learning) ont été conçu pour la résolution automatique des problèmes complexes par la prise de décisions sur la base d'observations de ces problèmes.
En effet, le progrès scientifique a permit d'augmenter sensiblement les capacités de

mesures et a ainsi augmenté la taille des bases de données. Il est à présent difficile pour un opérateur humain de traiter ces données dans un temps raisonnable. Ce qui a rendu la tâche de prise de décision plus difficile et complexe pour un opérateur humain.

Dans ce chapitre, nous présenterons tout d'abord au paragraphe (1.2) le problème de la classification. Nous verrons notamment que la classification se divise généralement en deux sous-problèmes distincts : la classification supervisée, appelée également analyse discriminante, et la classification non supervisée, dénommée aussi clustering ou classification automatique. Les méthodes classiques de l'analyse discriminante seront présentées au paragraphe (1.2.2) et les méthodes de classification automatique le seront au paragraphe (1.2.3). La classification semi supervisée sera introduite au paragraphe(1.2.4).

D'autre part, les processus d'acquisition des données ayant aussi progressé rapidement, la dimension des données à étudier est devenue très grande. Nous verrons au paragraphe (1.3) que la grande dimension des données pose des problèmes particulièrement en classification. Ce problème est généralement appelé *fléau de la dimension* (1.3.2), nous verrons dans la suite les solutions qui existent pour cette limitation des méthodes de classification dans la grande dimension (1.4).

1.2 Problème de classification

1.2.1 Formulation

La classification consiste en l'organisation d'un ensemble d'objets en k classes homogènes selon des critères prédéfinis. Ainsi, on définit une classe comme étant un sous-ensemble d'objets à classer et qui sont jugés similaires selon des critères prédéfinis.

Chaque élément X de l'ensemble des objets à classer est décrit par p variables descriptives. Le but général de la classification étant d'assigner chaque vecteur X à une des k classes déjà définies en se basant sur les attributs qui les décrits. On définit alors le label $Y \in \{1, ..., k\}$ pour chaque élément X tel que :

$$Y = i \;\; si \;\; X \;\; appartient \;\; \text{à} \;\; la \;\; i^{éme} \;\; classe.$$

Le classificateur sera amené ainsi à générer des règles de décision δ qui va associer à chaque vecteur $X \in \mathbb{R}^p$ de l'ensemble des données, un label $y \in \{1, ..., k\}$:

$$\delta : \mathbb{R}^p \to \{1, ..., k\} \qquad\qquad X \to y$$

Ces règles de décision seront construites à partir d'un jeu de données appelé données d'apprentissage. Selon la nature de ce jeu de données, on peut différencier trois principaux types d'apprentissage : apprentissage supervisé, apprentissage non supervisé et apprentissage semi-supervisé.

1.2.2 Classification supervisée

Dans le cadre de la classification supervisée, des données d'apprentissage labellisées sont disponibles afin d'apprendre les règles de décisions qui permettront de distinguer les k classes de données. Ces observations, dites d'apprentissage, supervisent la construction du classifieur.
La classification supervisée se base sur deux aspects principaux : descriptif et décisionnel. L'aspect descriptif essai de trouver une représentation qui permet l'interprétation des groupes grâce aux variables explicatives. La complexité de l'espace de représentation peut rendre cette tâche difficile, surtout quand la dimension des données dépasse la dimension 3.
Quant à l'aspect décisionnel, on cherche à définir la meilleure affectation d'un nouvel individu dont on ne connaît que les valeurs des variables explicatives qui le décrivent. Cet aspect est particulièrement intéressant dans des domaines où la notion de diagnostic est essentielle.
Apprentissage supervisé Dans ce cas, chacune des données utilisées dans l'apprentissage appartient à une des k classes. Les données $x_1, ..., x_n$, décrites par p variables explicatives et appartenants aux différentes k classes $y_1, ..., y_n$ seront ainsi représentées par un ensemble de couples (observations, labels) sous la forme suivante [Bouveyron, 2006] :

$$(x_1, y_1), ..., (x_n, y_n),$$

Les règles de décision seront alors générés à partir des informations contenues dans les labels des classes et les attributs.

1.2.2.1 Problème de classification supervisée

La mission principale de la classification supervisée est de prédire l'appartenance de chaque observation x, décrite par p variables explicatives, à une classe parmi les k

classes $C_1, ..., C_k$ définies à priori. Afin de déterminer l'appartenance de l'observation x, on dispose d'un ensemble d'apprentissage A définie par :

$$A = (x_1, y_1), ..., (x_n, y_n), tel \ \ que \ \ x_j \in \mathbb{R}^p, y_j \in 1...k \tag{1.1}$$

Où le vecteur x_j est la j^{eme} observation et y_j est son label qui indique la classe à laquelle x_j appartient. Le jeu de données A sera utilisé pour construire une règle de décision Č qui affectera chaque vecteur x de $\mathbb{R}^p$ à une des k classes $C_1, ..., C_k$. Cette règle de décision sera validée par la suite sur un ensemble d'observations, dit de test ou de validation.
On peut distinguer deux approches principales pour aborder la classification supervisée : les approches génératives et les approches discriminantes.

1.2.2.2 L'approche générative

Cette approche modélise la densité de chacune des k classes. L'estimation de la densité peut être non paramétrique et basée par exemple sur la méthode du noyau ou bien paramétrique et basée sur un modèle de mélange gaussien. Les deux méthodes les plus populaires de cette approche sont l'analyse discriminante quadratique (QDA) [Srivastava et al., 2007] et l'analyse discriminante linéaire (LDA) [Xiang and Huang, 2006].

1.2.2.3 L'approche discriminante

Au contraire de l'approche précédente, cette approche vise à estimer directement la probabilité à posteriori $P(Y|X)$ par la minimisation d'un coût de classification.Les méthodes discriminantes ont montré leur efficacité sur le plan pratique. Parmi les méthodes les plus réussies, les séparateurs à vaste marge (SVM) [Vapnick, 1996]. Il est à noter en revanche que cette méthode n'est pas nativement multi-classes. D'autres méthodes discriminantes existent telle que les k plus proches voisins, et les arbres de décisions.

1.2.3 Classification non supervisée

La principale différence entre la classification non supervisée ou le clustering, et la classification supervisée, est que le jeu de données ne comprend aucune information sur la distribution des données d'apprentissage. Autrement dit, le jeu d'apprentissage ne contient que les observations $x_1, ..., x_n$ et les labels $y_1, ..., y_n$, alors que le nombre des classe est manquant.

Apprentissage non supervisé Dans le cadre de l'apprentissage non supervisé, les classes des données ne sont pas connues à priori : les labels $y_1, ..., y_n$ ne sont pas donnés. Donc le jeu de données ne peut être représenté qu'avec les valeurs $x_1, ..., x_n$ décrites eux mêmes par les p variables explicatives [Bouveyron, 2006]. L'apprenant ou bien le classificateur ne peut se baser que sur la notion de similarité entre attributs pour pouvoir retrouver les règles de décision.

1.2.3.1 Problème de classification non supervisée

Contrairement à la classification supervisée, la classification non supervisée ne dispose pas d'un jeu d'apprentissage pour apprendre les caractéristiques discriminantes des k classes. Ainsi, sa mission consiste à identifier des groupes d'objets similaires appelés clusters à partir d'un ensemble de données sans en connaître au préalable la structure.
Autrement dit, elle essaye de prédire les labels $y_1, ..., y_n$ des observations $x_1, ..., x_n \in \mathbb{R}^p$ et ce en se basant sur la seule connaissance des valeurs prises par les p variables explicatives. Une difficulté supplémentaire vient s'ajouter en classification non supervisée est que l'on ne connaît pas nécessairement le nombre k de groupes [Jain et al., 1999].

1.2.3.2 Notion de similarité

L'analyse automatique d'un ensemble de données ou de connaissances nécessite un outil capable d'évaluer les ressemblances ou les relations qui existent au sein de ces données manipulées. Sur cette base, il devient alors possible d'ordonner les éléments de l'ensemble, de les hiérarchiser ou d'en extraire des invariants.
Cette notion de similarité a fait l'objet de plusieurs recherches dans divers domaines comme l'analyse des données, l'apprentissage symbolique ou encore la reconnaissance des formes. Suite à cette diversité, les fonctions de similarités sont exprimées sous des formes multiples, conçues autour des différents buts poursuivis [Bisson, 2000].
Comme on l'a déjà mentionné dans la section précédente, la classification automatique, vise à structurer un ensemble de données, en fonction de leurs ressemblances, sous la forme d'un ensemble de classes à la fois homogènes et contrastées comme le montre la figure 1.1. Ainsi, le critère de formation des classes consiste à maximiser la mesure de similarité intra-classes ou/et à minimiser la mesure de similarité inter-classes [Bisson, 2000].

Notion mathématique de distance : Le concept de similarité peut être dé-

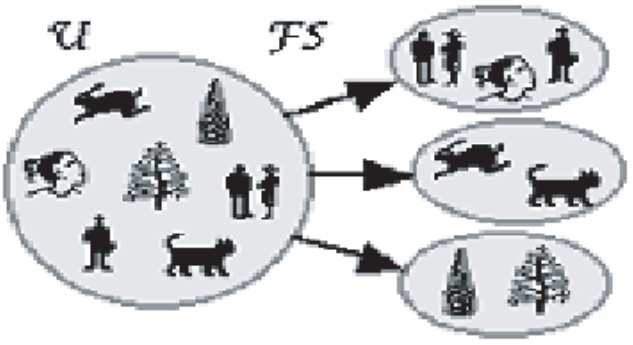

FIGURE 1.1 – Principe de Classification extrait de [Bisson, 2000]

fini à l'aide de la notion duale de dis-similarité ; deux individus sont d'autant plus similaires qu'ils sont proches au sens d'une mesure de dis-similarité. La définition générale d'une mesure de dis-similarité est :
Définition de mesure de dis-similarité [Guérif, 2006] : On appelle indice ou mesure de dis-similarité sur un ensemble Ω, une application $\Omega \times \Omega \rightarrow \mathbb{R}^+$ qui vérifie pour tout couple $(x, y) \in \Omega \times \Omega$ les propriétés suivantes : :

$$d(x, y) = d(y, x) \quad (symetrie) \tag{1.2}$$

$$d(x, y) = 0 \Leftrightarrow x = y \quad (separabilite) \tag{1.3}$$

L'homogénéité des individus regroupés au sein d'un groupe est souvent évaluée à l'aide d'un critère statistique appelé variance dont la définition est :
Définition de variance [Guérif, 2006] : On définit la variance $V(C_i)$ d'un groupe d'objets C_i comme suit :

$$V(C_i) = \frac{1}{N_i} \sum_{x_j \in C_i} d^2(x_j - \mu_i) \tag{1.4}$$

Où N_i et μ_i sont respectivement le nombre d'objets et le centroïde du groupe C_i [Guérif, 2006].
Dans le contexte de la classification non supervisée, on distingue généralement la variance intra-classe V_{intra}, que l'on souhaite minimiser, et la variance inter-classe

V_{inter}, que l'on cherche à maximiser :

$$V_{intra} = \frac{1}{N} \sum_{C_i \in C} N_i \times V(C_i), \tag{1.5}$$

$$V_{inter} = \frac{1}{N} \sum_{C_i \in C} N_i \times (\mu_i - \mu)^2 \tag{1.6}$$

avec N_i et μ_i représentent respectivement le nombre d'objets et le centroïde du groupe C_i. N et μ désignent respectivement le nombre d'objets et le centroïde de Ω. La première variance évalue l'homogénéité moyenne des groupes d'une partition et la seconde variance permet de quantifier la différence entre les groupes. La variance totale est définie par la relation suivante : $V_{total} = V_{intra} + V_{inter}$.

Classification des similarités

Une classification possible des fonctions de similarité peut se faire en se basant sur le langage utilisé pour la représentation des similarités [Bisson, 2000]. D'une part, on trouve les similarités numériques qui quantifient les ressemblances sous la forme d'une valeur. D'autre part, les similarités symboliques qui permettent de caractériser les ressemblances.

Une deuxième classification des similarités repose sur le critère du langage de description des données utilisé : les représentations de type proportionnel (vecteurs et attributs) et les représentations de type relationnel (graphes, objets et logique des prédicats).

Une troisième classification distingue les similarités non-informées qui ne portent que sur les données, des similarités informées qui utilisent explicitement des connaissances à priori.

En fait, des informations supplémentaires sur le contexte du travail sont généralement disponibles. Ces informations permettent, en conjonction avec les connaissances des buts poursuivis, de sélectionner la fonction de similarité la mieux adaptée ou bien d'en concevoir une spécifique. Ces informations peuvent être de nature statistique (modèle de répartition des valeurs) ou plutôt des modèles exprimés sous forme symbolique (règles).

Les distances les plus connues et les plus utilisées, pour la classification, sont la distance euclidienne ou de Manhattan, qui ne sont en réalité que des cas particuliers

de la mesure de Minkowski :

$$d_p(x,y) = (\sum_{i=1}^{K} W_i \times |x_i - y_i|^p)^{\frac{1}{p}} \tag{1.7}$$

En fonction du paramètre p :
$p = 1$ *distance de Manhattan.*
$p = 2$ *distance Euclidienne.*
$p = \infty$ *distance de Chebychev.*
Dans cette formule, les variables x_i et y_i représentent respectivement les valeurs du i^{eme} attribut décrivant les exemples x et y et le terme W_i représente le poids associé à cet attribut. Le terme de distance Euclidienne est aussi utilisé pour caractériser une autre famille de distances dite distance Quadratique exprimée par :

$$d(x,y) = ((x_i - y_i)^t M (x_i - y_i))^{\frac{1}{2}} \tag{1.8}$$

En fonction de la matrice M :
$M = 1$ *distance Euclidienne classique.*
$M = V^{-1}$ *distance de Mahanalobis.*
$M = D^{-1}$ *distance du* χ^2.
Dans cette formule, le terme $(x_i - y_i)$ représente le vecteur colonne des différences entre les attributs des deux exemples x et y. Pour la distance Mahanalobis, la matrice V correspond à la matrice de variance-covariance entre l'ensemble des exemples. Pour le χ^2 qui est une distance plutôt adaptée à la comparaison de profils de distribution de modalités, la matrice D est une matrice diagonale exprimant la fréquence relative d'apparition de chacune des modalités dans les exemples [Bisson, 2000].
Les fonctions de similarité peuvent être exprimées sous des formes multiples (cosinus, coefficient de corrélation de Pearson, Gaussienne, voire floue....) [Bouchon-Meunier et al., 1996].

1.2.3.3 Approches classiques pour la classification non supervisée

Méthodes hiérarchiques : La classification hiérarchique est une suite ordonnée de partitions emboîtées dont le premier terme est la partition la plus fine qui ne contient que des singletons, et le dernier terme est la partition la plus grossière qui ne comporte qu'une seule partie [Jain et al., 1999].
On distingue deux types d'approches de classification hiérarchique : les méthodes descendantes (divisives) et les méthodes ascendantes (agglomeratives).

1. Méthodes descendantes : On commence en considérant chaque point comme

une classe et on essaye de fusionner deux ou plusieurs classes pour former une nouvelle classe.

2. Méthodes ascendantes : En considérant tous les points comme une seule classe au début, on divise successivement les classes en classes plus raffinées.

Méthodes de partitionnement : Ces méthodes produisent une seule partition des données au lieu d'une structure de classes. Elles créent les classes en optimisant une fonction objective qui est définie d'une façon locale (sur un sous-ensemble des données) ou globales (sur tous les données). Chaque partition est représentée par un prototype, appelée aussi centre ou centroïde, et chaque objet est affecté au groupe dont il est le plus proche [Jain et al., 1999].

1. K-medoids : Une classe est représentée par un de ses points, qui s'appelle medoid. Une telle représentation peut s'adapter à n'importe quel type d'attributs, et le medoid est choisi comme une fraction des points prédominants dans une classe, donc il n'est pas sensible aux aberrants.
2. K-means : Chaque classe est représentée par sa moyenne ou la moyenne pondérée qui est nommée centroid. L'algorithme commence avec un ensemble de k centroids choisis par l'utilisateur. Les données seront partitionnées en k classes : un point appartient à une classe si le centroid de cette classe est le plus proche. La mise à jour des centroids et l'affectation des points des données aux classes sont réalisées au cours des itérations successives.

Modèle de mélange : On suppose que l'ensemble d'objets dont on dispose a été obtenu en fusionnant plusieurs sous populations qui suivent chacune une loi de probabilité propre [Jain et al., 1999]. La probabilité qu'un objet x soit issu de ce mélange de paramètres $\theta = (\alpha_1, \theta_1, ..., \alpha_i, \theta_i, ...)$ est alors donnée par :

$$p(x/\theta) = \sum_i \alpha_i \times p(x/\theta_i) \tag{1.9}$$

où les coefficients de mélange α_i satisfont $\sum_i \alpha_i = 1$, et les densités de chaque sous population C_i sont données par les lois $p_i(x/\theta_i)$ de paramètres θ_i.

Approches neuromimétiques : les cartes auto-organisées de Kohenen : Le procédé d'auto-organisation proposé par Kohenen cherche à transformer des signaux de départ de dimension quelconque en signaux à une ou deux dimensions. Le but principal étant de reproduire en sortie les corrélations qui sont présentes dans les données à l'entrée [Guérif, 2006].

D'une manière générale, les cartes auto-organisatrices vont projeter les données initiales sur un espace discret et régulier de faible dimension.
Grâce au procédé d'auto-organisation, la topologie qui lie les données initiales est conservée au niveau des réponses proposées par le réseau. La localisation des neurones actifs reproduit des liens existants au niveaux des données initiales.
Généralement, dans le cadre d'apprentissage non supervisé les relations de voisinage entre formes d'entrée sont inconnues. C'est l'observation des voisinages produits par la carte qui vont permettre l'interprétation des données initiales. En particulier, ils vont définir la notion de formes proches dans l'espace initial [Guérif, 2006].

1.2.4 Classification semi-supervisée

Ces dernières année, la classification semi-supervisée reçoit de plus en plus d'attention [Zhu, 2006], [Chapelle et al., 2006], [Zhao et al., 2008], vu que la procuration des données labellisées demande un effort humain couteux à l'encontre des données non labellisées, qui sont faciles à obtenir [Liu et al., 2008].
La classification semi-supervisée est à mi-chemin entre l'apprentissage supervisé et l'apprentissage non supervisé. En effet, en plus des données non labellisées, l'algorithme procure quelques informations supervisées pour quelques uns parmi tout l'ensemble des données. En général, cette information provient des labels de quelques exemples d'apprentissage.
Dans ce cas, l'ensemble des données $X_i, i = 1..n$ peut être divisé en deux parties : d'une part, on trouve les points $X_l = x_i..x_l$ pour lesquels les attributs $Y_i = y_1...y_l$ sont fournies et d'autre part, on trouve les points $X_u = x_{l+1}..x_{l+u}$ pour lesquels les labels ne sont pas connus.
D'autres formes de supervision partielle peuvent être fournies, par exemple sous formes de contraintes tel que les points ont (ou n'ont pas) le même label.
Ainsi, la classification semi-supervisée peut être considérée comme un problème non supervisé guidé sous contraintes.
La plupart des approches considère l'apprentissage supervisé comme étant un problème d'apprentissage non supervisé avec information additionnelle sur la distribution des données. Néanmoins, cette approche n'est facilement applicable dans le cas où le nombre ou la nature de classe ne sont pas connus à priori et doivent être estimés à partir des données [Liu et al., 2008].
Apprentissage semi supervisé la détermination des labels de chaque vecteur X s'avère parfois couteuse et difficile à détenir. Ainsi, dans le contexte de l'appren-

tissage semi-supervisé, quelques informations sont disponibles sur l'ensemble des données. Ces informations sont représentées sous la forme de données labellisées, soit sous la forme de ressemblance ou dissemblance au sein de l'ensemble de données [Zhao et al., 2008], [Liu et al., 2008].

1.2.4.1 Problème et suppositions

En comparaison avec l'apprentissage supervisé qui utilise seulement des données labellisées, l'apprentissage semi-supervisé espère obtenir plus de précision à partir des données non labellisées dans le cadre de l'apprentissage semi supervisé.
La connaissance de $p(x)$ qu'on peut déduire à partir des données labellisées, doit contenir des informations utiles pour l'inférence de $p(y/x)$ pour que l'apprentissage semi-supervisé apporte une amélioration à l'apprentissage non supervisé. Il peut arriver même que l'utilisation des données non labellisées dégrade la précision de la classification en trompant l'inférence [Chapelle et al., 2006].
Ainsi, certaines assomptions doivent être établies pour pouvoir utiliser l'apprentissage semi-supervisée (SSP) :

- Supposition de dérivabilité infinie de l'apprentissage semi-supervisé : si deux points x_1, x_2 sont proches alors les labels y_1, y_2 correspondants doivent être de même. Cette proposition implique que si deux points sont liés par un chemin de grande densité, alors leurs labels sont proches. D'autre part, si deux points sont séparés par une région de faible dimension, alors leurs labels n'ont pas à être proches.
- Supposition du clustering : si deux points sont dans un même cluster alors ils sont probablement de même classe. Supposons que les données de chaque classe tendent à former des clusters, alors les données non labellisées peuvent aider à former les frontières des clusters plus précisément. On peut appliquer un algorithme de clustering et utiliser les points labellisés pour assigner une classe à chaque cluster. Il est à noter que cette assomption n'implique pas que chaque classe doit former un seul compact cluster. Elle signifie généralement qu'on ne peut pas retrouver deux objets de classes différentes dans un même cluster.
- Supposition de la variété : les données de grande dimension sont liées à une variété de faible dimension. La malédiction de grande dimension affecte directement les approches génératives basées sur l'estimation de la densité dans l'espace d'entrée, et plus sévèrement les approches discriminantes vu que la distance entre deux points devient similaire dans la grande dimension et non significative.

Plusieurs approches existent pour entamer la classification semi-supervisée y compris les approches génératives, transductives, self-training ou bien co-training. En effet, les données labellisées sont rares, ce qui fait que les différentes méthodes se basent sur des modèles de suppositions rigides. Idéalement, le choix entre ces méthodes se fait selon la structure du problème.

1.3 Classification des données de grande dimension

Le développement de la technologie de mesure implique un nombre de plus en plus grand des données ainsi que leurs dimensions.
Ainsi, la taille des bases de données ne cesse d'augmenter comme dans le cas des données clients stockées par les entreprises [Feraud et al., 1998], des données spatiales, des données satellites, données textuelles [Bekkerman et al., 2003], données génomiques ainsi que l'analyse d'image (les résolutions actuelles des appareils photos numériques (12 mégapixels)) [Bouveyron, 2006].
Les difficultés qui apparaissent lors de la manipulation des données de grande dimension, sont attribuées au phénomène appelé "fléau de dimensionnalité". Quand la dimension des données augmente, les propriétés bien connues de l'espace euclidien usuel 2D et 3D ne sont plus valables. Ces données de grande dimension posent différents problèmes [Donoho, 2000], dans le contexte de la classification supervisée ainsi que la classification non supervisée ou clustering.

1.3.1 Fléau de dimensionnalité

Le terme fléau de la dimensionnalité a été évoqué la première fois dans le cadre d'optimisation d'une fonction par une recherche exhaustive dans un espace discrétisé par Bellman [Verleysen, 2003].
En effet, si on considère une fonction définie sur le cube unité dans un espace de dimension d, et 10 points sont considérés pour chaque dimension. On a besoin alors de 10^d évaluations de la fonction, ce qui est de point de vue computationel, compliqué même pour une dimension d modérée [Donoho, 2000]. Par exemple, si on considère le cube unité en dimension 20, alors il faudra effectuer 10^{20} évaluations de la fonction.
Le fléau de dimensionnalité réfère ainsi au fait qu'en absence d'hypothèses simplificatrice, le nombre des données nécessaires pour estimer une fonction multi-variables

dans un espace donné, avec un taux d'erreur donné, est d'autant plus grand que la dimension.
Ainsi, vu que dans un espace de grande dimension, plusieurs données d'apprentissage sont encore nécessaires pour optimiser ou approximer des fonctions, deux questions se posent. Une sur la limite entre une grande et une petite dimension de l'espace et une sur le nombre des points nécessaire pour effectuer un apprentissage dans un espace de grande dimension.

1.3.1.1 Limite entre une grande et une petite dimension de l'espace

Les méthodes d'apprentissage sont en général toujours à travers des exemples qui ne dépassent pas la dimension 3. Leurs performances par contre peuvent être mesurées dans un espace de grande dimension sans avoir pourtant une représentation possible. De plus, les problèmes réels sont en général de grande dimension. Néanmoins, les variables ne sont pas indépendantes l'une des autres.
Malgré que la redondance contenue dans l'ensemble des variables n'est pas caractérisée, quelques variables de l'espace d'entrée sont liées entre elles. Dés lors, un problème dans un espace de d dimension peut être ramené à une plus petite dimension m : dimension intrinsèque des données. Le phénomène de l'espace vide est ainsi considéré comme un des effets de la grande dimension qui va à l'encontre de la représentation habituelle. Il est alors considéré comme une caractéristique du fléau de dimensionnalité [Verleysen and François, 2005].
On considère le volume d'une boule en dimension d, son volume est déterminé par l'équation suivante :

$$V(d) = \frac{\pi^d}{\Gamma(\frac{d}{2}+1)} * r^d \tag{1.10}$$

avec r est le rayon et Γ est la fonction Gamma usuelle.
Le graphique de la figure 1.2 montre l'évolution du volume de la boule unité en fonction de la dimension de l'espace dans lequel elle se trouve. Il apparaît que le volume de la boule unité devient quasiment nul quand la dimension devient grande. Néanmoins, l'intention içi est de montrer que notre vision intuitive du volume d'une sphere est erronée dans un espace de grande dimension. Afin de mieux appréhender ce phénomène, il est par exemple possible de comparer le volume de la boule de rayon 0.9 à celui de la boule unité. Le rapport étant égal à 0.9 à la puissance d. Naturellement, ce rapport décroît vers 0 quand la dimension augmente. Les valeurs dans la figure 1.2 montrent que 90% du volume d'une sphère de dimension plus

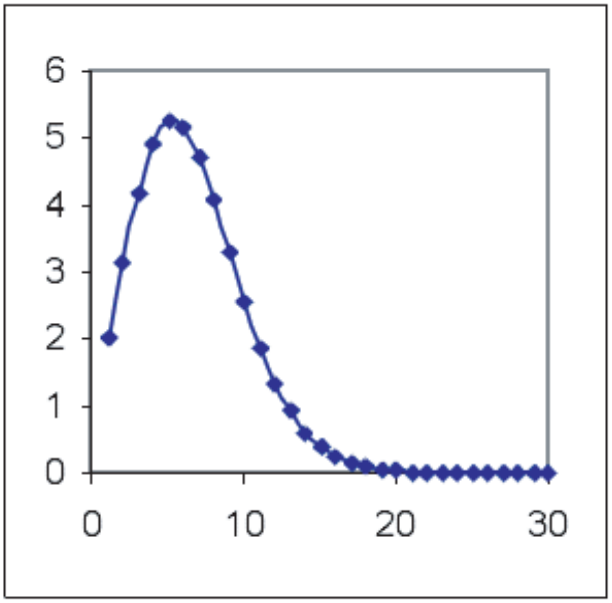

FIGURE 1.2 – Évolution du volume en fonction de la dimension extrait de [Verleysen and François, 2005]

grande que 20 est contenu dans la fenêtre sphérique d'épaisseur 10% du rayon initiale. Cet exemple montre que l'espace de dimension d est presque vide puisque la majorité des points se situe aux alentours d'un espace de dimension $d-1$.
De même, on peut considérer les fonctions gaussiennes dans un espace de grande dimension.
Intuitivement, les fonctions gaussiennes sont utilisées pour leurs propriétés locales : l'intégrité de la fonction est contenue dans un volume limité autour du centre. Quand la dimension augmente, la plupart du volume d'une fonction gaussienne est contenu dans les marges au lieu du centre. Ainsi, une fonction gaussienne peut ne pas être correctement représentée dans un espace de grande dimension vu ses propriétés locales [Verleysen and François, 2005].

1.3.1.2 Nombre de points nécessaires pour l'apprentissage dans un espace de grande dimension

L'apprentissage dans des espaces de grande dimension nécessite entre autre un grand nombre de données d'apprentissage comme on a déjà mentionné au début de la section. En effet, ce nombre croît exponentiellement avec la dimension de l'espace. Silverman [Silverman, 1986] a traité ce problème, en approximant une distribution

gaussienne quelconque avec des noyaux gaussiens fixés. Ses résultats peuvent être approximés par [Comon et al., 1994] :

$$log_{10}N(d) \approxeq 0.6(d - 0.25) \tag{1.11}$$

Le nombre d'observations N nécessaires à cette tâche avec une erreur maximale de 10% croît exponentiellement avec la dimension d.
La figure 1.3 montre l'évolution du nombre d'observations nécessaires pour l'approximation d'une distribution gaussienne quelconque, avec des noyaux gaussiens fixes en fonction de la dimension de l'espace avec une erreur maximale de 10% [Bouveyron and Girard, 2009]. En pratique, un ensemble de données qui ne croit

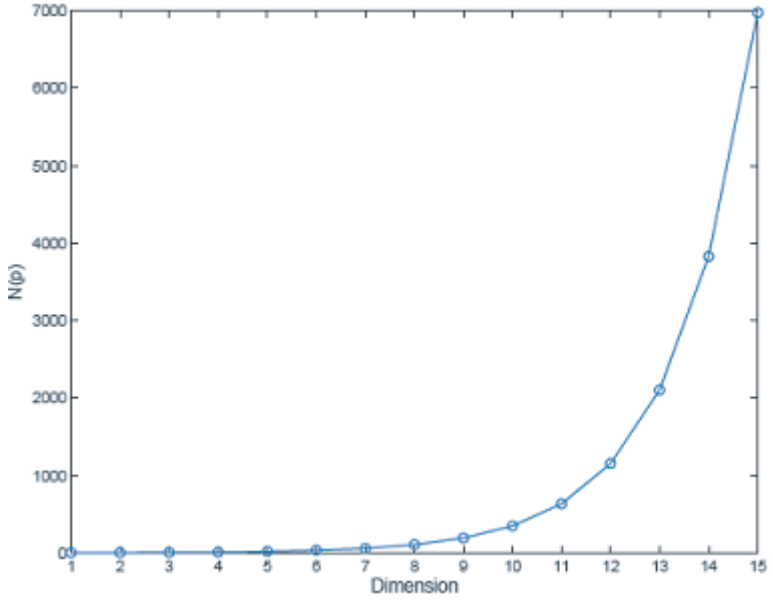

FIGURE 1.3 – Approximation d'une distribution gaussienne extrait de [Bouveyron and Girard, 2009]

pas exponentiellement avec l'espace sera jugé comme étant petit, ou contrairement la dimension sera jugée comme étant grande [Verleysen and François, 2005].
Ainsi, en grande dimension, il n'y aura pas suffisamment de données pour effectuer un apprentissage efficace. En considérant par exemple un problème dans lequel 10 observations sont nécessaires pour la dimension 1, et 100 observations pour la dimension 2, cela veut dire qu'on a besoin de 10^{10} et 10^{20} dans les dimensions 10 et 20 respectivement [Verleysen and François, 2005]. Ce n'est évidemment pas possible pour des problèmes du monde réel où la dimension est beaucoup plus grande que 20
.

Les problèmes du monde réel ne souffre pas aussi sévèrement du fléau du dimensionnalité, étant donné que les données sont localisées à proximité d'un espace de dimension m plus petit que d. La réduction du dimension des données s'avère ainsi nécessaire dans l'apprentissage en grande dimension.

1.3.2 Fléau de dimensionnalité en classification

Le fléau de dimensionnalité nuit énormément à l'efficacité de l'apprentissage des données de grande dimension. En effet, un grand nombre d'observations exponentiellement lié à la dimension, est nécessaire pour mener un apprentissage efficace.
Les approches génératives de classification souffre spécifiquement de ce phénomène, en particulier le LDA. Généralement, ces méthodes requièrent l'estimation d'un nombre de paramètres, pour les matrices de covariance, qui croît avec le carré de la dimension. La qualité de la classification est directement liée à l'estimation de ces matrices. De ce fait, si le nombre d'observations est trop petit devant la dimension de l'espace, alors les estimations de ces matrices seront singulières et leur inversion sera numériquement impossible. L'influence de la taille de l'échantillon d'apprentissage sur la règle de décision de classificateurs génératifs est ainsi évidente.
De plus, dans la classification supervisée, si un classificateur (génératif ou discriminatif) est trop complexe, il peut induire un problème de sur-apprentissage. De ce fait, La capacité de généralisation du classificateur sera dégradée. C-à-d que lorsque le jeu d'apprentissage n'est pas représentatif du processus qui l'a généré, ces règles de décision ne seront pas efficaces pour traiter de nouvelles données.
On déduit depuis la figure 1.4 le comportement typique des erreurs de prédiction des jeux d'apprentissage qui a tendance à décroître quand le degré de complexité du classificateur croît.
Si le modèle tend à être trop complexe, il n'est alors plus assez général et l'erreur de prédiction sur le jeu de validation croît de nouveau.
Dans le cas où le classificateur est trop simple, son efficacité sera réduite sur les données d'apprentissage et sur les données de test : problème de « sous-apprentissage ».
Garantir l'efficacité du classificateur réside dans la détermination du bon degré de complexité du classificateur, qui est lié directement à la dimension des données [Bouveyron and Girard, 2009].
Concernant les méthodes de classification basées sur les fonctions noyaux (kernel machines), elles ne sont pas tout à fait robustes contre le fléau de dimensionalité. En effet, un kernel n'est qu'une mesure de similarité entre deux observations, qui

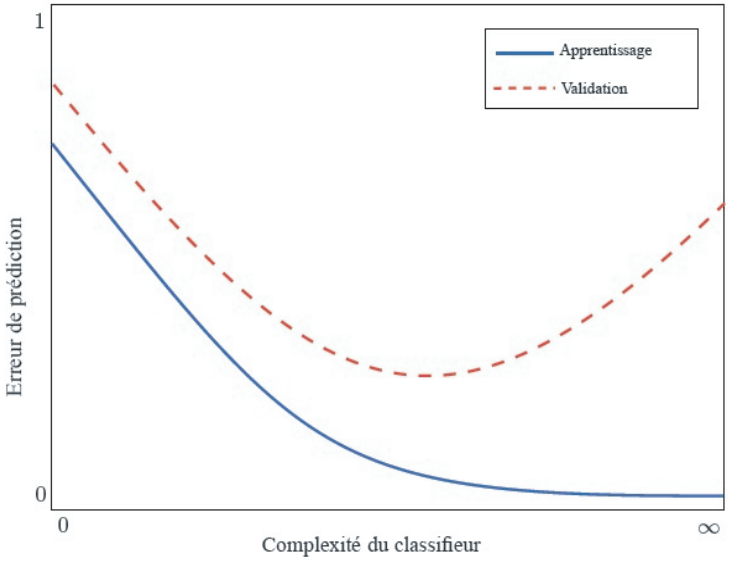

FIGURE 1.4 – Complexité vs. Erreur de prédiction extrait de [Bouveyron and Girard, 2009]

peut être modélisée comme une fonction à variables aléatoires. De ce fait, l'addition des variables non pertinentes et bruitées dégrade les performances du classificateur et la probabilité fondamentale de la reconnaissance de forme écrite sous la forme : [Evangelista et al., 2006].

$$(k(x_1, x_2) \mid y_1 \cong y_2) > (k(x_1, x_2) \mid y_1 \neq y_2) \tag{1.12}$$

Á cela vient s'ajouter le fait que la distance deux à deux de vecteurs aléatoires tendent à se concentrer dans un espace de plus petite dimension que l'espace original d'entrée. Ainsi, quand la dimension augmente, les données deviennent équidistantes, ce qui peut mettre en question l'approche de k-plus proche voisin. De même, les kernels machines sont affectés par ce phénomène : la sortie de la fonction noyau se concentre également sur une gamme étroite et sera donc incapable de distinguer clairement entre les observations d'entrée [Kärnä, 2007].

Généralement, le processus d'acquisition augmente artificiellement la dimension des données dans plusieurs applications. Alors qu'en réalité les données sont concentrées dans une plus petite dimension appelée dimension intrinsèque.
Dés lors, les données fournies, explicitées en très grande dimension p (un millier de dimensions), sont modélisées par un nombre n d'observations disponibles générale-

ment beaucoup plus faible que la dimension p.
Ceci peut induire que l'information contenue dans les données est redondante et un grand nombre de variables seront corrélées. Par conséquence, ramener l'espace d'entrée à une dimension d intrinsèque (ou proche) plus faible que la dimension initial d peut éventuellement nous éviter un tas de problèmes lié au fléau de dimensionnalité évoquée dans la section 1.3.1.
Néanmoins, la grande dimension de l'espace des attributs peut avoir des bénéfiques pour la classification des données : plus la dimension est grande, plus la classification des données est facile avec un classificateur adapté. En effet, dans [Bouveyron, 2006], il met en évidence ce phénomène en simulant des données issues de trois densités gaussiennes dans $R^p, p = 20, ..., 250$ avec une dimension intrinsèque respective de $2, 5$ et 10 et des moyennes très proches.
Dans la figure 1.5, on peut observer clairement que le taux de classification correct du classificateur de Bayes, croît avec la dimension de l'espace original. Ce phénomène est en particulier exploité par les méthodes de discrimination SVM, qui augmentent artificiellement la dimension des données pour faciliter leur discrimination.

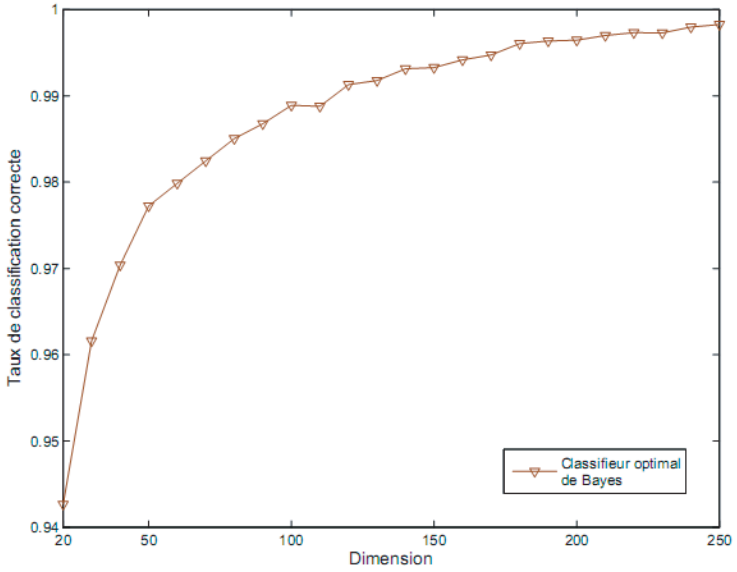

FIGURE 1.5 – Évolution de l'erreur de bayes en fonction de la dimension extrait de [Bouveyron, 2006]

1.4 Réduction de dimension

L'expansion des systèmes d'acquisition et de stockage de données met à notre disposition aujourd'hui des bases de données énormes. Pour comprendre le processus générateur des données et mener des prévisions, les méthodes d'apprentissage permettent d'inférer sur l'information contenue dans ces données et de l'exploiter. En effet, la classification consiste à établir un lien entre les variables explicatives et leur étiquette, à partir d'un échantillon d'apprentissage. Ce qui fait que la qualité des variables qui caractérise ces données influent fortement sur les règles de décision générées par le classificateur.

Dans la plupart des situations, on se retrouve avec un nombre de variables qui a tendance à dépasser largement le nombre d'observations. Par exemple, les données de biopuces disposent d'un grand nombre de variables (les gènes)et d'un nombre réduit d'observations (les facteurs d'expériences) [Ishak, 2007].

Ces données ne s'avèrent pas toute pertinentes pour le processus de classification. Certaines variables correspondent à du bruit, ou elles sont peu informatives, corrélées et redondantes ou même inutiles au problème de classification.

En plus, lors de l'exploration et l'analyse des données, plusieurs problèmes liés à la grande dimension des données peuvent dégrader les performances des classificateurs. En effet, un grand nombre de données non pertinente affecte gravement la qualité de la classification supervisé ou non. Le nombre de caractéristiques utilisées est directement lié à l'erreur finale.

De ce fait, il devient fondamentale de traiter préalablement ces données. Ceci permettra une meilleure compréhension des informations contenues dans ces données et facilitera ainsi la tâche de classification. Une identification efficace des variables pertinentes s'avère indispensables pour le succès de la tâche de classification.

La réduction de dimension est alors la solution la plus intuitive pour contribuer à la résolution de ces problèmes dans le domaine de l'apprentissage automatique (machine learning). Elle procède soit par appliquer une sélection des attributs pertinents, soit par une extraction d'attributs. Les informations redondantes et non utiles seront ainsi contournées afin d'avoir une meilleure représentation des données.

Les principaux objectifs de la réduction de dimension peuvent être décrits par : [Guérif, 2006] :

- identification des attributs pertinents,
- amélioration de la tâche de classification,
- facilitation de la visualisation et de la compréhension des données,

- réduction de l'espace de stockage nécessaire,
- réduction du temps d'apprentissage et d'utilisation.

Toutefois, l'élimination de certaines informations redondantes ou peu pertinentes peut augmenter l'erreur de classification, vu que ces informations peuvent s'avérer informatives si on les utilise conjointement [Ferchichi et al., 2009b].
La réduction de la dimension reste un problème complexe. Elle se divise en deux grandes catégories : la sélection d'attributs et l'extraction ou la transformation des attributs comme le montre les figures 1.6 et 1.7.
La sélection est appropriée quand l'acquisition de mesures est coûteuse. Son objectif

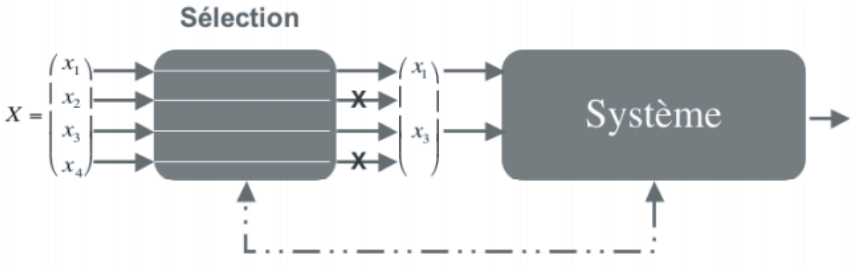

FIGURE 1.6 – Processus de Selection d'attributs extrait de [Guérif, 2006]

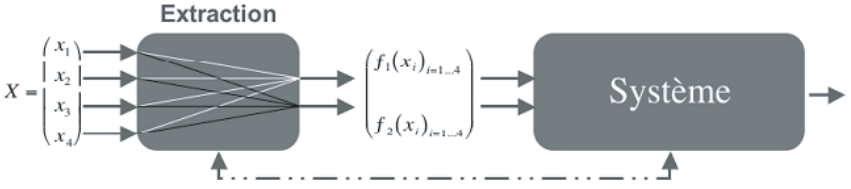

FIGURE 1.7 – Processus d'Extraction d'attributs extrait de [Guérif, 2006]

est de réduire le nombre de mesures requises et de choisir celles les plus informatives. Par contre, les techniques d'extraction (deuxième catégorie) utilisent la totalité de l'information contenue dans les attributs pour la combiner, la transformer et produire en final un vecteur de dimension plus petite. Ces techniques projettent un vecteur forme de l'espace de représentation dans un espace de dimension de re-description de plus petite dimension.

1.4.1 Sélection des variables

La sélection des variables est une thématique de recherche assez active depuis plusieurs décennies dans des domaines comme la reconnaissance des formes, la modélisation de séries chronologiques et l'identification de processus [Guyon and Elisseeff, 2003]. Elle constitue une étape importante dans le pré-traitement des données de grande dimension acheminées vers la classification supervisée ou non supervisée.
En effet, l'apparition des grandes bases de données dans le domaine de l'apprentissage et les systèmes de fouille de données "Data Mining" a exigé une réduction de dimension, avant d'entamer la tâche de classification vu les nombreux problèmes du fléau de dimensionnalité. Plusieurs travaux ont été proposés dans ce sens, et qui sont appliqués dans un grand nombre de domaines. De nouvelles techniques basées sur les machines à vastes marge ont été conçues pour aborder la tâche de sélection des variables [Grandvalet and Canu, 2002], .
La sélection des variables est un processus qui consiste à chercher dans l'ensemble des variables explicatives disponibles un ensemble optimal des caractéristiques les plus importantes à un système donné. Ceci est dans le but de mener à bien la tâche pour laquelle il a été conçu.
Définition (Sélection de variables) [Feraud et al., 1998] La sélection de variables est un procédé permettant de choisir un sous-ensemble optimal de variables pertinentes, à partir d'un ensemble original des variables, selon un certain critère de performance. En fait, le choix d'un ensemble optimal de descripteurs, ne signifie pas obligatoirement la sélection d'un ensemble composé seulement des variables jugées pertinentes et utiles. Il peut y contenir des variables non pertinentes, mais qui ont de meilleur performance, prises avec d'autres variables [Ferchichi, 2008].
Ainsi, la procédure de sélection de variables tente de sélectionner le sous ensemble le plus petit selon deux principaux critères :

- la précision de la classification ne se dégrade pas,
- la distribution des classes est proche de la distribution originale.

Idéalement, les méthodes de sélection d'attributs cherchent le sous-ensemble optimal de variables parmi les 2^N sous-ensembles candidats. Cependant, cette procédure est exhaustive, ce qui peut être très couteux en pratique. D'autre méthodes sont basées sur des procédures de recherche aléatoires et heuristiques en tentant de réduire la complexité. Ces méthodes ont besoin d'un critère d'arrêt pour éviter une recherche exhaustive.
Trois questions essentielles devraient se poser à ce stade afin de définir les éléments

essentiels d'une procédure de sélection de variables : [Ishak, 2007]

- Comment mesurer la pertinence des variables ?
- Comment former le sous-ensemble optimal ?
- Quel est le critère d'optimalité à utiliser ?

La pertinence d'une variable se définit selon deux critères. Soit son pouvoir discriminant permettant une meilleure séparation entre les différentes classes dans le cadre d'une classification. Soit son pouvoir prédictif garantissant une meilleure qualité de prédiction dans le cadre d'une régression. Il serait alors naturellement nécessaire de définir une mesure de pertinence ou un critère d'évaluation permettant de quantifier l'utilité d'une variable ou d'un ensemble de variables. De plus, dans une procédure de sélection, une procédure de recherche ou de construction du sous-ensemble optimal des variables pertinentes est nécessaire. Un critère d'arrêt peut être déterminé à travers une combinaison particulière entre la mesure de pertinence et la procédure de recherche.

Une procédure de sélection d'attributs peut être décortiquée en quatre étapes typiques illustrées dans la figure 1.8 [Dash and Liu, 1997].

- procédure de génération,
- fonction d'évaluation,
- critère d'arrêt,
- procédure de validation.

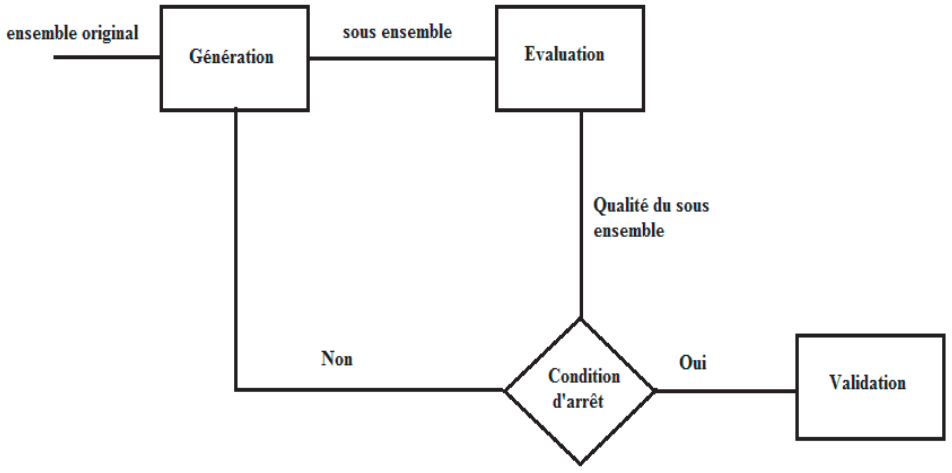

FIGURE 1.8 – Procédure de sélection avec validation

Trois grandes familles d'approches de sélection existent :

- Filtres : Ces méthodes sélectionnent les variables indépendamment du processus de classification. Ils se basent seulement sur les caractéristiques intrinsèques de

l'ensemble des données [Battiti, 1994] et [He et al., 2005].

- Wrappers : Au contraire des filtres, les méthodes wrappers n'ignorent pas l'influence de l'ensemble d'attributs sélectionnés sur la procédure de classification. Ces méthodes utilisent l'algorithme de classification comme enveloppe permettant d'évaluer le degré d'utilité des attributs pour le processus de la classification [Ferchichi et al., 2009a] et [Tahir, 2004].
- Embedded : Ces méthodes effectuent la sélection en parallèle avec la procédure de classification ou de la régression. Le sous-ensemble sélectionné sera choisi de façon à optimiser le critère d'apprentissage utilisé. Dans ce cadre, on retrouve les travaux dans [Grandvalet and Canu, 2002] et [Weston et al., 2000] qui utilisent la pondération des attributs pour aboutir à la sélection.

1.4.1.1 Procédure de génération

Le nombre optimal m de variables à sélectionner est en général inconnu et dépend de la taille, la qualité de la base d'apprentissage et la règle de décision utilisée. Pour un ensemble de n variables, il existe (2^{n-1}) combinaisons de variables possibles. Le nombre 2 représente les deux décisions possibles pour une variable : la sélectionner ou ne pas la sélectionner. Dans un espace d'entrée de très grande dimension, le nombre de combinaisons à examiner est très élevé et la recherche d'un sous ensemble optimal de variables devient un problème NP-difficile.

La procédure de génération n'est qu'une procédure de recherche. Elle génère un sous-ensemble d'attributs qui sera par la suite évalué selon un critère bien déterminé. La méthode de sélection procède par ajouts successifs (ou élimination successives) de variables. Elle peut être initialisée soit par un sous ensemble vide ou tout l'ensemble initial des attributs ou simplement avec un sous ensemble d'attributs aléatoire [Dash and Liu, 1997].

Pour chaque étape, la variable ou l'ensemble des variables qui améliore (ou dégrade) le critère d'évaluation sera sélectionné et rajouté (ou bien éliminé) à l'ensemble des variables déjà sélectionnées [Langley, 1994]. On retrouve les stratégies ascendantes (Forward Selection), descendantes (Bakward Selection) et bidirectionnelles (Bidirectional Selection) [Guérif, 2006]. Néanmoins, dans le cas d'une stratégie ascendante, les variables individuellement peu pertinentes peuvent être très informatives si on les utilise conjointement.

Vu le nombre exhaustif de sous-ensembles candidats d'attributs, les méthodes de génération peuvent être classées en trois approches :

1. génération complète : elle effectue une recherche exhaustive pour trouver l'ensemble optimal d'attributs sur tout l'espace des solutions possibles, qui est de l'ordre $O(2^N)$. Plusieurs procédures de recherche heuristique sont proposées afin de réduire le nombre des sous-espaces à évaluer.
2. génération séquentielle : à chaque itération de cette procédure, on considère à nouveau tout l'ensemble des attributs restants pour l'étape de la sélection. La génération des sous-ensembles est typiquement incrémentable (diminution/augmentation) dans un espace de recherche de l'ordre $O(N^2)$.
3. génération heuristique : bien que l'espace de recherche est de l'ordre $O(2^N)$, cette procédure n'évalue pas toutes les solutions possibles dans cet espace. Un nombre maximal d'itérations est imposé afin de limiter le temps de calcul. Plusieurs méthodes sont basées sur les algorithmes d'optimisation heuristique comme l'algorithme génétique ou recherche tabou [Ferchichi et al., 2009a].

1.4.1.2 Critères d'évaluation

L'amélioration des performances d'un système d'apprentissage par une procédure de sélection de variables nécessite dans un premier temps la définition d'une mesure de pertinence ou bien un critère d'évaluation. Typiquement, une fonction d'évaluation essaie de mesurer le pouvoir discriminant d'une variable ou d'un ensemble de variables pour discerner entre les différentes classes.
Par contre, pour un problème de régression, on teste plutôt la qualité de prédiction par rapport aux autres variables. La pertinence d'une variable (ou d'un ensemble de variables) peut être définie par :
Définition de pertinence d'une variable : [Bennani, 2001] Une variable pertinente est une variable telle que sa suppression entraîne une détérioration des performances du pouvoir de discrimination en classement ou la qualité de prédiction en régression du système d'apprentissage .
Toutefois, le choix d'un sous-ensemble optimal résultat de la procédure de recherche est relatif à la fonction d'évaluation utilisée. Ainsi, le changement du critère peut changer l'ensemble optimal en résultat.
Dés lors, plusieurs critères d'évaluation basés sur des hypothèses statistiques ou sur des heuristiques ont été proposées. Dans le cadre d'un problème de classification, les critères d'évaluation sont souvent basés sur les matrices de dispersion intra et inter classes, qui sont liées à la géométrie et la distribution des classes dans l'espace [Guérif, 2006]. D'autre critères d'évaluation utilisent des distances probabilistes

ou des mesures d'entropie basées sur l'information mutuelle entre les variables et les classes des observations [Slonim and Tishby, 1999].
Les méthodes de sélection peuvent être classées en deux grandes approches (les filtres et les wrappers), selon leur dépendance vis-à-vis l'algorithme inductif qui utilisera par la suite le sous-ensemble optimal des attributs [Kohavi and John, 1997]. Les méthodes filtres sont indépendantes de l'algorithme inductif, alors que les méthodes wrappers utilisent l'algorithme inductif comme une fonction d'évaluation. Selon [Dash and Liu, 1997], les fonctions d'évaluation peuvent être divisées en 5 catégories :

1. Mesure de distance : c'est une mesure de séparabilité, divergence ou bien mesure de discrimination comme par exemple la distance Euclidienne. Dans le cas d'une classification binaire, un attribut X est préféré à un autre attribut Y si X induit une plus grande différence entre la probabilité conditionnelle des deux classes en question.

2. Mesure d'information : cette mesure détermine l'information apportée par un attribut comme par exemple la mesure d'entropie. L'information apporté par un attribut X est déterminée comme étant la différence entre l'incertitude préalable et l'incertitude postérieur en utilisant X. Ainsi, un attribut X n'est préféré à un autre attribut Y que si l'information apportée par X est plus que celle apporté par Y.

3. Mesure de dépendance : c'est la mesure de corrélation qui peut qualifier la capacité de prédire la valeur d'une variable depuis une autre variable. Si la corrélation entre un attribut X et une classe C est supérieure à celle entre un attribut Y et la classe C, alors X est préféré à Y.

4. Mesure de consistance : il est proportionnel au pouvoir discriminant. Un sous-ensemble de variables ayant un taux d'inconsistance élevé signifie que ces variables ne permettent pas de bien prédire la classe et donc que ce sous-ensemble n'est pas un bon ensemble discriminant [Daviet, 2009].

5. Mesure d'erreur de classification : ce sont les méthodes wrappers qui utilisent ce type de fonction d'évaluation. Les attributs permettant d'améliorer l'erreur de classification sont sélectionnés. Ainsi, une grande précision de classification est garantie mais en revanche d'un calcul couteux en temps et mémoire [Ferchichi et al., 2009b].

1.4.1.3 Condition d'arrêt

Le nombre optimal de variables à sélectionner n'est pas connu a priori. Ce qui fait que l'utilisation d'une règle pour contrôler la sélection de variables (sous-ensemble de variables) permettra d'arrêter la recherche dans le cas où aucune variable (sous-ensemble de variables) n'est plus suffisamment informative.
Le critère d'arrêt est souvent défini en fonction d'une combinaison de la procédure de recherche et du critère d'évaluation. Prédéfinir un nombre maximal d'itérations à ne pas franchir est un critère d'arrêt assez commun. Cependant ce critère peut arrêter la recherche trop tôt ou bien à l'inverse trop tard. Un nombre maximal d'attributs peut aussi être utilisé comme critère d'arrêt. Cependant l'estimation du nombre optimal d'attributs n'est pas donnée préalablement. Définir le critère à base de la fonction d'évaluation est aussi envisageable. Dans ce cas, un seuil est fixé préalablement pour contrôler la variation de la fonction d'évaluation entre deux itérations consécutives.

1.4.2 Extraction de caractéristiques et des variables

Les méthodes d'extraction des caractéristiques construisent à partir des p variables originales, un nouveau ensemble contenant $d \leq p$ caractéristiques. Ces dernières contiennent la plus grande part possible de l'information initiale [Bekkerman, 2003]. Plusieurs variantes des méthodes existent et traitent l'extraction des variables.
Parmi les méthodes les plus connues, on trouve des méthodes linéaires telles que l'ACP (Analyse en Composantes Principales) [Pearson, 1901], MDS (Positionnement Multidimensionnel) [Hastie et al., 2001], LDA (Analyse Discriminante Linéaire) [Belhumeur et al., 1997] Des méthodes non linéaires ont été aussi développées comme ICA, l'ISOMAP [Lee and Verleysen, 2007], LLE [Saul and Roweis, 2000] et des versions non linéaire de ACP et LDA comme le Kernel PCA [Schölkopf et al., 1998], Kernel LDA [Mika et al., 1999].
En considérant un espace d'observations χ dans R^n et l'espace de caractéristiques H relié à l'espace d'observations par une application ϕ :

$$\Phi : \chi \rightarrow H \qquad x \rightarrow \phi(x) \tag{1.13}$$

L'ensemble des données d'apprentissage est constitué d'un ensemble fini de points x_i. Dans le cadre supervisé, cet ensemble sera modélisé par des couples (point, label) (x_i, y_i). La catégorisation des méthodes de réduction de dimension proposée dans [Lee and Verleysen, 2007], se base sur une qualification des objectifs de la méthode,

son modèle caché, le critère mathématique à optimiser et l'algorithme utilisé.

1. réduction de dimension sévère vs. souple : dans ce cas, la distinction se fait par rapport au taux de réduction. C-à-d entre la dimension initiale des données avant la réduction et la dimension finale après réduction. Dans le cas des problèmes modelés par un grand nombre de variables comme les images, la parole..., une réduction sévère est souhaitable, à l'inverse des problèmes modelés par une dizaine de variables.
2. modèle traditionnel vs générative : le modèle de chaque méthode réfère à la façon dont les variables latentes sont liées à celles observées. La plupart des méthodes se basent sur l'estimation des variables latentes à partir des observations. Les méthodes génératives essayent d'approcher plus la façon dont les données sont générées en modélisant les observations comme étant une fonction des variables latentes.
3. modèle linéaire vs. non linéaire : Les méthodes non linéaires sont plus puissantes que les méthodes linéaire. Elles arrivent à modéliser la liaison entre les observations et les variables latentes d'une façon plus riche. Cependant leur modèle engendre plusieurs paramètres qui requiert une grande masse de données pour son identification.
4. modèle discret vs. continu : La continuité est désirable si on a besoin de généraliser la réduction de dimension à d'autres points autres que ceux utilisés pour déterminer le modèle. Les méthodes continues comme l'ACP à l'inverse des méthodes discrètes comme la SOM (Self Organizing Map) [Wang et al., 2006], réalisent la réduction à partir d'une fonction paramétrée ou d'une projection entre l'espace de départ et l'espace final.
5. projection explicite vs implicite : un mappage implicite est défini par une fonction paramétrique comme pour le modèle de l'ACP qui définit un hyperplan. La généralisation dans ce cas est directe. Par contre, un mappage explicite associe une représentation dans l'espace de faible dimension à chaque point de données. La généralisation dans ce cas est difficile. Les paramètres de ce modèle sont les coordonnées et leur nombre est relié aux nombre d'observations.
6. estimation externe vs. intégrée de la dimension : La plupart des méthodes n'ont pas un estimateur de dimension intrinsèque intégré à part l'ACP. Cette fonctionnalité est assurée par une procédure externe.
7. plongement par couche vs. standalone : Les méthodes spectrales formulent le problème de la réduction de dimension en un problème de valeurs propres et

assemble les vecteurs propres pour former l'espace de projection. Ainsi, ajouter ou retirer une dimension ne nécessite pas un changement de coordonnées le long des autres dimensions. Par contre, d'autres méthodes calculent l'espace de projection pour une dimension spécifique. Si la dimension change, toutes les coordonnées doivent être re-calculées.

8. quantification vectorielle optionnelle vs. obligatoire : Quand la masse des données disponibles est très grande, on peut travailler avec un ensemble réduit d'observations représentatives. Ceci peut être réalisé en appliquant une quantification vectorielle, comme pour le SOM, qui consiste à remplacer les observations par un ensemble de centroides ou prototype tout en conservant la forme de distribution initiale des données.

9. algorithme en ligne vs. hors ligne : Selon le domaine de l'application, les observations peuvent arriver consécutivement ou bien alternativement. Dans le premier cas, on parle d'un système en ligne. Dans le deuxième cas, on parle d'un système hors ligne. Les algorithmes hors ligne ne peuvent pas travailler que si tout un ensemble d'observations est connu, à l'encontre des algorithmes en ligne (SOM).

10. optimisation exacte vs. approximée : Les algorithmes hors ligne sont le résultat d'un développement algébrique ou analytique qui donne une solution optimale comme l'ACP. Par contre, les algorithmes en ligne sont généralement associés aux procédures d'optimisation génériques comme la descente de gradient stochastique.

11. type du critère à optimisé : La réduction de dimension est motivée avec des considérations géométriques. Ainsi, les données sont interprétées comme étant une nuée de points dans un espace géométrique et assumées d'être liées à un plan ou une variété. Caractériser ce plan indépendamment du système de coordonnées, peut se faire en faisant apparaître la relation entre les points du plan. Ainsi, un bon critère sera de mesurer la proximité des observations pour conserver leur topologie initiale.

1.4.2.1 Méthodes linéaires

Nous rappelons brièvement les principes des trois méthodes classiques d'analyse de données, qui sont le fondement de plusieurs méthodes non linéaires plus récentes.

Analyse en Composantes Principales

L'analyse en composantes principales (ACP) projette les points originaux dans un sous-espace vectoriel de dimension plus réduite afin d'accomplir une réduction de dimension. En réalité, l'analyse en composantes principales a été introduite par Pearson, [Pearson, 1901], qui cherchait à approcher un "système de points dans l'espace" par un sous-espace linéaire de dimension plus petite. Plus précisément, Pearson étudiait le problème d'approcher des données multivariées par une droite telle qu'elle minimise la somme des écarts des points à la droite au carré. Le problème de l'ACP consiste à trouver le sous-espace affine E de dimension $d < p$. Souvent $d = 2$, tel que l'inertie J de l'ensemble des points du nuage par rapport à E soit minimale. L'inertie J est exprimé par [Bouveyron, 2006] comme suit :

$$J = \frac{1}{n}\sum_{j=1}^{n} \|x_j - P_E(x_j)\|^2, \tag{1.14}$$

Où $P_E(x_j)$ est la projection de x_j sur le sous-espace E.
Cela revient en général à rechercher les axes le long desquels la variance est maximale. Nous allons donc rechercher les axes qui maximisent la variance des vecteurs dans l'espace de projection. Le long d'un axe représenté par le vecteur unitaire u, la variance totale est :

$$V = \sum_{j=1}^{n}(u^t(x_j - \overline{x}))^2, \tag{1.15}$$

Où $\overline{x}$ est le barycentre des vecteurs $x_j, j = 1...n$.
Le vecteur propre qui réalise cette maximisation n'est que le vecteur propre associé à la plus grande valeur propre de la matrice de covariance $\widehat{\Sigma}_{totale} = \frac{1}{n}\sum_{j=1}^{n}(x_j - \overline{x})(x_j - \overline{x})^t$, noté respectivement μ_1 et λ_1. De même, dans l'espace orthogonal à μ_1, l'axe qui maximise la variance est supporté par le vecteur propre de $\widehat{\Sigma}_{totale}$, et ainsi de suite pour les p axes principaux. Les vecteurs μ_1 sont appelés les facteurs principaux. De même, les variables artificielles définies comme projections sur les facteurs principaux par la relation $c_j = X\lambda_j$, sont appelées *composantes principales*. Dans la base formée par ces axes, les coordonnées ne sont plus corrélées.
Le fait que $\widehat{\Sigma}_{totale}$ soit une matrice symétrique semi-définie positive nous assure que les valeurs propres sont toutes réelles, positives ou nulles et les vecteurs propres sont orthogonaux entre eux. De plus, la valeur propre $\lambda_l, l = 1, ..., d$, étant égale à la part de la variance totale portée par la composante principale associée. Cela permet de sélectionner les axes formant l'espace de projection. Il suffit alors de retenir les d

premiers vecteurs propres tels que $\sum_{j=1}^{d} \lambda_l$ représente une certaine proportion de la variance initiale (par exemple, 80%). Les composantes principales sont les combinaisons linéaires des variables initiales de variance maximale.

L'ACP étant une méthode de réduction de dimension. Il est important de savoir qu'on peut pas retenir la totalité de l'information contenue dans le nuage de points initial. Elle est très utilisée car elle est simple à mettre en oeuvre. Cependant elle est limitée par son caractère linéaire. En effet, l'ACP prend uniquement en compte les dépendances linéaires entre les variables et ne peut donc pas fournir une projection fidèle pour une distribution non linéaires de points (par exemple,des données réparties sur un tore en dimension n).

Combien de dimensions faut-il retenir ? Dans le cadre de l'ACP, la plupart des méthodes de recherche du nombre d'axes sont basées sur les valeurs propres de la matrice de covariance Σ des données. Ceci est justifié par le fait que les valeurs propres de Σ représentent la variance apportée par le vecteur propre associé.

Le critère empirique du *scree-test* de Cattell, [Cattell, 1966], est couramment utilisé. Il se base sur l'analyse des différences entre les valeurs propres consécutives et permet de détecter un "coude" dans la descente des valeurs propres. La dimension sélectionnée par la méthode est celle pour laquelle les différences entre les valeurs propres suivantes sont toutes plus petites qu'un certain seuil. La figure suivante illustre cette technique : La figure 1.9(a) montre les valeurs propres de Σ ordonnées

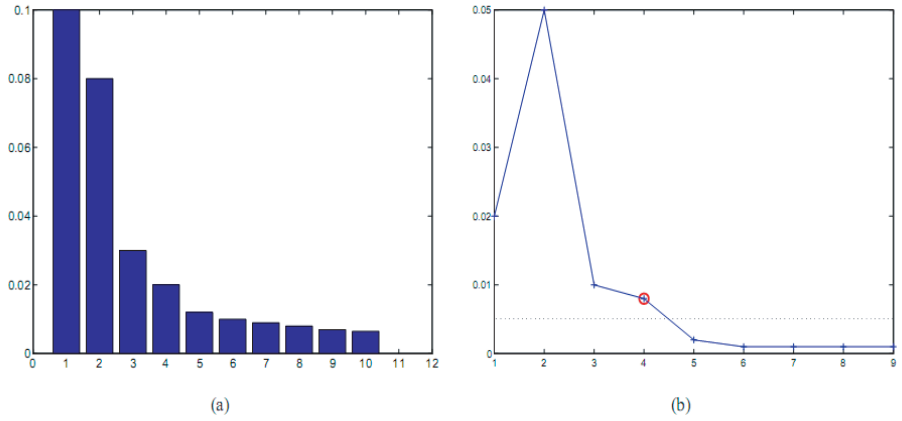

FIGURE 1.9 – Choix des composantes principales de l'ACP selon de test de Cattel [Bouveyron, 2006]

d'une façon décroissante, et la figure 1.9(b) montre les différences entre les valeurs propres consécutives. Dans cet exemple, le seuil a été fixé à 10% de la plus grande différence et le test de scree-test identifie un coude au niveau de la $4^{ème}$ dimension.

Analyse Discriminante de Fisher

À l'encontre des autres méthodes d'extraction, l'analyse discriminante de Fisher [Belhumeur et al., 1997] prend en considération la distribution des données par classes dans le cadre de la classification supervisée. Elle a été proposé par Fisher et appelée aussi analyse discriminante linéaire de Fisher ou encore analyse factorielle discriminante. Elle cherche un espace vectoriel de faible dimension qui maximise la variance inter-classe et minimise la variance intra-classe. En effet, dans le cas d'FLD ou LDA, on souhaite trouver une représentation des données qui permet de discriminer les classes le mieux possible. Il faut alors que les projections des k centres de gravité soient les plus séparées possible, tandis que les données de chaque classe doivent se projeter de façon groupée autour du centre de gravité de leur classe.
Supposons qu'on dispose d'un ensemble de données de cardinal n et de dimension p et qui appartiennent à k classes différentes. L'analyse factorielle discriminante consiste alors à projeter les données de $\mathbb{R}^p$ sur les $d = (k-1)$ axes discriminants. Ce sont les axes qui maximisent le critère J donné par le rapport entre la matrice de dispersion inter et intra classe :

$$\begin{aligned}
&J = \tfrac{w^t S_B w}{w^t S_w w}, \\
&avec \\
&S_B = \textstyle\sum_{i=1}^{k} n_i (m_i - m)(m_i - m)^t \\
&S_W = \textstyle\sum_{i=1}^{k} S_i; \\
&S_i = \textstyle\sum_{x \in D_i} (x_i - m_i)(x_i - m_i)^t; \\
&m = \tfrac{1}{n} \textstyle\sum_{j=1}^{n} x_k; \\
&m_i = \tfrac{1}{n_i} \textstyle\sum_{x \in D_i} x;
\end{aligned} \tag{1.16}$$

Ainsi, une direction w qui maximise $J(w)$ satisfait l'équation suivante :

$$S_B w = \lambda S_W w. \tag{1.17}$$

Si S_W n'est pas singulière, l'équation précédente devient un problème de valeurs propres :

$$S_W^{-1} S_B w = \lambda w. \tag{1.18}$$

On n'a qu'à chercher les valeurs et vecteurs propres de $S_W^{-1} S_B$. Le rang de S_B est égal à $k-1$ au maximum, donc on ne peut extraire que $k-1$ nouveaux attributs.
Cette stratégie qui combine la réduction de dimension et la discrimination, est souvent profitable. Elle est efficace pour des données de grande dimension et facilement

séparables. La figure suivante 1.10 représente la projection d'un jeux de données sur les axes principaux d'une part et sur les axes discriminants d'autre part.

Toutefois, cette méthode nécessite l'inversion de la matrice de covariance de classe

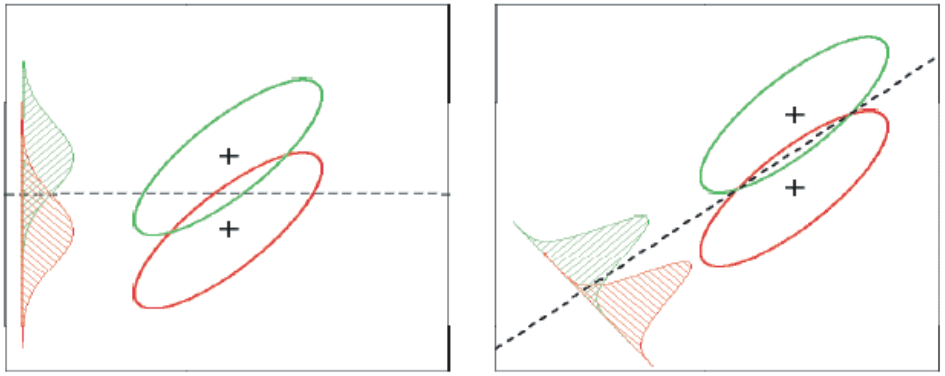

FIGURE 1.10 – La figure (a) présente les axes principaux déterminés par l'ACP et la figure (b) présente les axes discriminants obtenus par application de LDA, extrait de [Hastie et al., 2001]

S_W ce qui peut poser un problème si celle ci est mal conditionnée ou bien singulière.

Singularité des matrices de covariance de classe : Dans le cas où le nombre d'observations n est trop petit par rapport à la dimension de l'espace p, l'estimation de ces matrices sera singulière et leur inversion est impossible. Les $p-n+1$ plus petites valeurs propres sont alors estimées par 0. Les vecteurs propres correspondants sont alors arbitraires et le LDA ne pourrait pas être utilisée dans ce cas.

Mauvais conditionnement des matrices de covariance de classe : Dans ce cas, l'inversion de ces matrice entraînera un important biais sur le calcul de la règle de décision. Donc une importante erreur de prédiction. Un exemple dans le cas gaussien a été proposé par [Pavlenko and Rosen, 2001].

Analyse en composantes indépendantes

L'analyse en composantes indépendantes à l'objectif est de révéler des facteurs explicatifs qui soient indépendants au sens plein, et non plus seulement au sens de corrélation [Barbedor, 2006] [Aguilar, 2008]. Soit une variable aléatoire X sur $\mathbb{R}^d, d \geq 2$, telle que :

$$X = AS \tag{1.19}$$

Où A est une matrice carré inversible et S une variable aléatoire latente dont les composantes sont mutuellement indépendantes. On se propose d'estimer A, pour atteindre $S_1, ..., S_n$, à partir de la donné d'un échantillon $X_1, ..., X_n$ indépendant,

identiquement distribué. C'est à dire, X_i est indépendant de X_j pour $j \neq i$, mais les composantes $X_i^1, ..., X_i^d$ d'une même observation X_i ne sont pas à priori mutuellement indépendantes.
Cette formulation du ACI considère uniquement des superpositions linéaires de signaux indépendants, résultant du mixage par A.
C'est souvent une restriction légitime ; par exemple les systèmes de transmission sont des milieux linéaires où les signaux agissent comme s'ils étaient présents indépendamment les uns des autres, ils n'interagissent pas mais s'additionnent [Pierce, 1980].
En pratique, la transformation ACP de la matrice $d \times n, M = (x_1, ..., x_n)$ contenant le signal observé fournit la première partie de la réponse au problème au ACI, par décorrélation simple, et toute la réponse si les signaux mesurés, sont purement gaussien. Dans le cas contraire, pour résoudre le problème entièrement, la procédure usuelle consiste à minimiser une certaine fonction de contraste $C = C(W)$ qui s'annule si et seulement si les composantes de WX sont indépendantes, où W est une matrice $d \times d$ candidate à l'inversion de A.
D'autres formulations prennent en considération des mélanges dits post non linéaire [Taleb and Jutten, 1997]. Dans d'autre cas encore, on ne s'appuie pas sur l'indépendance, et on exploite au contraire une corrélation temporelle des signaux sources. Ils existent aussi des modèles convultifs, où le mixage par A n'est pas instantané, mais de la forme $X(t) = \sum_u A(u)S(t-u), A(u)$ désignant une suite de matrices inversibles [Castella et al., 2003].
Le problème ACI standard est toujours paramétrique en A. Il est paramétrique ou non en S suivant les hypothèses fonctionnelles appliquées à la densité de probabilité de S. Le modèle de densité de l'ACI s'écrit de la façon suivante :
soit f la densité de S par rapport à la mesure de Lesbegue, la variable observée $X = AS$ admet la densité f_A, définie par :

$$f_A(x) = |detA^{-1}|f(A^{-1}x) = |detB|f^1(b_1x)...f^d(b_dx), \tag{1.20}$$

où b_l est la ligne numéro l de la matrice $B = A^{-1}$; cette écriture résulte d'un changement de variable étant donnée que f, la densité de S, est le produit de ses marges $f^1, ..., f^d$.
Dans le modèle ACI ainsi exprimé, f et A sont les deux inconnus, et la donnée consiste en un échantillon indépendant et identiquement distribué $X_1, ..., X_n$ de f_A.
Principales approches du problème ACI : Plusieurs méthodes ont été proposées pour l'ACI. La plupart des méthodes classiques ne font pas d'hypothèses paramé-

triques pour f, mais n'entraînent pas pour autant la mise en oeuvre de méthodes typiquement non paramétriques [Barbedor, 2006].
Ces méthodes ont recours à des contrastes de substitution dont l'annulation n'implique pas exactement l'indépendance mutuelle. De point de vue théorique ils sont plus facile à estimer qu'un critère exact. Les méthodes mettant en avant leurs critères non paramétriques sont basées sur des contrastes exactes dont l'annulation implique l'indépendance mutuelles des composants.
Les méthodes classiques sont issues du maximum de vraisemblance et des contrastes basés sur l'information mutuelle ou autres mesures de divergence. On trouve quatre catégories de méthodes telles que :

- Maximisation du critère non gaussien,
- Maximum de vraisemblance,
- Méthodes tensorielle,
- Décorrélation non linéaire.

Les méthodes d'essence non paramétriques peuvent être classées en trois catégories :

- Méthodes à noyau : Kernel ICA,
- Méthodes à contraste exacte : Information mutuelle,
- Méthodes directes : Méthode de la fonctionnelle matricielle.

l'ACI généralise l'ACP dans le sens que les variables latentes sont non gaussiennes et A est autorisée à être une transformation non-orthogonale à l'encontre de l'ACP. L'ACP ne considère qu'une transformation orthogonale en assumant implicitement des composants gaussiens.

Positionnement MultiDimensionnel

Le principe de la famille des méthodes positionnement multidimensionnel ou spécifiquement Multi-Dimensional Scaling (MDS), est de construire une représentation en faible dimension des points, tout en conservant la distance originale entre eux.
Les premières versions de MDS ont été développées selon un modèle purement Euclidien. Par la suite des méthodes pour le MDS non métrique ont été élaborées, en focalisant sur l'information au lieu de la distance entre points. La distance entre un ensemble de points peut être ainsi mesurée selon la distance euclidienne ou une mesure de similarité plus sophistiquée. La première approche est typique pour un MDS métrique, alors que la dernière approche est typiquement non métrique [Verleysen and François, 2005].
MDS classique : La méthode MDS préserve le produit scalaire deux à deux au

lieu de préserver la distance deux à deux. Comme l'ACP, la méthode MDS se base sur un simple modèle génératif. Les observations y sont liés aux variables latentes x via la matrice de transformation W.

$$y = Wx, \tag{1.21}$$

Où les composants de x sont indépendants ou au moins décorrélés et W est une matrice $D \times P$ tel que $W^T W = I_p$. Les variables observées et les variables latentes sont assumées centrées.
Pour un ensemble fini de N points centrés, écrits sous la forme : $Y = [.., y(i), .., y(j), ...]$, le produit scalaire de $y(i)$ et $y(j)$ s'écrit sous la forme :

$$\begin{aligned} s_y(i,j) &= s(y(i), y(j)) && (1.22)\\ &= < y(i).y(j) > . && (1.23) \end{aligned}$$

et donc :

$$\begin{aligned} S[S_y(i,j)]_{1\leq i,j\leq N} &= Y^T Y && (1.24)\\ &= (WX)^T (WX) && (1.25)\\ &= X^T X. && (1.26) \end{aligned}$$

Á partir de l'équation précédente, les variables latentes peuvent être retrouvées tout en calculant la décomposition en valeurs propres de la matrice de Gram :

$$\begin{aligned} S &= U\Lambda U^T && (1.27)\\ &= (U\Lambda^{\frac{1}{2}})(\Lambda^{\frac{1}{2}} U^T) && (1.28)\\ &= (\Lambda^{\frac{1}{2}} U^T)^T (\Lambda^{\frac{1}{2}} U^T), && (1.29) \end{aligned}$$

Où U est une $N \times N$ matrice orthonormale et Λ est une matrice diagonale contenant les valeurs propres. Les valeurs propres sont ordonnés dans l'ordre décroissant, le vecteur p-dimensionnel des variables latentes peut être calculé en utilisant la formule suivante :

$$\widehat{X} = \mathbf{I}_{P\times N} \Lambda^{\frac{1}{2}} U^T. \tag{1.30}$$

L'équivalence entre ACP et MDS a été démontrée [Verleysen and François, 2005]. Ainsi, la méthode MDS possède tous les avantages et les inconvénients de la méthode PCA : méthode simple, robuste mais strictement linéaire. Hormis que MDS est plus

flexible que l'ACP (elle accepte le produit scalaire ou la distance euclidienne comme métrique), elle requiert plus de mémoire que l'ACP pour stocker la matrice de Gram $N \times N$ au lieu de la matrice de covariance $D \times D$.

1.4.2.2 Méthodes non linéaires

La plupart des méthodes linéaires se basent sur l'utilisation d'une distance Euclidienne ou d'un produit scalaire. Nous avons vu dans la section(1.3) que la distance Euclidienne n'a pas un grand sens dans le cas d'espace de très grande dimension. En particulier elle suppose que toutes les variables sont convenablement normalisées.
Dans le cadre de la théorie des espaces de Hilbert, il est devenu possible de définir d'autres produits scalaires qui sont basés sur des fonctions noyaux $k(x, y)$. k n'est qu'une mesure de similarité entre les points de l'ensemble de données, qui définit implicitement une application de l'espace d'origine vers un espace d'attributs H, de dimension infinie. Les méthodes basées sur les produits scalaires entre les données d'apprentissage peuvent se réécrire en version non linéaire. En remplaçant le produit scalaire habituel par un noyau k : c'est le truc du noyau (kernel trick) proposé initialement dans le cadre des machines à vecteurs support introduite dans [Vapnick, 1996].

Kernel ACP :

La méthode ACP est une méthode linéaire. La représentation obtenue en faible dimension est générée par des projections linéaires. Pourtant, les vecteurs et valeurs propres dépendent d'une façon non linéaire des données. Le kernel ACP a été proposé par [Schölkopf et al., 1998] pour appliquer l'ACP au cas de données situés sur une variété non linéaire en utilisant le truc du noyau.
Le calcul de l'ACP ne fait intervenir que des produits scalaires entre les points (pour le calcul de la matrice de covariance). L'idée du passage à la version non linéaire consiste dans le remplacement du produit scalaire par une fonction noyau.
Sur des données transformés tel que : $x \rightarrow \phi(x) \in F$, avec F est un espace de Hilbert appelé espace d'attribut. Il existe une fonction kernel $k(x_i, x_i)$, tel que :$\forall x_i, x_j \in \mathbf{R}^d, k(x_i, x_i) = \phi(x_i).\phi(x_j)$.
Puisque la méthode dépend seulement du produit scalaire, on n'a pas à calculer $\phi(x)$ explicitement, on peut juste la remplacer avec la forme kernel. On calcule donc les composantes principales dans l'espace des caractéristiques F, et on peut ainsi accéder à des corrélations d'ordre supérieur entre les variables observées.

Supposons qu'on a un ensemble de données, la matrice de covariance C est donnée par :

$$C = \frac{1}{m}\sum_{j=1}^{m}(x_j - \mu)(x_j - \mu)^t \tag{1.31}$$

l'ACP cherche à diagonaliser cette matrice en résolvant l'équation des valeurs propres $Cv = \lambda v$. Dans l'espace de projection F, la matrice de covariance des données projetées s'écrit sous la forme suivante :

$$\widehat{C} = \frac{1}{m}\sum_{j=1}^{m}(\phi(x_j) - \mu)(\phi(x_j) - \mu)^t \tag{1.32}$$

avec $\phi_i \equiv \phi(x_i)$ et $\mu = \frac{1}{m}\sum_i \phi_i$. On cherche ainsi les valeurs et vecteurs propres solutions de v :

$$\widehat{C}v = \lambda v \Longleftrightarrow (\phi_i - \mu)^t \widehat{C}v = \lambda(\phi_i - \mu)^t v \tag{1.33}$$

avec $v = \sum_i \alpha_i(\phi_i - \mu)$.
Considérons la matrice kernel K_{ij}, matrice de covariance dans F tel que :

$$K_{ij} = k(x_i, x_j) \equiv \phi_i.\phi_j, i, j = 1...m. \tag{1.34}$$

On doit effectuer un centrage de K_{ij} par une matrice de projection $P = \mathbf{I} - \frac{1}{m}ee'$ avec $\mathbf{I}$ est la matrice unité et e est le vecteur de taille m ne contenant que des $\mathbf{1}$. Ainsi, on obtient $K^c = PKP$ et l'équation 1.33 devient :

$$K^c K^c \alpha = \widehat{\lambda} K^c \alpha \tag{1.35}$$

avec $\alpha \in \mathbf{R}^m$ et $\widehat{\lambda} = m\lambda$.
Toute solution de $K^c\alpha = \widehat{\lambda}\alpha$ est une solution de 1.33 plus un vecteur β orthogonal à α et qui satisfait : $\sum_i \beta_i(\phi_i - \mu) = 0$ et qui ne contribue pas à v. On doit les normaliser pour avoir une longueur de $\frac{1}{\sqrt{\widehat{\lambda}}}$.
Ainsi, pour extraire le $i^{éme}$ composant principal dans F en utilisant le KACP, on doit procéder par ces étapes :

1. Calculer le $i^{éme}$ vecteur propre principale de K^c, avec les valeurs propres $\widehat{\lambda}$
2. Normaliser les vecteur propre α correspondant pour avoir une longueur de $\frac{1}{\sqrt{\widehat{\lambda}}}$
3. Pour un vecteur d'apprentissage x_k, le composant principal est donné par :

$$(\phi(x_k) - \mu).v = \widehat{\lambda}\alpha_k \tag{1.36}$$

4. Pour un vecteur de test x, le composant principale est :

$$(\phi(x_k) - \mu).v = \sum_i \alpha_i k(x, x_i) - \frac{1}{m} \sum_{i,j} \alpha_i k(x, x_j) \tag{1.37}$$

$$-\frac{1}{m} \sum_{i,j} \alpha_i k(x_i, x_j) + \frac{1}{m^2} \sum_{i,j,n} \alpha_i k(x_j, x_n) \tag{1.38}$$

Le KACP applique ainsi l'ACP dans l'espace d'arrivée F de dimension plus grande que d. Les projections utiles peuvent ainsi excéder la dimension initiale d. Le KACP focalise essentiellement sur l'extraction des attributs plus que la réduction de dimension. On peut calculer la projection d'un point ne faisant pas partie de l'ensemble d'apprentissage, contrairement à d'autres méthodes de réduction de dimension non linéaires.

Isomap :

C'est la technique la plus simple pour la réduction non linéaire de la dimension. Elle utilise la distance de graphe pour approximer la distance géodésique. La seule différence entre la méthode MDS et la méthode Isomap est que celle ci utilise la distance de graphe au lieu de la distance Euclidienne, dans la procédure algébrique de la métrique MDS. L'introduction de la distance de graphe, transforme la méthode linéaire MDS en une méthode non linéaire. En fait, pour les points proches, la distance Euclidienne est une bonne approximation de la distance géodésique sur la variété. Un graphe est ainsi construit reliant chaque point à ses k plus proches voisins. La distance géodésique est estimée en cherchant la longueur du plus court chemin entre deux points dans le graphe. On peut alors appliquer MDS aux distances obtenues afin d'obtenir un positionnement des points dans un espace de dimension réduite.
Le modèle MDS est restreint à la projection sur un hyperplan. La matrice D de distance deux à deux, utilisée dans MDS doit contenir que la distance euclidienne mesurée entre des points liés à un hyperplan. Ainsi, si la distance dans D n'est pas Euclidienne, il est implicitement assumé que la métrique de remplacement est équivalente à la distance Euclidienne mesurée dans un hyperplan transformé. Dans le cas de la méthode Isomap, lors du calcul de la matrice D, la distance Euclidienne est remplacée par une distance de graphe. Il est plus simple ainsi d'assumer que la distance de graphe approxime parfaitement la distance géodésique réelle. Ainsi,

les données s'accordent au modèle d'Isomap si la distance géodésique calculée entre les points deux à deux peuvent être projetées à une même distance dans un espace Euclidien.
L'Isomap procède comme la méthode MDS, sauf que pour le calcul de la distance de graphe, les données doivent être stockées sous forme de coordonnées dans une matrice Y. Isomap n'a pas de procédure d'interpolation pour les données de test.

Algorithm 1 ISOMAP algorithm

1: construire un graphe avec soit la k-règle ou ϵ-règle
2: pondérer le graphe en donnant un label à chaque arc avec sa longueur Euclidienne
3: calculer toutes les distance de graphe deux à deux avec l'algorithme de Dijkstra, les mettre au carré et les stocker dans une matrice D
4: convertir la matrice D en une matrice de Gram S avec un double centrage
5: calculer la décomposition spectrale de la matrice de Gram, $S = U\Lambda U^t$
6: obtenir une P-représentation de Y en calculant le produit $\widehat{X} = \mathbf{I}_{P\times N}\Lambda^{\frac{1}{2}}U^t$

Les données sont groupées ensemble et traitées hors ligne. Toutefois, il est possible d'adapter la même procédure d'interpolation de la méthode MDS, décrite dans [Verleysen and François, 2005].

Plongement localement linéaire :

C'est une méthode qui préserve la topologie des données au lieu de la distance deux à deux. D'autres méthodes de la même catégorie essayent de préserver la topologie en gardant les points voisins proches les uns des autres. En d'autres termes, ils traduisent concrètement la notion qualitative de la topologie en des proximités relatives [Verleysen and François, 2005].
La méthode de plongement localement linéaire (LLE) propose une autre approche basée sur une transformation conforme, c'est une transformation qui conserve les angles. La préservation des angles et la préservation des distances sont liées. Ceci peut être interprété comme deux différentes approches pour préserver le produit scalaire.
La première tâche du LLE est de déterminer les angles à prendre en considération. Pour ce faire, LLE sélectionne un couple de voisins pour chaque point $y(i)$ dans l'ensemble de données $\mathbf{Y} = [..., y(i), ..., y(j), ...]_{1\leq i,j\leq N}$. On procède par associer à chaque $y(i)$ soit les k plus proches voisins ou bien tous les points appartenant à une cercle centrée en $y(i)$. Si l'ensemble des données est suffisamment large et n'est

pas trop bruité, alors on peut assumer qu'il existe une valeur ϵ tel que la variété peut être approximé linéairement dans l'échelle locale des k-voisins. Par la suite, chaque point $y(i)$ sera remplacé par une combinaison linéaire de ses voisins. Ainsi, la géométrie locale de la variété sera caractérisée par des coefficients linéaires qui reconstruit chaque point de donnée à partir de ses voisins [Saul and Roweis, 2000]. L'erreur totale de reconstruction peut être mesuré par :

$$\varepsilon(W) = \sum_{i=1}^{N} \|y(i) - \sum_{j \in N(i)} w_{i,j} y(j)\|^2 \tag{1.39}$$

Avec $N(i)$ est un ensemble contenant tous les voisins des points $y(i)$, $w_{i,j}$ sont les poids des voisins dans la reconstruction de $y(i)$ et les entrées la matrice W de dimension $N \times N$. Pour calculer ces coefficients, une fonction coût est optimisée sous les contraintes :

- mettre les coefficients $w_{i,j}$ pour les points en dehors du voisinage de $y(i)$ à zéro : $w_{i,j} = 0 \forall j \notin N(i)$
- la somme des lignes de la matrice de coefficients est égale à un : $\sum_{j=1}^{N} w_{i,j} = 1$.

1.5 Conclusion

Au cours de ce chapitre, nous avons présenté la problématique de la sélection de variables et de l'extraction des caractéristiques pour les problèmes de classification. Nous avons rappelé les principes de quelques méthodes de base comme l'ACP, l'ICA et le LDA. En effet, les outils d'analyse de données moderne doivent travailler dans un espace d'entrée de grande dimension. De plus, ses composants ne sont pas indépendants. Les espaces de grande dimension exhibent des propriétés assez surprenante et qui vont à l'encontre de l'intuition géométrique et qui ont une grande influence sur les outils dédiés à l'analyse de données. Comme le phénomène de la concentration de norme. En effet, la norme Euclidienne et les noyaux gaussiens, qui sont généralement utilisées dans ces modèles, deviennent inappropriés dans un espace de grande dimension [Verleysen and François, 2005].

Les techniques d'extraction non supervisée sont soit limités par leur caractères linéaires (ACP, MDS), soit difficilement utilisables à cause de leur complexité algorithmique(LLE, Isomap). Le caractère non supervisé engendre plus de complexité sur un problème de classification car aucune information n'est disponible pour guider la procédure. L'extraction non supervisée repose sur des mesures de similarité entre

attributs ou sur des mesures de variance. Néanmoins, elle n'est pas prédisposée au sur-apprentissage.

Chapitre 2

Approche de classification non supervisée pour l'extraction d'attributs

Contents

2.1 Introduction

Dans ce chapitre, nous présentons une nouvelle approche d'extraction d'attributs ou caractéristiques qui se base sur une classification non supervisée de ces attributs.

Notre approche fait appel à une méthode de clustering pour réaliser la classification d'attributs. Dans une première partie, nous commençons par expliquer le principe de la classification d'attributs. Par la suite, nous allons exposer le principe ainsi que la formulation de notre méthode d'extraction en explicitant la mesure de similarité conçue à base de l'analyse de tendance. Un aperçu sur l'analyse de la tendance et de la distance "Value Distance Metric" (VDM), utilisée pour la conception de notre mesure de similarité, sera exposée. La formulation de la nouvelle mesure de similarité sera décrite après. Nous finissons cette partie par la présentations de l'algorithme du clustering.
La deuxième partie de ce chapitre sera consacrée à l'introduction de l'Information Mutuelle (IM). Elle sera utilisée comme critère de recherche de la meilleure transformation à appliquer sur les clusters d'attributs. Le but étant de trouver un centre qui sera représentatif des attributs dans le cluster correspondant. Ces centres seront considérés comme les nouveaux attributs pour décrire les données d'origine. L'IM entre les attributs d'un cluster et son centre sera ainsi optimisée. Le but est de trouver les scores adéquats à appliquer à l'ensemble d'attributs du même cluster.

2.2 La classification pour l'extraction des variables

La classification non supervisée ou bien le Clustering, consiste à affecter chaque observation à une classe sans avoir recours à aucune information sur la distribution des données de l'apprentissage.
Le clustering cherche ainsi à identifier les groupes des observations similaires dans l'ensemble des données [Luxburg, 2007]. Par la suite, elle remplace un groupe de variables "similaires" par le centroïde du cluster.
Pour ce faire, le clustering se base sur la définition d'une mesure de similarité qui lui permet d'identifier les liens entre les observations et de les classer en c classes. La notion de similarité s'avère ainsi une étape critique pour les approches de classification non supervisée.
Cette approche s'avère intéressante dans le cadre de l'extraction ou la construction des variables. En effet, le clustering peut être utilisé pour identifier les partitions dans l'espace d'attributs. Selon la présence de relation et de similarité, les classes d'attributs seront alors formées par les éléments qui ont une similarité de comportement et qui peuvent être ainsi regroupés. Chaque centroïde de groupe obtenu devient un attribut lui même [Guyon and Elisseeff, 2003]. Les algorithmes les plus populaires

incluent la k-moyenne et le clustering hiérarchique.
Le clustering est en général associé au problème d'apprentissage non supervisé, ceci n'empêche qu'une forme de supervision peut être introduite pour obtenir des attributs plus discriminants pour le classifieur. Le clustering distributionnel a été développé dans ce cadre [Pereira et al., 1993], en se basant sur la théorie de "Information Botteleneck (IB)" de [Slonim and Tishby, 1999].
En considérant $\widehat{X}$ la variable aléatoire représentant les attributs extraits, et X l'ensemble original des attributs et Y les labels de chaque observation. La méthode IB cherche à minimiser l'information mutuelle entre X et $\widehat{X}$ donnée par $I(X, \widehat{X})$ tout en préservant $I(\widehat{X}, Y)$. Une fonction objective globale est construite en introduisant les multiplicateurs de Lagrange β :

$$J = I(X, \widehat{X}) - \beta I(\widehat{X}, Y) \tag{2.1}$$

La méthode cherche ainsi à réaliser une compression maximale tout en gardant l'information essentielle à propos des classes des observations.
La technique de clustering est généralement utilisée pour l'extraction textuelle [Bekkerman et al., 2
[Baker and McCallum, 1998], [Dhillon et al., 2003], [Cleuziou, 2004], où les observations sont les textes et les attributs représentent les mots. Chaque attribut ou variable est associée ainsi à un mot et est proportionnel à la fraction du document dans laquelle elle apparaît.
Dans ce contexte d'extraction textuelle, la méthode clustering a été utilisé pour grouper les mots et former alors des "groupes de mots". Une forme de supervision est introduite en exploitant les catégories des documents. L'ensemble des variables, contenant le compte de la fréquence de l'apparition de chaque mot dans les documents sera remplacé par un vecteur de variables plus court qui contient le compte de la fréquence d'appartenance des documents par catégorie. Les mots seront ainsi représentées par leurs distributions tout au long des catégories ou classes des documents.
Dans [Baker and McCallum, 1998] et [Bekkerman et al., 2003], le clustering distributionnel a été appliqué pour l'extraction d'attributs dans le cadre de traitement de texte. L'auteur dans [Baker and McCallum, 1998] argumente que si deux mots différents "vote" semblablement parmi les différentes possibilités, alors joindre ces deux mots n'affectera pas les performances. Le clustering des mots peut ainsi améliorer les performances spécifiquement pour les données parcimonieuses. La similarité entre les mots est mesurée par la distribution des labels des classes associées aux

mots en question. Les groupes sémantiques ainsi produits sont dépendants des labels des classes : les mêmes mots sont synonymes dans un contexte déterminé et non pas dans un autre.

L'algorithme du clustering définit une mesure de similarité entre les mots et réduit les mots similaires en un seul événement, qui ne distingue plus entre ses constituants. Considérons une variable aléatoire le long des classes C et sa distribution par rapport à un mot donné w_t. On appelle cette distribution $P(c/w_t)$. Quand deux mots w_s et w_t seront groupés ensemble, la nouvelle distribution sera représentée par la moyenne pondérée des distributions individuelles donnée par :

$$P(c/w_t \wedge w_s) = \frac{P(w_t)}{P(w_t) + P(w_s)} P(c/w_t) + \frac{P(w_s)}{P(w_t) + P(w_s)} P(c/w_s) \qquad (2.2)$$

Dans le contexte de la classification des texte, le clustering distributionnel mesure la similarité entre deux mots w_t et w_s comme étant la similarité entre la distribution des variables par classe induite : $P(c/w_t)$ et $P(c/w_s)$ [Baker and McCallum, 1998]. Les auteurs de [Dhillon et al., 2003], ont présenté une implémentation simple en utilisant l'algorithme K-moyenne. Une mesure de similarité a été conçue qui est basée sur l'analyse des distributions des mots sur les classes de documents, dans le corpus d'entraînement. L'algorithme est basé sur l'optimisation d'un critère global d'information mutuelle, inspiré de la méthode Information Bottleneck. Dans [Bekkerman et al., 2003], les auteurs ont implémenté une combinaison du clustering distributionnel des mots et le classificateur SVM pour la catégorisation des textes. Le clustering distributionnel à base de la méthode "Information Bottelneck (IB)" génère une représentation efficace et compacte des documents. Combinée avec l'SVM, elle produit de très bonne performance pour la catégorisation des textes par rapport à la représentation "sac-à-mots", spécifiquement pour les problèmes complexes. Un autre travail qui porte sur ce thème est [Cleuziou, 2004].

Dans [Bonet et al., 2006], l'approche du clustering a été appliqué pour l'extraction d'attributs dans une séquence de protéine pour l'analyse d'ADN. Il s'agit d'un problème important dans la bio-informatique. Une représentation commune utilisée est basée sur la séquence : une séquence primaire est généralement décrite par un attribut pour chaque amino-acide, ce qui produit plusieurs attributs. En considérant par exemple un problème avec une séquence de longueur n et 20 amino-acides, on se retrouve donc à 20^n attributs possibles.

Le clustering implémenté dans [Bonet et al., 2006] est basé sur l'algorithme du k-moyenne combiné avec le SVM pour la classification finale. Une mesure de similarité

biologique basée sur le contact d'énergie de l'amino-acide et la position dans la séquence.
Le majeur inconvénient dans les méthodes déjà exposé réside dans l'utilisation de la mesure de similarité. D'une part il y a des approches qui ont développé une mesure spécifique à l'application comme dans le cas de [Bonet et al., 2006]. D'autre part, les approches d'extraction textuelles ont développé une mesure de similarité à base de l'information mutuelle. IM est une mesure assez puissante mais le problème de son estimation consomme un temps énorme et une grande complexité qui augmente avec le nombre des variables. Toutefois, des formulations d'estimation non paramétrique développées dans la littérature ont résolu ce problème.

2.3 Approche d'extraction par classification des attributs

2.3.1 Principe et formulation

2.3.1.1 Principe

Dans cette section, nous introduisons notre méthode pour l'extraction d'attribut à base de la classification non supervisée [Ferchichi et al., 2011]. Elle vise à réaliser une extraction d'attributs sans prise en compte d'informations concernant la structure des données ni leurs appartenances. Nous nous focalisons plutôt sur l'identification des clusters d'attributs śimilaires ́dans leur comportement le long de la base de données. L'analyse de la tendance de chaque attribut peut révéler une certaine similarité entre eux et induit à la connaissance de l'une des formes de la redondance d'informations dans une base de données.
En réalité, la redondance d'informations est une caractéristique intrinsèque des données de grande dimension, qui complique la tâche d'apprentissage et dégrade les performances du classificateur [Ripley and Hjort, 1995]. Identifier les attributs redondants, nous permet de réduire la dimension sans perte d'information importante ni de pouvoir discriminant.
Ce problème est généralement traité ou bien par une approche de sélection ou par une approche d'extraction des variables [Guyon and Elisseeff, 2006], [Guyon and Elisseeff, 2003]. La première approche se base généralement sur des filtres qui éliminent totalement les attributs non-informatives. Pourtant, ces attributs jugés redondants ou non perti-

nents, peuvent interagir implicitement avec les autres attributs. Ils peuvent contenir des informations qui groupées avec d'autres contribuent effectivement à la tâche de discrimination [Ferchichi et al., 2009b]. Ainsi, l'élimination totale de ces attributs risque de faire virer, de l'espace d'attributs, des informations utiles et induire ainsi un classificateur moins performant. Notre approche proposée pour l'extraction cherche à contenir l'information qu'apportent ces attributs. Nous nous basons sur une mesure de similarité qui exploite l'analyse de la tendance pour pouvoir identifier les interactions internes dans l'espace d'attributs.
Le Clustering est en effet une approche très efficace pour découvrir la structure inhérente des données [Fern and Brodley, 2006], [Luxburg et al., 2007]. Son but n'étant pas de trouver la meilleur partition d'un ensemble d'exemples donné mais plutôt d'approximer la partition réelle de l'espace à travers la minimisation d'une fonction coût.
Nous proposons ainsi d'exploiter les techniques de Clustering, afin d'identifier des groupes d'attributs similaires et les transformer. Pour obtenir des représentants ou centroïdes qui seront les nouveaux attributs utilisés pour décrire l'ensemble des observations.

2.3.1.2 Formulation

Le processus d'extraction d'attributs consiste à construire un nouveau ensemble d'attributs à partir de l'ensemble original. Ces nouveaux attributs devraient représenter complètement les observations afin de faciliter la tâche de classification et améliorer sa qualité.
L'extraction d'attributs consiste à appliquer une transformation T sur l'ensemble composé de L observations de dimension D, $X \in R^{D \times L}$ et tel que :

$$y_i = T(x_i); 1 < i < L. \tag{2.3}$$

Avec $y_i \in R^{d<D \times 1}$ représente l'observation x_i décrite dans le nouveau espace d'attributs composé de d nouveaux attributs.
Notre approche procède par appliquer le clustering sur l'espace d'attributs original composé de D attributs $v_1, ..., v_D$, chacun de dimension L. La distance entre deux vecteurs d'attributs v_i et v_j doit être définie d'une façon appropriée et fidèle à nos objectifs. Ceci est dans le but que l'algorithme de clustering puisse trouver la partition optimale. Ainsi, on a besoin de définir une mesure qui puisse détecter la similarité

entre le comportement de deux attributs donnés.
La distance développée dans la section suivante 2.3.2 constitue la métrique sur laquelle se base notre algorithme. Elle exploite le concept d'analyse de tendance pour pouvoir définir une signature de chaque attribut et la comparer par la suite avec les autres signatures et conclure sur les similitudes entres eux.
Finalement, l'algorithme du clustering guidé par la nouvelle mesure de similarité identifie les groupes d'attributs similaires et nous donne la partition optimal. Ainsi, $d < D$ groupes d'attributs ou clusters $C_{k\in\{1,..,d\}}$ sont formés. Chacun sera représenté par son centre ou centroïde correspondant donné par :

$$g_k = f(S_k) = \sum_{i=1}^{L}\sum_{s=1}^{n_k} w_i v_s \tag{2.4}$$

Avec S_k est l'ensemble de n_k vecteurs attributs appartenant au cluster C_k et f est la transformation appliquée à l'ensemble des attributs dans C_k , définie par les poids élémentaires w_i attribuées à chaque attribut appartenant à C_k.
L'ensemble de centroïdes obtenu $\{g_k\}_{k\in\{1,..,d\}}$ représente alors l'ensemble des d nouveaux attributs qui vont re-décrire l'ensemble original des L observations de la base de donnée.

2.3.2 Mesure de similarité

2.3.2.1 Choix d'une mesure de similarité

Vu que la similarité est une notion fondamentale pour la définition d'un cluster, la détermination d'une mesure de similarité entre deux objets est alors essentielle pour la plupart des procédures de classification non supervisée [Mao and Jain, 1996]. De plus, la variété des types des attributs fait que cette mesure doit être choisie avec précaution [Mao and Jain, 1996].
Dés lors, le groupement des individus similaires dans des clusters nécessite la définition d'une métrique qui mesure la similarité entre individus selon les objectifs poursuivis.
Il s'agit en effet de trouver la meilleure adéquation entre le but qui est poursuivi et le comportement effectif de la mesure. Or ce comportement est extrêmement variable comme l'illustre la figure suivante (similarité) [Bisson, 2000].

Parmi les distances les plus populaires, nous trouvons la distance Euclidienne (Minkowski), utilisée généralement pour évaluer la proximité des objets en deux et trois

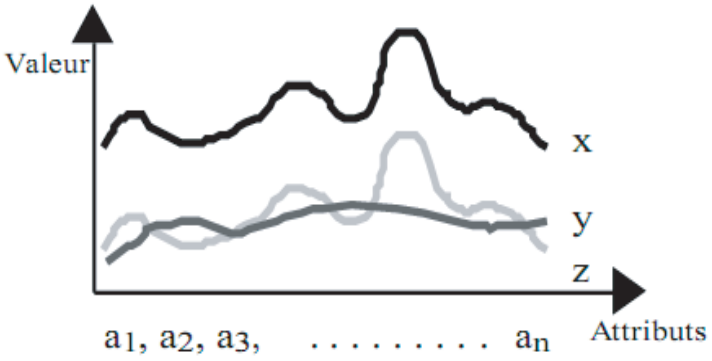

FIGURE 2.1 – Similarité entre attributs extrait de [Bisson, 2000]

dimensions. Elle est efficace pour des données dont la représentation est "compact" ou représentent des clusters isolés [Jain et al., 1999]. Son inconvénient que représente les métriques de Minkowski, réside dans la tendance de certaines données les plus calibrées à dominer les autres. La normalisation des données est ainsi nécessaire pour remédier à ce problème. En plus, la corrélation linéaire entre les données peut distordre les mesures. Cette distorsion peut être alors réduite en appliquant une transformation de "whitening" ou en utilisant la distance Mahalanobis. Cette distance assume implicitement que les densités conditionnelles des classes soient des Gaussiennes multidimensionnelles [Mao and Jain, 1996]. La distance du chi2 (χ^2) par contre, compare le profil global. Elle compare les variations relatives des attributs les uns par rapports aux autres. Ainsi, le χ^2 est une distance plutôt adaptée à la comparaison de profils de distribution de modalités [Bisson, 2000].

Ces distances sont efficaces en général pour des attributs numériques, mais ne peuvent pas manipuler les données proprement symboliques (non linéaire et non ordonnées) [Wilson and Martinez, 1996]. La distance VDM, par exemple, compare non pas le profil des valeurs, mais le profil de leur distribution dans les classes. Elle favorise ainsi les attributs possédant les valeurs les plus discriminantes (typiques d'une classe) [Bisson, 2000].

D'autres distances ont été conçu selon les différents objectifs considérés. Par exemple dans [Mao and Jain, 1996], la distance développée cherche à caractériser une similarité entre deux points en considérant l'effet des données voisines qui les entourent

.

2.3.2.2 Analyse de la tendance

L'analyse de la tendance est une approche efficace pour l'extraction en ligne des informations depuis des données numériques et sa représentation symbolique de façon qualitative ou semi-qualitative. Elle se trouve utile particulièrement pour le diagnostic et la surveillance des procédés industriels spécifiquement les procédés chimiques. En effet, l'apparition d'un défaut se traduit par une forme temporelle particulière appelée signature de défaut prise par une ou plusieurs variables comme le montre la figure 2.2.

Une procédure de détection en ligne est capable de reconnaître l'occurrence de la

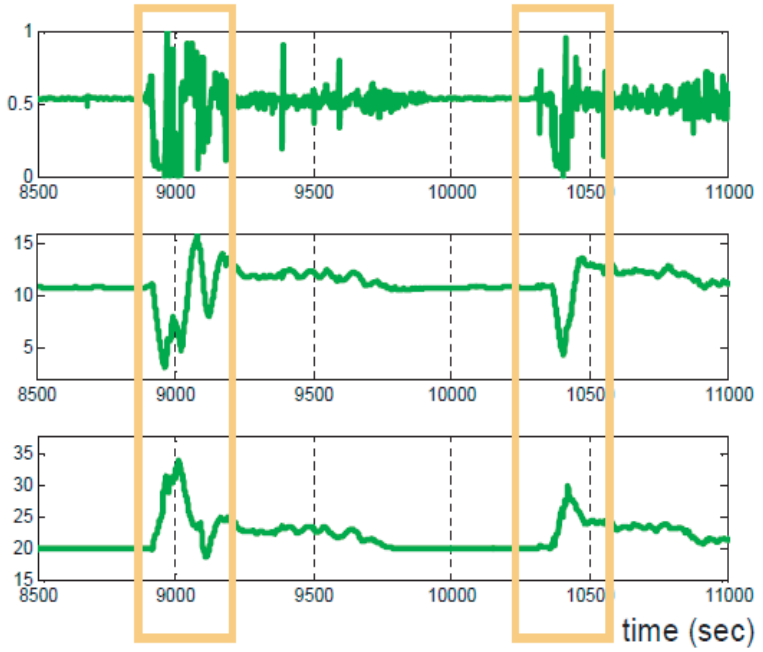

FIGURE 2.2 – Détection de défaut dans un signal par analyse de tendance

forme temporelle sur l'ensemble des signaux affectés par le défaut.

La tendance est composée d'une suite continue d'épisodes de longueur variable, qui sont consécutifs et qui ne se chevauchant pas. Une épisode est un intervalle de temps, défini par un instant de début et un instant de fin, à laquelle est associée une forme, appelée primitive. Cette primitive est choisie parmi une bibliothèque de formes décrivant la dynamique du signal. Elle a l'avantage de fournir une représentation explicite de l'évolution temporelle, la mise en évidence de transitions pertinentes et la compression des informations comme l'illustre la figure 2.3.

Plusieurs méthodes d'analyse de la tendance existe dans la littérature, qu'on peut classer en deux classes, selon la complexité des primitives associées [Charbonnier and Gentil, 2007].

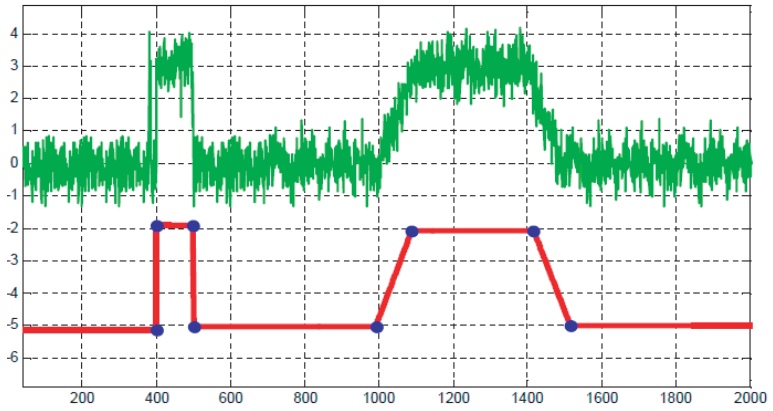

FIGURE 2.3 – Siganture d'un signal par analyse de tendance

Celles qui se basent sur des formes de premier et second ordre et celles qui se basent seulement sur des formes du premier ordre. Ceci peut être réalisé par une dérivation numérique du signal, ce qui nécessite un filtrage à priori, ou bien par la reconnaissance de forme en utilisant les réseaux de neurones.
La tendance décrite par des primitives de premier ordre est extraite soit en calculant le signe de la pente de la fonction linéaire qui s'adapte au mieux au signal sur une fenêtre de temps qui se déplace ou bien en utilisant un algorithme de segmentation qui divise les données en des segments linéaires.
Les signaux du processeur chimique sont décrits par tendance en utilisant les primitives de premier ou de second ordre. La tendance extraite des signaux physiologiques est généralement basée sur seulement trois primitives croissant, décroissant, stable qui correspondent aux mots utilisés par les physiciens pour décrire l'évolution d'un signal.
Nous nous intéressons dans la suite de ce manuscrit aux tendances extraites à partir des primitives du premier et/ou second ordre.

- **Tendances construites à partir de primitives du premier et second ordre** :
  Une tendance d'un signal est décrite à l'aide de 7 primitives données par la figure 2.4, linéaires ou quadratiques [Cheung and Stephanopoulos, 1990]. Elles sont définies à partir de la valeur qualitative $(0, +, -)$ des dérivées premières et secondes du signal. Chaque épisode peut être représentée par une forme triangulaire.
  Dans [Janusz, 1991], deux autres symboles sont ajoutés qui sont le minimum et le maximum, définis par l'état qualitatif {dérivée première nulle et dérivée seconde positive ou négative}. Dans [Colomer et al., 2002], 4 nouvelles primitives

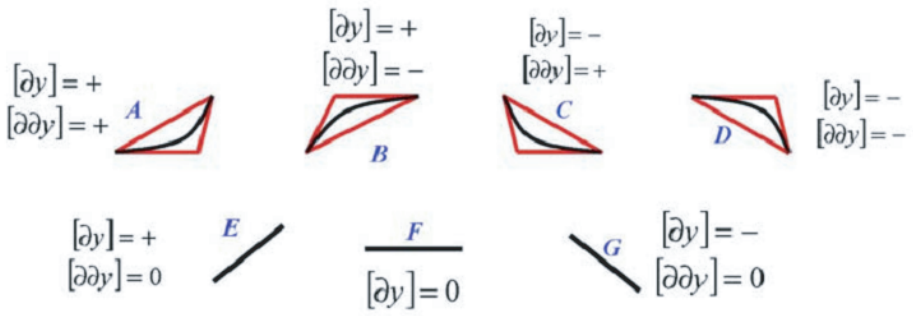

FIGURE 2.4 – Primitives de l'analyse de tendance

représentant explicitement les discontinuités ou ruptures dans le signal. Elles sont caractérisées par le fait que la dérivée seconde prend une valeur importante lors de leur apparition.

L'extraction de la tendance à partir du signal consiste en premier lieu, à identifier les primitives en ligne. Une première solution consiste à calculer des dérivées numériques premières et secondes du signal à chaque période d'échantillonnage. Les deux dérivées sont classées suivant trois classes en fonction de la valeur calculée : positive ou négative, selon le signe, ou bien nulle. Le problème d'amplification du bruit s'impose suite au calcul de la dérivée, et un filtrage passe-bas à priori se trouve nécessaire. Une approche alternative à la dérivation est possible, qui transforme le problème d'identification des primitives en un problème de reconnaissance de forme [Rengaswamy et al., 2001] ce qui nécessite une base d'apprentissage assez riche.

- **Tendances construites à partir de primitives du premier ordre :**
  Dans le cas d'un signal fortement bruité, il est difficile de représenter la tendance par des primitives de second ordre, comme le cas des signaux biologiques ou physiologiques. La tendance consiste ainsi à utiliser trois primitives seulement {croissant, décroissant, stable}, pour décrire l'évolution du signal sous forme simplifiée. La première dérivée, peut être estimée par régression linéaire sur une fenêtre temporelle glissante [Makivirta et al., 1998] ou bien à partir d'une segmentation de signal [Hunter and McIntosh, 1999].
  La tendance peut être utilisée comme entrée à un système d'expertise conçu pour le diagnostic de défaut en utilisant le raisonnement par cas (case-based reasoning). Plusieurs applications dans le domaine du diagnostic des systèmes médicales ont été conçues en utilisant des modèles de tendance [Charbonnier and Gentil, 2007]. En fait, un modèle de tendance est une structure variable dans le temps prédéfinie

dans plusieurs variables associées à un diagnostic. Le diagnostic automatique est réalisé en déterminant si les signaux surveillés en ligne font apparaître les modèles de tendance spécifiques à un défaut.

2.3.2.3 Distance VDM

La distance VDM ou Value Difference Metric est une métrique symbolique utilisée par différents algorithmes de KNN.Elle a été utilisé dans le cadre de la sélection des variables en réduisant le nombre d'attributs irrélèvants [Payne and Edwards, 1998]. Elle utilise des techniques probabilistes simples pour pondérer les attributs dans l'espace des observations.
Cette métrique a été proposée pour déterminer la distance entre deux valeurs symboliques [Stanfill and Waltz, 1986]. La distance est déterminée en comparant la probabilité de distribution conditionnelle par classe, entre les deux valeurs i_a et j_a, prises par le vecteur attribut a.

$$vdm(i,j) = \sum_{a=0}^{A} \delta(i_a, j_a).w(i_a) \tag{2.5}$$

$$\delta(i_a, j_a) = \sum_{c \in C} |P(c \mid i_a) - P(c \mid j_a)|^2 \tag{2.6}$$

$$w(i_a) = [\sum_{c \in C} P(c \mid i_a)^2]^{\frac{1}{2}} \tag{2.7}$$

Avec C est l'ensemble des classes, A étant le nombre total des attributs et $P(c \mid i_a)$ est la probabilité conditionnelle par classe de la variable i_a.
En d'autre termes : c'est la probabilité d'apparition de la valeur i_a dans l'ensemble des données pour l'attribut a dans les instances de la classe c. Cette probabilité est déterminée directement à travers l'ensemble d'apprentissage en comptant le nombre d'instances contenant la valeur i_a pour l'attribut a, puis en déterminant la proportion que contient aussi la classe c :

$$P(c \mid i_a) = \frac{|instances\ contenant\ i_a \wedge classe = c|}{|instances\ contenant\ i_a|} \tag{2.8}$$

Ce processus peut être illustré à travers un exemple pris de [Payne and Edwards, 1998] : considérant la distribution discrète par classe de trois différents valeurs symboliques X, Y, Z, données par la figure 2.5.

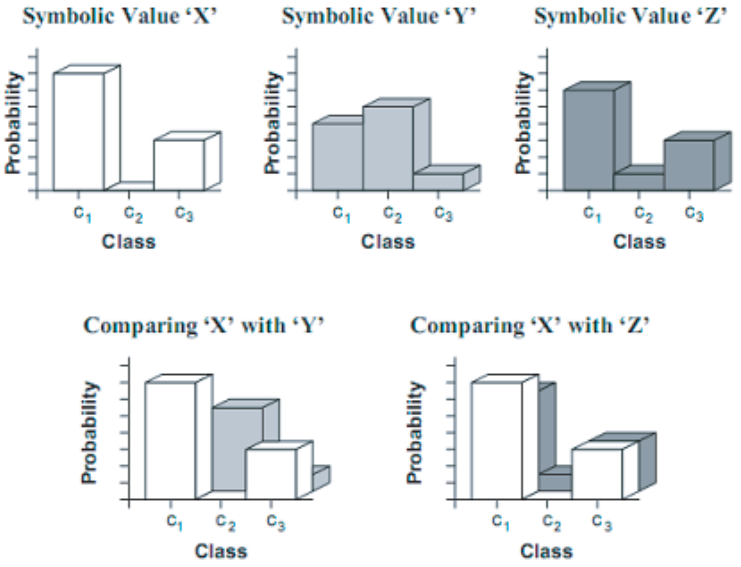

FIGURE 2.5 – Comparaison des valeurs symboliques avec VDM extrait de [Payne and Edwards, 1998]

Chaque distribution consiste en trois distributions conditionnelles par classe, représentées par des barres verticales. La figure représente la comparaison entre chaque paires des trois valeurs symboliques. Pour chaque classe, la différence $\delta(i_a, j_a)$ dans la probabilité conditionnelle est déterminée. Ces différences sont par la suite combinés dans $vdm(i, j)$ et résulte en la mesure de la distance entre deux valeurs symboliques de l'attribut a. Ainsi, pour calculer la distance entre deux symboles 'X' et 'Y', la différence dans la probabilité conditionnelle par classe est déterminée pour chaque classe. Dans cet exemple, les différences sont : $0.3, -0.5, 0.2$ pour les classes c_1, c_2 et c_3 respectivement. Enfin, la distance finale entre ces deux symboles est la somme de ces différences élevées au carré : $\delta(X, Y) = 0.3^2 + (-0.5)^2 + 0.2^2 = 0.38$.
Le poids $w(i_a)$ donne une idée sur la contribution d'un attribut dans la discrimination entre les différentes classes. Il peut varier entre un minimum qui dépend du nombre de classes et la valeur 1 qui représente un discriminateur idéal (une valeur d'attribut qui apparait dans une seule classe). Le minimum représente une distribution uniforme par classe où une valeur d'attribut apparait dans les instances de toutes les classes avec une probabilité égale et peut être calculé par : $w(u) = |C|^{-0.5}$. Le poids peut être utilisé pour contrôler l'effet de l'attribut sur la distance [Payne and Edwards, 1998].
Plusieurs extensions ont été apportées à la distance original VDM afin de manipuler les attributs continus [Wilson and Martinez, 1996] et pour manipuler les données

hétérogènes [Rodriguez et al., 2008].

2.3.2.4 Distance proposée : similarité de comportement

La notion de distance ou mesure de similarité entre deux observations est l'information la plus importante lorsqu'on traite les algorithmes de clustering. Et ce pour identifier les groupes dans l'espace des observations. Usuellement, c'est la distance Euclidienne qui est fréquemment utilisée dans les algorithmes du Clustering comme le KNN et la K-moyennes. Elle calcule la moyenne de la différence entre deux vecteurs sans utiliser aucune information sur la tendance, ce qui peut induire en erreur les résultats de classification. En effet, deux vecteurs peuvent avoir la même moyenne mais des tendances totalement différentes.
A l'encontre, notre méthode focalise sur la définition d'une mesure qui caractérise la similarité dans le comportement de deux vecteurs attributs donnés. Ainsi, pour caractériser un vecteur, nous avons besoin d'appliquer une analyse de tendance afin d'extraire sa signature et la comparer avec les autres signatures et identifier la similitude entre eux. Nous proposons içi d'analyser la tendance de chaque vecteur attribut le long de la base d'observations en se basant sur l'étude de la monotonie.
En effet, l'analyse de tendance, déjà présentée dans la section 2.3.2.2, est une représentation condensée de la dynamique d'un signal [Charbonnier, 2006], reposant sur une bibliothèque de formes utilisées pour la description qualitative ou semi-qualitative du signal.
Ainsi, pour déterminer la tendance d'un vecteur attribut v, nous calculons sa dérivé première. Nous calculons la différence entre deux valeurs v_i et v_{i-1}, consécutives prises par l'attribut v à observation x_i et x_{i-1} :

$$\frac{dv}{dx} = \frac{v_i - v_{i-1}}{x_i - x_{i-1}} \tag{2.9}$$

Nous déterminons par la suite le signe de la dérivé première $\alpha = sign(\frac{dv}{dx})$, pour caractériser la monotonie, tel que :

$$Si \begin{cases} \frac{dv}{dx} < 0 \\ \frac{dv}{dx} > 0 \\ \frac{dv}{dx} = 0 \end{cases} \text{ alors } \alpha = \begin{cases} -1 \text{ (décroissant)} \\ 1 \quad \text{(croissant)} \\ 0 \quad \text{(stable)} \end{cases} \tag{2.10}$$

Un vecteur attribut $v \in \mathbb{R}^L$ est finalement représentée par L variables tel que $\alpha \in \{1, 0, -1\}$. Après l'extraction de la tendance de chaque vecteur attribut, nous

pouvons comparer les tendances deux à deux. Nous calculons ainsi le carré de la somme des différences absolues δ_α, donné par l'équation 2.12, entre l'occurrence d'une valeur spécifique de α pour deux vecteurs données comme c'est fait par la distance VDM, introduite à la section 2.3.2.3. Ceci permet de garantir l'indépendance de la distance développée entre deux vecteurs attributs par rapport aux positions de chaque α (expression d'une tendance) grâce à la distribution conditionnelle de chaque tendance extraite.
La distance entre deux attributs est alors exprimée par :

$$d(v_i, v_j) = \sqrt{\delta_1(v_i, v_j) + \delta_{-1}(v_i, v_j) + \delta_0(v_i, v_j)} \tag{2.11}$$

Avec

$$\delta_\alpha(v_i, v_j) = |p(v_i/\alpha) - p(v_j/\alpha)|\,, \alpha \in \{0, 1, -1\} \tag{2.12}$$

$$p(v_i/\alpha) = \frac{\text{occurence de } \alpha}{L}\,. \tag{2.13}$$

$p(v_i/\alpha)$ est déterminé en calculant le nombre de fois où la valeur $\alpha = 0$, $\alpha = 1$ ou $\alpha = -1$ apparaît dans le vecteur attribut v_i comme l'illustre la figure 2.6.

Une autre idée consiste à travailler avec deux variables (α, β) comme nous l'avons proposé dans [Ferchichi et al., 2011]. Ainsi, nous comparons l'occurrence d'une combinaison de ces deux variables avec 9 possibilités au lieu de 3. nous aurons à calculer l'occurrence de chacune des possibilités suivantes pour chaque deux vecteurs à comparer :
$(\alpha, \beta) \in \{(1,0), (1,1), (1,-1), (-1,0), (-1,1), (-1,-1), (0,0), (0,1), (0,-1)\}$.
Une matrice de similarité M peut ainsi être déterminée, en mesurant la similarité entre attributs deux à deux : $M = d(v_i, v_j)$. Deux vecteurs identiques auront une mesure nulle $d(v_i, v_i) = 0$. D'autant que les vecteurs sont différents, la distance est supérieure à 0.
Deux chemins se présentent à ce stade : comparaison de la signature de chaque attribut classe par classe (version supervisée) ou bien les comparer indépendamment des classes des observations et donc d'une façon non supervisée.

2.3.3 Algorithme de classification des attributs

La procédure d'extraction d'attribut proposée dans ce travail, est basée sur un algorithme de classification non supervisée ou clustering, donné par l'algorithme 2. Elle

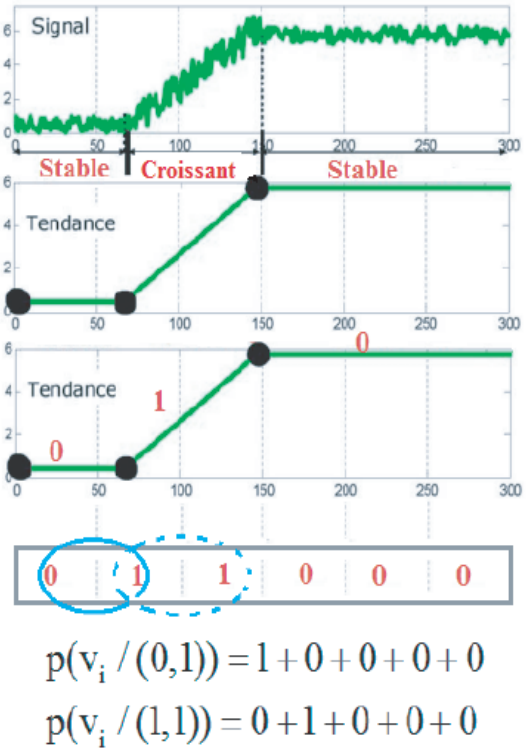

FIGURE 2.6 – Analyse de tendance et détermination de la mesure de similarité

utilise la mesure de similarité à base d'analyse de tendance déjà présentée dans la section 2.3.2.2.

La première étape dans notre algorithme d'extraction d'attributs consiste à analyser la tendance de chaque vecteur attribut v_i de la matrice des attributs V. Une matrice de même dimension que V contiendra la tendance de chaque attribut codée en $0, 1, et -1$. La probabilité conditionnelle $p(v_i/(\alpha, \beta))$ de trouver la combinaison $(\alpha, \beta))$ dans v_i est alors déterminée. La distance deux à deux entre les attributs sera ainsi calculée et la matrice de similarité M est alors construite. Les valeurs de cette matrice sera ordonnée dans l'ordre décroissant.
Une approche de clustering C-means est appliquée pour trouver les partitions dans l'espace d'attributs en se basant sur la matrice de similarité M. Un nombre initial de clusters est fixée $d = \frac{N}{epsilon}$. Chaque cluster C_k est initialisé par l'attribut classé en premier dans la matrice M et il est considéré en tant que premier centre du cluster. Par la suite, les ϵ premiers attributs classés dans M par rapport à ce centre sont sélectionnés. Le nouveau centre est alors mis à jour en calculant la moyenne des

Algorithm 2 Algorithme de FEMC de base

Require: $X \in \mathbb{R}^{LD}$: matrice contenant les observations,
ε : seuil,
C : nombre d'attributs initiale

construire la matrice $V \in \mathbb{R}^{DL}$ qui contient les attributs originaux. Chaque ligne représente un vecteur attribut
extraire la tendance de chaque vecteur attribut v_i de la matrice V
calculer le signe de la tendance de chaque attribut : déterminer $sign(dv_i/dx)$
calculer la probabilité conditionnelle $p(v_i/(\alpha, \beta))$ pour chaque possibilité $(\alpha, \beta) \in \{(1,0),(1,1),(1,-1),(-1,0),(-1,1),(-1,-1),(0,0),(0,1),(0,-1)\}$
calculer la matrice de distance M, avec $M(i,j) = d(v_i, v_j)$ tel que $i, j \in 1, ..., D$.

ordonner la matrice M dans l'ordre croissant
initialiser le premier cluster $C_{k=1}$, $k = 1..d$, avec le premier attribut v_1 dans la matrice V et mettre l'indice $index = 1$
fixer le nombre final des cluster d, $d = Ddiv\epsilon$
construire les clusters :
sélectionner les ε attributs classés en premiers par rapport l'attribut v_{index} dans la ligne numéro $index$ dans la matrice M.
mettre les attributs sélectionné dans le cluster C_k.
calculer le centroid g_k correspondant au cluster C_k.
mettre à jour le nombre d'attributs $C \leftarrow C - \varepsilon$
mettre à jour l'indice $index \leftarrow index + \varepsilon + 1$
déterminer les attributs communs entres les cluster $\forall\{k_1, k_2\} \in \{1..d\}, C_{k_1} \bigcap C_{k_2}$: intersection entre clusters
attribuer chacun des attributs commun au cluster le plus proche selon la distance euclidienne
mettre à jour les centres g_k des clusters
return les nouveaux attributs sont l'ensemble des centres g_k
return la transformation de la matrice de donnée $T(X) \in \mathbb{R}^{dL}$

attributs dans chaque cluster. Ce centre peut être décrit par la forme suivante :

$$g_{j\in\{1...d\}} = W_j V_j \,. \tag{2.14}$$

Avec, $W_j = \mathbf{1} * \frac{1}{n_j}$ est le vecteur de transformation appliqué à l'ensemble de n_j attributs $V_j \in C_j$.

Cette procédure permet en fait, un chevauchement entre les clusters : un même attribut peut appartenir à plusieurs clusters. Pour remédier à ce problème, nous appliquons une procédure d'intersection entre chaque paire de clusters obtenus. Les attributs ainsi commun seront assignés au centre le plus proche selon la distance

Euclidienne :

$$\forall v_j \in \{C_{k1} \cap C_{k2}\}, v_i \in C_h avech = argmin \|g_{j\in\{k1,k2\}} - v_i\|^2 . \quad (2.15)$$

Avec h est soit l'index $k1$ ou bien l'index $k2$.
Les centres des clusters seront par la suite re-calculés et mis à jour en tenant compte des nouvelles appartenances. Le processus est arrêté quand les d clusters sont formés.

2.4 Information Mutuelle pour l'extraction d'attributs

Outre le calcul de la mesure de similarité et la formulation de la distance, un autre point parait très important dans les approches de classification non supervisée. C'est celui de la détermination du centre du cluster. Nous avons pénétré alors sur cette problématique et dans ce contexte nous avons exploré l'idée de l'Information Mutuelle (IM) comme critère de recherche des centres. Dans cette section nous aborderons en premier lieu le concept de l'Information Mutuelle et sa formulation issue de la théorie de l'information. La formulation du critère de recherche des centres de chaque clusters à base de la mesure IM sera par la suite explicitée.

2.4.1 Information Mutuelle : Préliminaires

La théorie de l'information fournit une mesure quantitative de la notion d'information apportée par une observation. Cette notion fut introduite par Shannon [Shannon, 1948]. Nous présentons par la suite quelques préliminaires sur l'information mutuelle ainsi que les approches pour son estimation.
Supposons que nous avons deux distributions de variables X et Y. L'information mutuelle peut alors se définir comme étant la mesure de réduction d'incertitude d'une variable au vu de la connaissance de la seconde variable. Autrement dit, l'information mutuelle de deux variables aléatoires est une quantité mesurant la dépendance statistique de ces variables. L'information mutuelle de deux variables discrètes avec $P(X,Y)$ étant leur densité de probabilité jointe, est donnée par l'équation suivante :

$$I(X,Y) = \sum_X \sum_Y P(X,Y) \ln \frac{P(X,Y)}{P(X)P(Y)} \quad (2.16)$$

Dans le cas continu, l'information mutuelle est donnée par l'équation suivante :

$$I(X,Y) = \int_X \int_Y P(x,y) \log \frac{P(x,y)}{P(x)P(y)} dxdy \tag{2.17}$$

Si l'information mutuelle entre deux variables est grande (petite), cela signifie que ces deux variables sont liées (non liées).
Parmi les propriétés de l'information mutuelle, on trouve :

1. $I(X,Y) = 0$ si et seulement si X et Y sont des variables aléatoires indépendantes,
2. L'information mutuelle est positive ou nulle : $I(X,Y) \geq 0$,
3. L'information mutuelle est symétrique : $I(X,Y) = I(Y,X)$.

Dans le cas continu, il est assez difficile de trouver les fonctions de densité de probabilité *(p(x), p(y), p(x,y))* et d'effectuer ces intégrations. Pour cette raison, l'espace d'attributs en entrée est généralement divisé en plusieurs partitions discrètes, et le calcul de l'information mutuelle se fait ainsi en utilisant la définition du cas discret. L'erreur inhérente qui existe dans cette procédure est à prendre en considération dans le cas de variables continues.

2.4.2 Estimation non paramétrique de l'Information Mutuelle

Plusieurs approches d'estimation non paramétrique de l'IM existe dans la littérature dont l'approche par Histogramme, estimation par noyau et estimation de densité par fenêtre de Parzen [Schaffernicht et al., 2010].
La première approche est une approche standard qui partitionne les axes en différentes barres (ou classes) de largeur w_i et par la suite détermine le nombre n_i des observations dans la classe i. La probabilité normalisée est obtenue par : $p_i = \frac{n_i}{Nw_i}$, avec $\int p(x)dx = 1$. La fonction de probabilité est ainsi constante le long de la largeur de chaque barre. Une approche dérivée consiste à utiliser un ensemble d'histogrammes avec différentes largeurs de barres. La IM finale est calculée à partir de la moyenne des valeurs estimées de IM pour chaque barre. Une autre dérive existe appelée partitionnement adaptative du plan XY [Cellucci et al., 2005]. La largeur de barre de l'histogramme est variable en en fonction des données. Le plan est partitionné en divisant chaque axe en plusieurs segments équiprobable.
La deuxième approche estime la fonction de densité de probabilité de X par la su-

perposition de fonctions noyaux $k(u)$, centrées sur les points de données. Elle est donnée par l'expression suivante $p(x) = \frac{1}{Nh^d} \sum k(\frac{x-x_n}{h})$.
La troisième approche estime directement la fonction du taux de densité $w(x, y) = \frac{p(x,y)}{p(x)p(y)}$ par une fonction approximé $\tilde{w}_\alpha(x, y) = \alpha^T \varphi(x, y)$. Avec α est un vecteur paramètre à apprendre à partir des données et φ est un vecteur de fonctions de base, tel que $\varphi(x, y) \geq 0$. α est déterminé en minimisant l'erreur quadratique.
L'approche par KNN utilise un nombre fixe k de voisins pour estimer la IM. Pour chaque point de la base de données, le volume minimal qui encapsule k points est déterminé. En comptant le nombre de points dans le volume, la IM peut ainsi être estimée.

2.4.2.1 Estimation de densité par fenêtre de Parzen

Les travaux de Principe en 2000 [Principe et al., 2000] portent sur l'utilisation de l'entropie de Renyi combinée avec la méthode de la fenêtre de Parzen pour l'estimation de la densité. Dans Torkkola [Torkkola, 2003], l'entropie de Renyi n'est pas utilisé directement. Il a montré que les mesures qui incluent des formes quadratiques de densité, peuvent être estimées directement à partir des données d'une façon non paramétrique.

Entropie de Renyi

C'est une famille de mesures paramétriques qui inclue l'entropie de Shannon comme un cas particulier. Dans le cas d'une variable discrète C et une variable continue Y, l'entropie de Renyi d'ordre α est définie par :

$$H_{R_\alpha}(C) = \frac{1}{1-\alpha} \log \sum_c p(c)^\alpha; \tag{2.18}$$

$$H_{R_\alpha}(Y) = \frac{1}{1-\alpha} \int_y p(y)^\alpha dy. \tag{2.19}$$

Où $\alpha > 0, \alpha \ncong 1$, $lim_{\alpha \to 1} H_{R_\alpha} = H$. Dans notre cas, nous considérons le cas de la mesure quadratique $\alpha = 2$.

Fenêtre de Parzen

La mesure d'entropie des variables continues est basée sur la fonction de densité de probabilité des variables. Une des méthodes non paramétriques pour estimer la densité est la méthode de fenêtre de Parzen [Xu et al., 2008]. La méthode implique

le placement d'une fonction noyau (kernel) sur le top de chaque échantillon et l'évaluation de la densité comme étant la somme des fonctions noyaux (kernels). Ainsi, il s'avère que la mesure quadratique de Renyi combinée avec l'estimation de densité par la méthode de Parzen en utilisant les noyaux gaussiens assure une économie signifiante en terme de computation [Torkkola, 2003].
Une fonction noyau dans un espace de d-dimension est donnée par :

$$G(y, \Sigma) = \frac{1}{(2\Pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} exp(\frac{-1}{2} y^T \Sigma^{-1} y) \tag{2.20}$$

Avec y est un échantillon aléatoire et Σ est la matrice de covariance. Pour deux fonctions kernels, on a :

$$\int_y G(y - a_i, \Sigma_1) G(y - a_j, \Sigma_2) dy = G(a_i - a_j, \Sigma_1 + \Sigma_2) \tag{2.21}$$

La convolution de deux Gaussiennes centrées à a_i et a_j est une Gaussienne centrée à $a_i - a_j$ avec une covariance égale à la somme des covariances originales. Cette propriété facilite l'évaluation de la mesure quadratique de Renyi pour l'estimation de la densité, qui n'est qu'une fonction du carré de la fonction de densité.
Supposons maintenant que la densité de Y est estimée comme étant une somme de Gaussiennes sphériques centrée chacune à un échantillon y_i (estimateur de Parzen) :

$$p(y) = \frac{1}{N} \Sigma_{i=1}^{N} G(y - y_i, \sigma I) \tag{2.22}$$

Le symbole I indique une matrice unité. L'entropie quadratique de Renyi est donc donnée par :

$$\begin{aligned} H_{R_2}(Y) &= -\log \int_y p(y)^2 dy \\ &= -\log \frac{1}{N^2} \int_y (\sum_{k=1}^{N} \sum_{j=1}^{N} G(y - y_k, \sigma I) G(y - y_j, \sigma I)) dy \\ &= -\log \frac{1}{N^2} \sum_{k=1}^{N} \sum_{j=1}^{N} G(y_k - y_j, 2\sigma I) \end{aligned} \tag{2.23}$$

Ainsi, l'entropie quadratique de Renyi peut être estimée comme étant une somme d'interactions locales, comme définie par le kernel, le long de toutes les paires d'exemples. En pratique, seulement la moitié sera évaluée grâce à la symétrie.

2.4.2.2 Information Mutuelle Quadratique

L'entropie d'une fonction multivariée de densité peut être calculée d'une façon non paramétrique à partir de l'ensemble des données. Elle est perçue comme étant une interaction entre les paires des échantillons. Ceci peut être étendu au cas de l'information mutuelle entre les variables. Ainsi pour appliquer l'équation 2.21, l'information mutuelle doit être exprimée comme une fonction de la densité des variables au carré, ou bien sous une forme de produit de convolution de deux fonctions kernels. [Kapur, 1994] argumente que si le but n'étant pas de calculer la valeur absolue de la divergence, mais plutôt de trouver une distribution qui maximise/minimise la divergence, les axiomes utilisés pour dériver la mesure peuvent être relaxés et le résultat de l'optimisation sera toujours la même distribution. [Kapur, 1994] présente un large nombre de tels mesures pour deux distributions discrètes P et Q, parmi lesquels il y a la mesure suivante :

$$D(P,Q) = \frac{1}{\alpha(\alpha-1)} \sum_{i=1}^{n} [p_i^{\alpha} - \alpha p_i q_i^{\alpha-1} + (\alpha-1) q_i^{\alpha}], \alpha \neq 0, \alpha \neq 1. \tag{2.24}$$

Choisir $\alpha = 2$, permet d'étendre la mesure de l'équation 2.24 au cas de densité continue donne simplement :

$$D(f,g) = \int_x (f(x) - g(x))^2 dx, \tag{2.25}$$

Il est clair que la mesure est toujours positive et qu'elle s'annule quand $f(x) = g(x)$ pour toutes les observations. Vu que l'information mutuelle donnée par l'équation 2.16 peut être exprimée comme étant une divergence entre la densité jointe et le produit des marginaux, leur insertion dans l'équation 2.25 conduit à une formulation de mesure d'information mutuelle quadratique entre deux variables continues Y_1 et Y_2 dénotée I_T :

$$I_T = \int \int (p(y_1, y_2) - p(y_1)p(y_2))^2 dy_1 dy_2 \tag{2.26}$$

2.5 Sélection et extraction par Information Mutuelle

L'information mutuelle étant un concept connu de la théorie d'information, utilisée pour capter toute structure de dépendance entre une paire de variables aléatoires. A l'encontre des autres approches comme le coefficient de corrélation, IM n'est pas

limitée au cas des dépendances linéaires mais elle les dépasse au cas des dépendances non linéaires.

L'Information Mutuelle a été exploitée dans la sélection des variables est une approche très intuitive et largement abordée. Elle a été utilisée comme un critère pour filtrer des variables [Battiti, 1994], [Yang and Moody, 1999], [Kwak and Choi, 2002]. Le défit d'utiliser l'IM pour la sélection des variables, de point de vue pratique, réside dans l'estimation de cette mesure à partir des données disponibles. Ainsi, plusieurs méthodes paramétriques et non paramétriques ont été proposées pour l'estimation de IM ont été évoqué dans la section 2.4.2.

Évaluer l'IM entre deux variables scalaires à base des histogrammes a été exploité dans la sélection des variables [Battiti, 1994], [Yang and Moody, 1999], [Kwak and Choi, 2002]. En fait, l'estimation de IM basée sur les histogrammes ne fonctionne que pour deux ou trois variables au maximum. Elle échoue dans les grandes dimensions à cause de la parcimonie de l'ensemble des données. L'approche KNN pour l'estimation de IM, ne peut pas être considérée dans ce travail car l'estimation de IM est à optimiser avec une structure basée sur le gradient [FisherIII et al., 1998].

L'extraction à base de l'information mutuelle a été appliqué dans le cadre de l'apprentissage supervisé. En fait, le but était de chercher les directions qui permettront de maximiser la mesure IM entre l'ensemble des variables et les labels des classes [Torkkola, 2003], [Kwak, 2007], [HildII et al., 2006], [FisherIII et al., 1998]. Dans [Torkkola, 2003], l'entropie de Reny combinée avec un estimateur non paramétrique de Parzen a été utilisée. Ceci a permis de réduire la complexité de $O(N_T^2)$ à $O(N_T)$ [HildII et al., 2006]. Dans [HildII et al., 2006], la matrice d'extraction est contrainte à être une pure matrice de rotation. Dans [FisherIII et al., 1998], la somme des IMs estimées entre chaque variables et les labels des classes à part au lieu d'estimer l'information mutuelle totale entre les variables et les labels des classes. La somme des MIs constitue ainsi la IM totale. Dans [Kwak and Choi, 2002], l'auteur utilise l'estimateur de Parzen pour calculer directement la IM.

2.6 Approche d'extraction d'attributs basée sur IM

2.6.1 Formulation basée sur IM

L'objective de cette section est d'appliquer une transformation plus adaptée à chaque groupe d'attributs identifié. En général, après la construction de chaque cluster/groupe d'attributs, nous appliquons une transformation linéaire sur les données pour dériver un bary-centre g_j qui sera l'attribut représentatif du cluster C_j en question :

$$g_j = W_j V_j, \quad avec\, W_j = \mathbf{1}\frac{1}{n_j} \tag{2.27}$$

Avec n_j est le nombre d'attributs dans un cluster C_j, et V_j est la matrice contenant les attributs appartenant à ce cluster.
Dans la procédure d'extraction que nous proposons dans ce travail, notre but est d'utiliser les barycentres des clusters obtenus comme étant les nouveaux attributs ou descripteurs de notre base de données. Chaque barycentre sera le représentatif de sa classe d'attribut et incorpora les informations nécessaires pour l'identification de son groupe. Ainsi, il devient intuitive de chercher une transformation plus appropriée W^* qui répond au besoin de trouver les centroids les plus représentatifs.
Nous avons besoin alors d'un critère qui capte dans un même cluster d'attributs les relations linéaires/non linéaires entre un centre candidat et le reste des attributs. L'information mutuelle introduite dans la section précédente représente une mesure fiable qui préserve les dépendances linéaires et non linéaires entre les variables.
Le critère à adopter pour le choix des centroides sera la maximisation de la mesure d'information mutuelle entre chaque attribut dans un cluster et son centroide correspondant.
Nous cherchons ainsi à définir une transformation f que nous allons appliquer pour chaque attribut $v_i \in C_j$ de l'ensemble d'attributs V_j de C_j pour obtenir le centre $g_{ij} = f(w, v_i)$ et qui maximise $I(g_j, V_j)$.
Le principe est ainsi décrit par la figure 2.7. Pour chaque cluster en entrée représenté par la matrice V, on donne un barycentre initial g_0. L'IM $I(g, V)$ sera calculée entre le centre initial et les attributs dans le clusters. Une procédure de descente de gradient se chargera dés lors de chercher les poids w qui maximisent la MI.

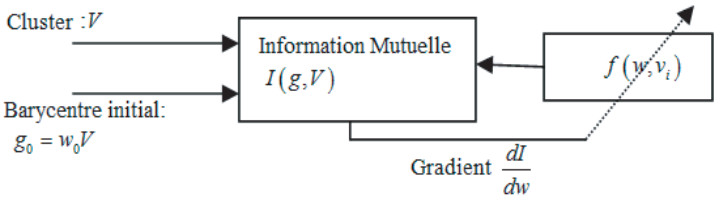

FIGURE 2.7 – Procédure de maximisation de IM entre les attributs et leur centre

En utilisant l'équation 2.26, on obtient :

$$I(g_j, V_j) = \int\int (p(g_j, v) - p(g_j)p(v))^2) dg_j dv \tag{2.28}$$

On a à développer $p(g_j, v)$ pour calculer I définie dans l'équation 2.28. Vu que le centre recherché $g_{ij} = f(w, v_i)$ appartient au cluster C_j en question, le cluster final C_j^{final} d'attributs sera constitué de l'ensemble original d'attributs V_j et en plus le centre g_j. Il est donné par : $C_j^{final} = V_j \bigcup g_j$.
La densité de probabilité jointe $p(g_j, v)$ est exprimée par :

$$p(g_j, v) = p(g_j)p(g_j/v) = p(g_j)(p(v) - p(g_j)) \tag{2.29}$$

en insérant l'équation 2.29 dans l'équation 2.28, L'information mutuelle sera exprimée par :

$$I(g_j, V_j) = \int\int p(g_j)^4 dg_j dv \tag{2.30}$$

On applique l'estimation par la méthode de la fenêtre de Parzen pour déterminer $p(g_j)$ sur l'ensemble d'attribut finalement obtenu C_j^{final} comptant $n_j + 1$ attributs. La densité de probabilité $p(g_j)$ est donnée par la méthode de la fenêtre de Parzen :

$$p(g_j) = \frac{1}{n+1} \sum_{i=1}^{n+1} \phi(g_j - v_j, \sigma I) \tag{2.31}$$

En utilisant l'équation de la fenêtre de parzen.
On note $I(g_j, V_j) = I$ et $g_j = y$ et $v = x$, pour simplifier l'écriture, et on sait que $\int Z_c N_x(\mu_c, \Sigma_c) dx = Z_c$.

On aura alors la IM donnée par :

$$
\begin{aligned}
I &= \int\int \frac{1}{(n+1)^4}(\sum_{i=1}^{n+1}\phi(y-y_i,\sigma I))^4 dydx \\
&= \frac{1}{(n+1)^4}\int\int \sum_{i=1}^{n+1}\phi(y-y_i,\sigma I)\sum_{j=1}^{n+1}\phi(y-y_j,\sigma I)\sum_{k=1}^{n+1}\phi(y-y_k,\sigma I)\sum_{l=1}^{n+1}\phi(y-y_l,\sigma I)dydx \\
&= \frac{1}{(n+1)^4}\int\int \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}\sum_{l=1}^{n+1}\phi(y-y_i,\sigma I)\phi(y-y_j,\sigma I)\phi(y-y_k,\sigma I)\phi(y-y_l,\sigma I)dydx \\
&= \frac{1}{(n+1)^4}\int\int \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}\sum_{l=1}^{n+1}\prod_{s=1}^{4}\sigma_y(\mu_s,\Sigma_s)dydx \\
&= \frac{1}{(n+1)^4}\int \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}\sum_{l=1}^{n+1}\hat{z}dx
\end{aligned}
\tag{2.32}
$$

l'expression de l'information mutuelle devient alors :

$$
\begin{aligned}
I(g,V) &= \int_g\int_v \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}\sum_{l=1}^{n+1}\prod_{s=1}^{4}4\Phi_g(\mu_s,\Sigma_s)dgdv \\
&= \int_g\int_v \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}\sum_{l=1}^{n+1} z\Phi_g(\mu,\Sigma)dgdv \\
&= \int_v \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}\sum_{l=1}^{n+1} z dv\,.
\end{aligned}
\tag{2.33}
$$

Avec

$$
\begin{aligned}
z = \frac{|2\pi\Sigma_d|^{\frac{1}{2}}}{\prod_{s=1}^{4}|2\pi\Sigma_d|^{\frac{1}{2}}}\prod_{a<b} exp(-\frac{1}{2}(\mu_a-\mu_b)^T B_{ab}(\mu_a-\mu_b)), \\
\Sigma_d = (\sum_{s=1} 4\Sigma_s^{-1})^{-1}, \\
\Sigma_s = \sigma^2 I, \\
B_{ab} = \Sigma_s^{-1}\Sigma_d\Sigma_s^{-1}
\end{aligned}
$$

Dans la figure 2.7, nous avons maximisé MI=$I(g,V)$ pour obtenir une transformation optimale f^* et la matrice des poids optimale W^* et déterminer le centre g final. Nous devons utiliser un algorithme de descente de gradient pour trouver la solution. Selon la figure 2.7, la descente de gradient a besoin de calculer le gradient $\frac{dI}{dw}$, ce qui induit

la différentiation de $I(g, V)$. Rappelons que $g = WV$, on obtient alors la forme suivante :

$$\frac{dI}{dw} = \frac{dI}{dg}\frac{dg}{dw} = \frac{dI}{dg}V. \tag{2.34}$$

2.6.1.1 Optimisation par descente de gradient

L'optimisation par descente de gradient est une des méthodes les plus simple d'optimisation dans un espace continu ou discret. Elle consiste à déterminer, dans un espace des hypothèses H la direction qui minimise une fonction f. Un vecteur w^t est considéré le vecteur de l'espace $\Re$ caractérisant une hypothèse à l'étape t. La valeur de l'hypothèse courante peut alors être notée $f(w_t)$. Pour approcher l'hypothèse cherchée, la descente de gradient consiste à ajuster le vecteur w. Nous obtenons alors, une séquence de vecteurs $w^0, w^1, ..., w^t$ obéissant à l'équation de récurrence :

$$w^{t+1} = w^t - \alpha \frac{\partial f(w)}{\partial w}|_{w^t} \tag{2.35}$$

Où α est un réel positif que l'on appelle *le pas d'apprentissage* et $\frac{\partial f(w)}{\partial w}|_{w^t}$ est la dérivée partielle du critère $f(w)$ par rapport au paramètre w à l'étape w^t. Pour choisir la prochaine valeur w^{t+1}, on retranche à w^t une quantité proportionnelle à la valeur de la dérivée du risque empirique par rapport à w au point w^t.
Dans l'espace $\Re$, l'illustration de la méthode et de ses faiblesses est donnée par la figure 2.8, on note que la fonction à optimiser est f et l'unique valeur cherchée est notée u^* [Cornuéjols and Miclet, 2009]. On y note trois cas caractéristiques :

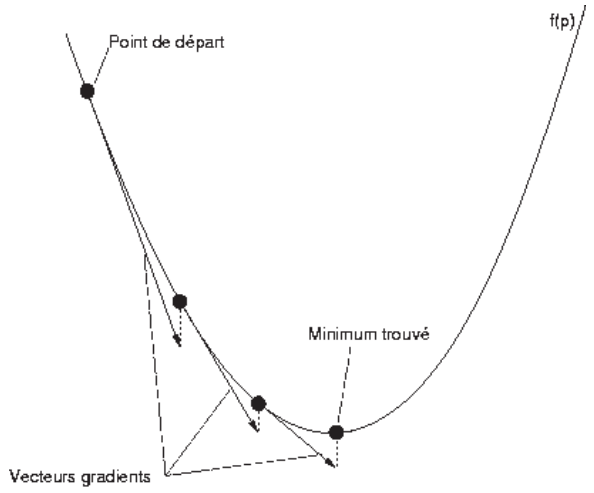

FIGURE 2.8 – Méthode de descente de gradient extrait

• Si u^t se trouve dans la partie droite du graphe, le gradient $\frac{\partial f(u)}{\partial u}|_{u^t}$ est positif et

grand.

Par conséquence, nous prenons :

$$u^{t+1} = u^t - \alpha \frac{\partial f(u)}{\partial u}|_{u^t} \tag{2.36}$$

avec α réel positif, on obtiendra u^{t+1} au point indiqué. À cet endroit, le gradient est plus petit, quoique positif. Le point suivant u^{t+2}, calculé avec la même valeur de α, sera donc la plus proche de u^{t+1} que u^{t+1} l'était de u^t. On assiste de la sorte à la convergence de la suite vers la valeur minimale u^* cherchée.

- Construisons maintenant une autre suite à partir du point v^t, avec une valeur supérieure pour α. Cette suite converge aussi, mais il y a deux différences avec le cas précédent :
  - La suite n'est plus strictement croissante ou décroissante, mais converge par valeurs alternativement supérieures et inférieures à la valeur finale,
  - La valeur atteinte n'est pas le minimum u^*, mais une u_{locale}. Cette "erreur" de convergence vers un minimum local se rencontre très souvent en pratique.
- à partir du point noté w^t, en prenant α encore plus grand la suite va diverger.

2.6.1.2 Formulation de la minimisation de MI par descente de gradient :

Pour minimiser la MI, nous utilisons la procédure de descente de gradient déjà décrite à la section précédente. Il s'agit de trouver le poids w^* optimal. D'après l'équation 2.33 la quantité I est exprimée par :

$$I(g, V) = \int_v \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} \sum_{k=1}^{n+1} \sum_{l=1}^{n+1} z dv \,. \tag{2.37}$$

La procédure de descente de gradient procède par dériver I par rapport à w ce qui méne à calculer $\frac{dI}{dg}$ d'après l'équation 2.37. Et en notant que g n'est autre que l'élément $(n+1)$ de l'ensemble V, on a donc : $\frac{dI}{dg} = \frac{dI}{dv_{n+1}}$.

On calcule alors la quantité :

$$\frac{dI}{dv_{n+1}} = \frac{dQ_1}{dv_{n+1}} + \frac{dQ_2}{dv_{n+1}} + \frac{dQ_3}{dv_{n+1}} + \frac{dQ_4}{dv_{n+1}} + \frac{dQ_5}{dv_{n+1}} \tag{2.38}$$

Les quantités Q_1, Q_2, Q_3, Q_4 et Q_5 sont données par :

$$Q_1 = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} q(l,k,j,i) \tag{2.39}$$

$$Q_2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} q(l=n+1,k,j,i) \tag{2.40}$$

$$Q_3 = \sum_{i=1}^{n}\sum_{j=1}^{n} q(l=n+1,k=n+1,j,i) \tag{2.41}$$

$$Q_4 = \sum_{i=1}^{n} q(l=n+1,k=n+1,j=n+1,i) \tag{2.42}$$

$$Q_5 = q(l=n+1,k=n+1,j=n+1,i=n+1). \tag{2.43}$$

Avec

$$q(l,k,j,i) = exp((v_i-v_j)^T B(v_i-v_j)) + (v_i-v_k)^T B(v_i-v_k) + (v_i-v_l)^T B(v_i-v_l) + (v_j-v_k)^T B(v_j-v_k) + (v_j-v_l)^T B(v_j-v_l)) + (v_k-v_l)^T B(v_k-v_l) \quad . \tag{2.44}$$

Ainsi, on retrouve :

$$\frac{dQ_1}{dv_{n+1}} = 0 \tag{2.45}$$

$$\begin{aligned}\frac{dQ_2}{dv_{n+1}} &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\frac{dq(l,k,j,i)}{dv_{n+1}} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{dq(l=n+1,k,j,i)}{dv_{n+1}} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(\frac{v_{n+1}-v_i}{\det(B)} + \frac{v_{n+1}-v_j}{\det(B)} + \frac{v_{n+1}-v_k}{\det(B)})exp(e_2).\end{aligned} \tag{2.46}$$

$$\begin{aligned}\frac{dQ_3}{dv_{n+1}} &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\frac{dq(l,k,j,i)}{dv_{n+1}} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{dq(l=n+1,k=n+1,j,i)}{dv_{n+1}} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{2*((v_{n+1}-v_i)+(v_{n+1}-v_j))}{\det(B)})exp(e_3)\end{aligned} \tag{2.47}$$

$$\frac{dQ_4}{dv_{n+1}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\frac{dq(l,k,j,i)}{dv_{n+1}}$$
$$= \sum_{i=1}^{n}\frac{dq(l=n+1,k=n+1,j=n+1,i)}{dv_{n+1}} \tag{2.48}$$
$$= \sum_{i=1}^{n} 3*(v_{n+1}-v_i)exp(e4).$$

$$\frac{dQ_5}{dv_{n+1}} = 0 \tag{2.49}$$

Avec

$$\begin{aligned} e_2 &= (v_i-y_j)^T B(v_i-v_j) + (v_i-v_k)^T B(v_i-v_k) + (v_i-v_{n+1})^T B(v_i-v_{n+1}) + \\ &\quad (v_j-v_k)^T B(v_j-v_k) + (v_j-v_{n+1})^T B(v_j-v_{n+1})) + (v_k-v_{n+1})^T B(v_k-v_{n+1}). \\ e_3 &= (v_i-v_j)^T B(v_i-v_j) + 2*(v_i-v_{n+1})^T B(v_i-v_{n+1}) + 2*(v_j-v_{n+1})^T B(v_j-v_{n+1}). \\ e_4 &= 3*(v_i-v_{n+1})^T B(v_i-v_{n+1}). \end{aligned} \tag{2.50}$$

$\frac{dI}{v_{n+1}}$ est finalement donnée par l'expression suivante :

$$\frac{dI}{dv_{n+1}} = 4*\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{(v_{n+1}-v_i)+(v_{n+1}-v_j)+(v_{n+1}-v_k)}{\det(B)}exp(e2) +$$
$$6*\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{2*((v_{n+1}-v_i)+(v_{n+1}-v_j))}{\det(B)}exp(e3) + 4*\sum_{i=1}^{n}(v_{n+1}-v_i)exp(e4). \tag{2.51}$$

Pour alléger la complexité de l'estimation de la fenêtre de Parzen, l'optimisation par descente de gradient stochastique est implémentée. Ainsi au lieu de calculer le gradient complet comme a été fait dans la formulation précédente, on prend des échantillons aléatoires représentant chaque cluster. Ainsi, le gradient sera calculé sur un sous ensemble de variables dans chaque cluster et non pas sur toutes les variables appartenants au cluster.

2.6.2 Algorithme de classification des attributs avec IM

La procédure d'extraction d'attribut par classification non supervisée mené par l'Information Mutuelle, est constituée de deux modules. Le premier étant le module d'extraction non supervisée par classification d'attributs explicité dans le section 2.3 et donné par l'algorithme 3. Le deuxième module consiste en un bloc de maximisation de IM pour la recherche de nouveaux centres optimaux donné par l'algorithme

4.

Algorithm 3 Algorithme de FEMC avec IM

Require: $X \in \mathbb{R}^{LD}$: matrice contenant les observations,
ε : seuil,
C : nombre d'attributs initiale

construire la matrice $V \in \mathbb{R}^{DL}$ qui contient les attributs originaux. Chaque ligne représente un vecteur attribut
extraire la tendance de chaque vecteur attribut v_i de la matrice V
calculer le signe de la tendance de chaque attribut : déterminer $sign(dv_i/dx)$
calculer la probabilité conditionnelle $p(v_i/(\alpha, \beta))$ pour chaque possibilité $(\alpha, \beta) \in \{(1,0),(1,1),(1,-1),(-1,0),(-1,1),(-1,-1),(0,0),(0,1),(0,-1)\}$
calculer la matrice de distance M, avec $M(i,j) = d(v_i, v_j)$ tel que $i, j \in 1, ..., D$.

ordonner la matrice M dans l'ordre croissant
initialiser le premier cluster $C_{k=1}$, $k = 1..d$, avec le premier attribut v_1 dans la matrice V et mettre l'indice $index = 1$
fixer le nombre final des cluster d, $d = Ddiv\epsilon$
construire les clusters :
while $C \leq 1$ **do**
    sélectionner les ε attributs classés en premiers par rapport l'attribut v_{index} dans la ligne numéro $index$ dans la matrice M.
    mettre les attributs sélectionné dans le cluster C_k.
    calculer le centroid g_k correspondant au cluster C_k.
    mettre à jour le nombre d'attributs $C \leftarrow C - \varepsilon$
    mettre à jour l'indice $index \leftarrow index + \varepsilon + 1$
end while
déterminer les attributs communs entres les cluster $\forall\{k_1, k_2\} \in \{1..d\}, C_{k_1} \bigcap C_{k_2}$: intersection entre clusters
attribuer chacun des attributs commun au cluster le plus proche selon la distance euclidienne
recherche des centres g_k des clusters par maximisation de MI : appeler algorithme de maximisation
return les nouveaux attributs sont l'ensemble des centres g_k
return la transformation de la matrice de donnée $T(X) \in \mathbb{R}^{dL}$

La procédure de recherche du centre optimal pour chaque cluster consiste à appliquer l'algorithme de descente de gradient stochastique pour optimiser IM entre l'ensemble des attributs dans un cluster et leur centre recherché v_{n+1}. Un ensemble aléatoire d'attributs dans chaque cluster est choisi. On initialise par la suite le vecteur des poids w_0 aléatoirement. La dérivée de IM est calculée pour déterminer la pente. Le poids suivant est ainsi calculé en fonction du poids précédent. Si la condition d'arrêt est vraie, l'algorithme s'arrête sinon il recherche un nouveau poids. Enfin, le poids optimal sera déterminé et le nouveau centroid est calculé.

Algorithm 4 Algorithme de maximisation de IM

1: Choisir un sous ensemble aléatoire d'attributs dans chaque cluster
2: Initialiser la matrice des poids w_0
3: Calculer le dérivée de MI $\frac{dI}{dv_{n+1}}$ en fonction de $W(:, j)$ et, ci
4: Calculer le poids suivant $W(:, j+1) = W(:, j) + c * d'$
5: Tester si $norm(W(:, j+1) - W(:, j)) < delta$ donc break,
sinon retour à étape 3
6: déterminer le poids optimal $wopt = (W(:, end))$
7: **return** le bary centre $bary = wopt * ci$

2.7 Conclusion

Dans ce chapitre, nous avons présenté les principales contributions apportées au domaine d'analyse de données pour les problèmes de classification. Notre approche étant totalement non supervisée, ne présente pas les même limitations que les autres méthodes supervisées notamment la méthode LDA. De plus, notre méthode ne procure d'aucune information supplémentaire et n'assume pas des hypothèses sur la distribution des données comme le fait l'ACP ou la LDA.

Dans notre méthode, nous appliquons un algorithme de clustering qui permettra d'approximer la structure inhérente des attributs. Pour ce faire, nous utilisons la nouvelle mesure de similarité qu'on a conçu. Cette mesure de similarité utilise l'analyse de la tendance de chaque attribut pour fournir à chacun une signature. Les signatures sont comparées une à une pour construire la matrice de similarité totale.

Dans une deuxième partie, nous avons abordé le concept de l'information mutuelle issue de la théorie de l'information. Cette approche permet d'identifier toute liaison linéaire ou non linéaire susceptible d'exister entre les variables. Ainsi, nous avons développé un critère à base de IM à maximiser pour pouvoir identifier les centres appropriés pour chaque groupe d'attributs. Ces nouveaux attributs seront les nouveaux descripteurs de la base d'observations.

Chapitre 3

Expérimentation

Contents

3.1 Introduction

Dans ce chapitre, on présente les principaux résultats obtenus lors de la validation de notre approche d'extraction à base de la classification non supervisée des attributs, sur différents problèmes de classification. Notre méthode est une méthode d'extraction non supervisée, qui fonctionne sur des données issues des problèmes de classification diversifiés. Elle ne présente pas des contraintes sur la structure des données ni sur leur distribution.

La première partie est consacrée aux problèmes de classification issues des bases réelles prises du Centre d'apprentissage automatique et de systèmes intelligents (UCI machine learning repository) disponible sur le lien http ://archive.ics.uci.edu/ml/.

Elle consiste en la validation sur six différentes bases : Sonar, Pima, Breast cancer, Inonosphere, Wine et Iris. Chacune de ces bases représente un défit de fléau de dimension pour les algorithmes de classification.
Une deuxième partie est consacrée aux résultats obtenus de l'expérimentation sur des bases de reconnaissance de visage : Yale et ORL bases. Ces bases comportent un grand nombre d'attributs y compris des données non utiles, redondantes et bruitées.
Une troisième partie consiste à valider notre approche sur un problème de diagnostic des procédés industriels, spécifiquement le processus "Tenessee Easteman Processus" (TEP).

3.2 Expérimentation sur les bases UCI

Dans cette section, on a implémenté notre méthode d'extraction d'attributs à base de clustering avec et sans MI, pour plusieurs benchmark issus de UCI machine learning repository [Blake et al.,]. On procède par appliquer l'extraction et valider avec une procédure de classification supervisée. Pour les problèmes binaires, on a utilisé le SVM (donné par Matlab toolbox)en utilisant un noyau gaussien avec le paramètre σ est définit empiriquement après plusieurs tests. Quant aux problèmes multi-classes, on a utilisé le classificateur K-Nearest-Neighbors (KNN). On a utilisé la validation croisée (k-fold cross validation) comme stratégie de validation. Les bases sur lesquelles on a expérimenté, sont présentés dans la section suivante.

3.2.1 Présentation des bases UCI

- La base Sonar : elle a été construite pour discriminer entre les signaux sonar rebondis en retour de la part d'un métal cylindrique et ceux rebondis de la part d'un rocher [Blake et al.,]. Elle consiste en 208 instances décrites avec 60 attributs. Ces données sont divisées en deux classes non séparables linéairement qui sont mine ou rock .
- La base Pima Indian Diabetes : elle est constituée de 768 instances décrites par 8 attributs. Les observations sont réparties en 2 classes d'une façon non équilibrée. En effet, la première classe contient 500 instances alors que la deuxième classe contient 268 (presque la moitié de la première classe).
- La base Breast Cancer : elle est constituée de 699 observations provenant de tissus bénins et de tissus malignes et qui sont répartis en deux classes non équi-

librées. En effet, la première classe contient 458 instances et la deuxième classe contient 241 instances seulement. Il existe 16 valeurs manquantes qui ont été remplacées par la valeur moyenne de l'attribut correspondant comme dans le travail de [Kwak and Choi, 2002].

- La base Ionosphère : Cette base provient d'un système de radar constitué de 16 antennes de haute fréquence. La cible étant les électrons libres dans l'ionosphère. Les bons retours des radars sont ceux mettant en évidence un type de structure particulier dans l'ionosphère. Les mauvais retours sont ceux qui ne montrent rien et leurs signaux passent à travers l'ionosphère. La base est alors décrite par 34 attributs et contient 351 instances.
- La base Wine : elle contient les résultats d'une analyse chimique des vins produits dans la même région en Italie, mais provenants de trois cultivateurs différents [Blake et al.,]. Elle est constituée de 178 données décrites par 13 attributs. Les données sont divisées en 3 classes non séparables linéairement, contenant respectivement 59, 79 et 48 instances.
- La base Iris : Cette base est constituée de 150 instances décrites par 4 attributs. Les données sont issues de 3 classes différentes de la plante Iris [Blake et al.,]. Une des classes est linéairement séparable des deux autres classes alors que les autres sont non linéairement séparables.

3.2.2 Résultats et discussion

Dans cette section, on présente les différents résultats obtenus lors de l'expérimentation de notre méthode d'extraction d'attributs sur les bases de UCI. En fait, notre approche d'extraction d'attributs consiste à appliquer une classification non supervisée pour grouper les attributs similaires. Suite à la formation des groupes d'attributs similaires, un centre sera calculé et deviendra lui même le nouveau attribut extrait. Le calcul du centre se fait habituellement par un calcul de la moyenne des attributs dans un cluster. Pour améliorer la qualité d'extraction de notre approche , on a utilisé la maximisation de MI entre les attributs dans un cluster et leur centre. Ceci permettra de rechercher la combinaison optimale à appliquer pour trouver le bon centre. Ce centre sera le nouveau attribut qui décidera de la qualité de classification des observations par la suite.

Nous allons comparer les performances de notre approche en terme de taux d'erreur de classification avec deux méthodes standard d'extraction non supervisée qui sont ACP et ICA. L'ACP étant une méthode linéaire, qui se base sur le calcul des corré-

lations entres les attributs. L'ICA étant une méthode non linéaire, qui se base sur le calcul de l'information mutuelle entre les attributs (une méthode forte pour révéler les structures non gaussienne dans les données [Torkkola, 2003])

Base Sonar

On a évalué les performances en classification de notre méthode en utilisant le classificateur SVM avec un noyau gaussien. Ses paramètres sont définit tel que $\sigma = 1$ et $C = 1000$. La stratégie de validation utilisée est la validation croisée (*13-fold cross-validation*), comme a été fait dans [Kwak, 2007]. Les 208 instances sont divisés aléatoirement en 13 ensembles chacun contenant 16 instances. A chaque expérimentation, 12 ensembles sont utilisés pour l'apprentissage et le $13^{éme}$ ensemble restant sera réservé pour la phase de validation. L'expérimentation est refaite 13 fois tel que chaque ensemble apparait une fois dans la base de test. Le tableau 3.1 présente la précision de classification pour différents nombre d'attributs extraits. La dernière ligne présente la précision de classification sans extraction (i.e avec tout l'ensemble des attributs originaux).
Les performances de FECM sont largement meilleures que les performances des méthodes PCA et ICA sauf dans le cas de la dimension 1. Dans ce cas, la méthode ICA dépasse en performances les autres.

TABLE 3.1 – Précision de classification sur la base Sonar

No. d'attributs	Précision de classification (SVM)			
	PCA	ICA	FEMC (sans MI)	FEMC (avec MI)
1	58.6	67.2	60.60	65.94
3	54.7	69.7	71.68	75.05
6	63.0	70.2	77.39	77.89
9	70.2	68.7	82.38	82.32
12	75.1	71.7	85.02	81.40
60	82.7			

On note de plus que dans le cas de la dimension 9 et 12, FECM a pu obtenir un taux de bonne classification supérieur de la précision initiale (82%). Ceci est meilleur que PCA et ICA qui n'arrivent qu'à environ 70%.
En utilisant FECM avec MI, notre méthode arrive à une meilleure précision de classification, spécialement pour le cas de dimension 1 : elle arrive à 66% similaire à celle de ICA. Ainsi, MI augmente la précision de FECM surtout dans les faibles

dimensions comme la dimension 1 et 3.
Dans ce cas de l'extraction d'un seul attribut FECM n'est pas la méthode la plus effective, puisque elle n'arrive pas à extraire un seul attribut qui soit capable de séparer les deux classes. Toutes fois notre méthode reste supérieure aux autres méthodes dans des dimension plus grandes que la dimension 1.
Étant donné que le concept de notre approche est de former des groupes d'attributs similaires. Dans ce cas, l'extraction d'un nombre très faible d'attributs signifie que les attributs seront forcer à être groupées dans un petit nombre de clusters. L'utilisation de MI dans ce cas permet de trouver des combinaisons non linéaire pour retrouver un centre adéquat à chaque cluster. Ainsi, Dans le cas de faible dimension, l'utilisation de MI améliore largement la qualité de classification de notre méthode FECM.

Base Pima

La base Pima est une base qui présente des données bien séparables, ce qui est adéquat pour la méthode supervisée LDA avec un taux de (75%) pour un seul attribut. Le classificateur utilisé dans ce problème est le SVM à noyau gaussien. Ses paramètres sont : $\sigma = 10$ et $C = 1000$. La stratégie de validation utilisée est la validation croisée : 10 *fold cross-validation.*
Les résultats sont présenté dans le tableau 3.2 en terme de précision de classification pour différents nombre d'attributs extraits. La dernière ligne présente la précision de classification sans extraction (i.e avec tout l'ensemble des attributs originaux).
On peut remarquer depuis le tableau que la précision de classification de PCA et

Table 3.2 – Précision de classification sur la base Pima

No. d'attributs	Précision de classification (SVM)			
	PCA	ICA	FEMC (sans MI)	FEMC(avec MI)
1	66.3	73.2	68.08	67.80
2	75.1	76.7	69.72	69.85
3	75.5	76.8	75.72	75.72
5	75.5	77.2	76.80	82.32
8	78.0			

ICA deviennent de plus en plus proches, que le nombre d'attributs extraits devient plus grand.
Les performances de FECM sont meilleures que les performances de la méthode PCA et sont proches des performance de la méthode ICA dans les faibles dimensions 3 et 5. Toute fois, la méthode ICA dépasse la méthode PCA et FECM pour la

dimension 1. La méthode ICA est une méthode non linéaire arrive ainsi à extraire non linéairement un seul attribut avec un taux de bonne classification égal à 73%. Toutefois, dans de plus grande dimension (dimension 3 et 5), notre méthode se rattrape rapidement et arrive à avoir des résultats meilleurs que ceux de l'ACP et proches de ceux de ICA. En utilisant la MI, les précision de FECM a largement augmenté surtout pour la dimension 5, où notre méthode dépasse les deux méthodes PCA et ICA ainsi que la précision initiale qui est (70%) en obtenant 83.32%. L'extraction du centre en utilisant la MI est alors bénéfique pour notre procédure d'extraction d'attributs et permis d'avoir des attributs qui séparent au mieux les classes d'observations.

Base Breast Cancer

En fait, la base Breast cancer comporte des données manquantes, et qui ont été remplacé par la moyenne des attributs correspondant. Les valeurs prises par les attributs dans ces observations ne sont pas précises et donc les attributs sont légèrement bruité.
Une stratégie de validation croisée a été adopté avec $k = 10$ et on a utilisé un noyau Gaussien pour le classificateur SVM avec $\sigma = 0.01$. Les résultats de classification sont donnée par tableau 3.3. IL s'agit du taux de bonne classification des trois approches d'extraction PCA, ICA et FECM, pour différents nombre d'attributs extraits. La dernière ligne présente la précision de classification sans extraction (i.e avec tout l'ensemble des attributs originaux).
On peut remarquer depuis le tableau que juste avec un seul attribut extrait, la méthode FECM arrive à avoir le maximum de précision de classification qui est de 96.86%. Pour un grand nombre d'attributs extraits, la méthode PCA dépasse les deux méthodes ICA et FECM et obtient le maximum de précision de classification avec 6 attributs.
La MI dans ce cas améliore légèrement les performances de la méthode FECM. On retrouve une grande similarité entre les attributs 2 et 4 avec un indice proche de 1. en revenant sur la description de la base, l'attribut 2 correspond à l'age du malade, et l'attribut correspond à la taille de la tumeur. La relation entre ces deux facteurs est bien établie dans le diagnostic de la maladie. En réalisant une analyse de corrélation, on retrouve que tous les attributs sont fortement corrélés entre eux.

TABLE 3.3 – Précision de classification sur la base Breast Cancer

No. d'attributs	Précision de classification (SVM)			
	PCA	ICA	FEMC (sans MI)	FEMC(avec MI)
1	85.8	85.1	96.72	96.86
2	94.7	90.3	96.57	96.71
3	95.9	91.3	94.71	94.28
6	96.6	94.3	85.11	85.11
9	96.6			

Base Ionosphere

Dans le tableau 3.4, on présente les résultats de l'application des trois approches d'extraction à savoir ACP, ICA et FECM sous forme de taux de bonne classification pour différents nombre d'attributs extraits. La dernière ligne présente la précision de classification sans extraction (i.e avec tout l'ensemble des attributs originaux). On a utilisé une validation croisée avec $k = 10$ et $\sigma = 0.01$.
On peut remarquer depuis le tableau que juste avec un seul attribut extrait, la méthode FECM dépasse largement les autres méthodes.
Pour les dimensions les plus grandes, FECM obtient des performances similaires ou légèrement supérieures que les méthodes PCA et ICA. Elle arrive à atteindre la meilleure précision de classification 89% avec 12 attributs extraits.
Dans les faibles dimensions, on remarque que la FECM avec MI atteint la meilleure précision 76.09% spécifiquement pour le nombre d'attributs 1.

TABLE 3.4 – Précision de classification sur la base Ionosphere

No. d'attributs	Précision de classification (SVM)			
	PCA	ICA	FEMC(sans MI)	FEMC(avec MI)
1	64.07	61.28	72.09	76.09
3	85.21	81.80	75.20	83.50
6	84.79	86.05	85.21	85.49
9	84.83	86.52	87.52	87.80
12	86.31	88.04	89.20	89.20
34	91.73			

Base Wine

On note que la base Wine est considéré comme un problème non séparable linéairement. Le classificateur KNN avec un seul voisin a été utilisé pour la phase de classification. On a utilisé une validation croisée avec $k = 10$. Le tableau 3.5 présente la précision de classification pour différents nombre d'attributs extraits. La

dernière ligne présente la précision de classification sans extraction (i.e avec tout l'ensemble des attributs originaux).
Les performances de la méthode FECM sont largement meilleures que les méthodes PCA et ICA pour les faibles dimensions. Sachant que dans le cas de la dimension 2, la méthode ICA dépasse notre méthode FECM. Pour des dimensions plus grande, FECM réalise la meilleure précision de classification et devient proche de la précision initiale de 80.27%.
En utilisant MI, la méthode FECM atteint meilleure performance spécialement dans le cas de la dimension 3, où elle arrive à 81.49% qui est meilleure que la précision initiale.

TABLE 3.5 – Précision de classification sur la base Wine

No. d'attributs	Précision de classification(KNN)			
	PCA	ICA	FEMC(sans MI)	FEMC(avec MI)
1	67.93	67.42	71.02	71.88
2	71.94	73.84	72.36	76.25
3	72.48	75.22	78.19	81.49
5	75.74	91.01	79.92	80.45
13	80.27			

Base Iris

On a utilisé la validation croisée comme stratégie de validation et le classificateur KNN pour les problème multi-classe, avec un seul voisin pour la tâche de classification. Le paramètre k-fold est fixé à $k = 10$. Dans le tableau 3.6, on présente les résultats de classification des trois approches d'extraction sous forme de précision de classification pour différents nombre d'attributs extraits. La dernière ligne présente la précision de classification sans extraction (i.e avec tout l'ensemble des attributs originaux).
Les résultats montrent que la méthode FECM peut atteindre le maximum de précision de classification avec seulement deux attributs. De plus, pour de plus grande dimensions, FECM arrive à avoir une meilleur précision que la précision de classification initiale.
FECM avec MI atteint la meilleure précision de classification pour la dimension 1 avec 89.33%, quant aux autres dimensions, MI améliore légèrement les performance de FECM.

TABLE 3.6 – Précision de classification sur la base Iris

No. attributs	Précision de classification (KNN)			
	PCA	ICA	FEMC(sans MI)	FEMC(avec MI)
1	80.0	88.67	87.33	89.33
2	94.67	95.33	97.33	96.67
3	96.0	96.0	98.0	98.0
4	96.0			

3.3 Expérimentation sur les bases de reconnaissance de visages

3.3.1 Présentation des bases de reconnaissance de visages

Le problème de reconnaissance de visage

La reconnaissance de visage est un des problème les plus étudié dans le domaine de reconnaissance de formes. Le problème consiste en la grande dimension. L'extraction d'attributs devient inévitable. De plus, l'extraction d'attributs pour la reconnaissance de visage peut être utilisée comme un outil de codage pour la compression des images.

Plusieurs méthodes ont été appliquées sur ce problème avec succès. Parmi eux, les méthodes Eigenfaces (basée sur l'application de l'ACP)et Fisherfaces (basée sur l'application de LDA) sont les plus populaires [Xiang and Huang, 2006]. Elles permettent une caractérisation effective d'un espace de faible dimension tout en préservant la qualité perceptive d'une image de très grande dimension. La méthode ICA a prouvé aussi de très bonne performances surtout pour la reconnaissance de visage avec des changements en pose et pour la classification des actions faciales [Kwak, 2003].

Dans cette section, notre méthode d'extraction à base de clustering a été appliquée pour le problème de reconnaissance de visage. Ses performances ont été comparées avec les autres méthodes PCA, ICA et LDA.

Pour appliquer notre méthode, on a besoin tout d'abord de déterminer la matrice d'attributs originale, à partir de laquelle les nouveaux attributs seront extraits. Il existe plusieurs méthodes en fait comme les ondelettes, analyse de fourrier [Tzanakou, 2001]. La méthode la plus intuitive est de considérer chaque pixel comme un attribut. C'est la méthode la plus simple sans perdre trop d'information. Pourtant, la dimension de l'espace d'attributs devient trop large pour être manipulée facilement. Ainsi, chaque image a été re-dimensionnée en une taille acceptable.

On mentionne que avant d'appliquer LDA et ICA, on procède selon la figure 3.1 par appliquer ACP pour réduire le bruit dans l'image (whitening).

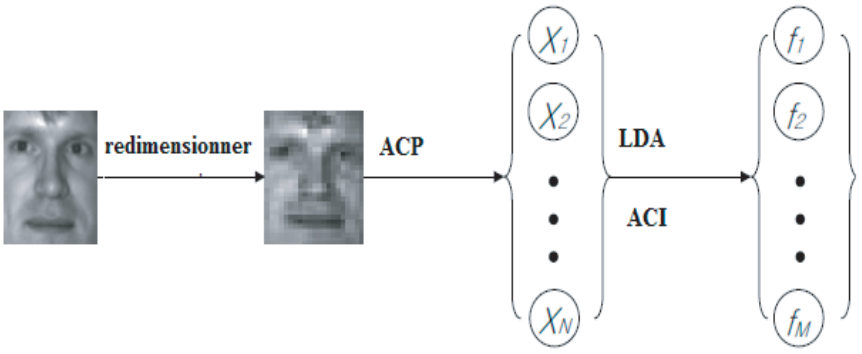

FIGURE 3.1 – Procédure Expérimentale pour la reconnaissance de visage

- Base Yale : La base Yale contient un ensemble d'images de visages de 15 personnes. Chaque personne étant caractérisée par 11 images, cela fait un total de 165 images. Les images pour chaque personne ont été acquises selon des conditions différentes : de lumière, d'expressions du visage (yeux ouverts/yeux fermés, avec/sans sourire..) et de détails de visages (avec/sans lunettes). La figure 3.2 représente un échantillon d'images dans la base.
- Base ORL : La base ORL (Olivetti Research Laboratory) contient un ensemble d'images de visages de 40 personnes. Chaque personne étant caractérisée par 10 images, cela fait un total de 400 images. Les images pour chaque personne ont été acquises selon des conditions différentes : de lumière, d'expressions du visage (yeux ouverts/yeux fermés,avec/sans sourire..) et de détails de visages (avec/sans lunettes). La figure 3.2 représente un échantillon d'images dans la base.

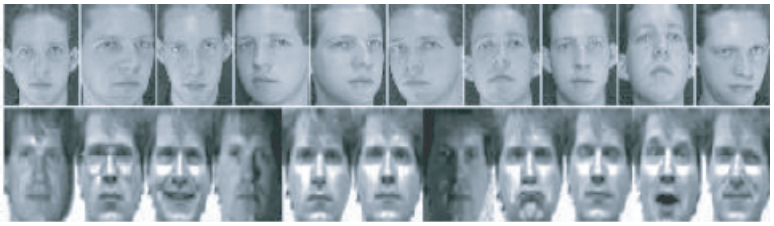

FIGURE 3.2 – Bases ORL et Yale

3.3.2 Résultats et discussion

3.3.2.1 Base Yale

La taille de chaque image est de $320 * 246$. Chaque image a été réduite afin d'avoir un nombre initial de pixels de 784. Les performances en classification des attributs extraits par PCA, ICA et FECM sont obtenues par la validation "leave one out". Le nombre d'attributs extraits par ICA est le même que PCA. En effet, PCA est utilisée comme phase de pré-traitement avant d'appliquer la méthode ICA.
Dans l'expérimentation, PCA a été initialement appliquée sur les 784 attributs et les premiers 30 composantes principales sont utilisées à l'entrée du processus de classification par KNN. Le tableau 3.7 contient les différents erreurs de classification correspondant à chaque méthode avec le nombre d'attributs extraits. On précise que la première ligne rapporte les résultats du classification du KNN sans extraction d'attributs.
On remarque que malgré que ICA et PCA produisent le même nombre d'attributs, la méthode ICA est légèrement plus performante que PCA. Notre méthode FECM réalise une meilleure précision de classification(avec un taux d'erreur inférieur au taux initial)avec seulement un nombre d'attributs égale à 14. Les performance de la méthode d'extraction supervisée LDA dépasse toutes les autres méthodes en matière de taux d'erreur mais elle souffre de la limitation en nombre d'attributs : elle ne peut produire qu'un nombre d'attributs égale à : nombre de classe - 1, qui est 14 dans notre cas. Les autres méthodes (ACP et ICA) ne souffre pas en effet de cette limitations, puisque le nombre d'attributs 30 est le nombre optimal qui contient le maximum de la variance (80%).

TABLE 3.7 – Précision de classification sur la base Yale

Méthode	taux d'erreur (%)	No. d'attributs
KNN	21.82	783
PCA	24.85	30
ICA	23.03	30
LDA	8.48	14
FECM	21.00	14

3.3.2.2 Base ORL

Le même protocole d'expérimentation appliquée sur la base Yale a été reproduit içi avec la base ORL. Le schéma de validation utilisé est le "leave one out". PCA est

utilisée comme phase de pré-traitement avant d'appliquer la méthode ICA.
PCA a été initialement appliquée sur les 952 attributs et les premiers 40 composantes principales sont utilisées par la suite à l'entrée du processus de classification par KNN. Le taux d'erreur de classification est donné par le tableau suivant 3.7. On y compare les performances des quatre approche : ACP, ACI, LDA et FECM en terme de nombre d'attributs et taux d'erreur de classification.
Depuis le tableau, on remarque que la méthode linéaire ACP a dépassé la méthode non linéaire ICA en terme de taux d'erreur pour le même nombre d'attributs qui est égale à 40. En effet, elle produit un taux d'erreur de 4% inférieur à celui de l'ICA qui est de 4.25%. On note que l'ICA coïncide avec l'ACP dans le cas de données gaussiennes mais elle est plus performante dans le cas de données non gaussienne. Notre méthode FECM réalise un taux d'erreur légèrement supérieur à ceux de PCA et ICA mais avec seulement un nombre d'attributs beaucoup inférieur à celui du ACP et ICA et aussi LDA, et qui est égal à 23.
Le taux d'erreur produit par la méthode d'extraction supervisée LDA est inférieur aux taux d'erreurs de toutes les autres méthodes. Pourtant, elle ne peut produire qu'un nombre d'attributs égal à : $nombredeclasse - 1$ qui est 39 dans notre cas.

TABLE 3.8 – Précision de classification sur la base ORL

Méthode	taux d'erreur (%)	No. d'attributs
KNN	3.00	952
PCA	4.00	40
ICA	4.25	40
LDA	2.00	39
FECM	5.00	23

3.4 Expérimentation sur le diagnostic d'un procédé industriel : TEP

3.4.1 Présentation du procédé TEP

Le Tennessee Eastman Process (TEP) est un procédé qui a été modélisé par la société Eastman Chemical Company. Il a été conçu afin de fournir une simulation d'un procédé industriel réel pour le test de méthodes d'asservissements et/ou de surveillance de procédé [Verron, 2007]. Il modélise un procédé chimique réel mais qui a été modifié pour assurer la confidentialité du procédé réel. Le TEP

a été très utilisé pour comparer certaines méthodes de surveillance et le diagnostic des défauts [Kulkarni et al., 2005], [Chiang et al., 2004], [Verron et al., 2008], [Kano et al., 2000].

Ce procédé présenté dans la figure 3.3 est composé de cinq éléments principaux : un réacteur, un compresseur, un décapeur, un séparateur et un condenseur.

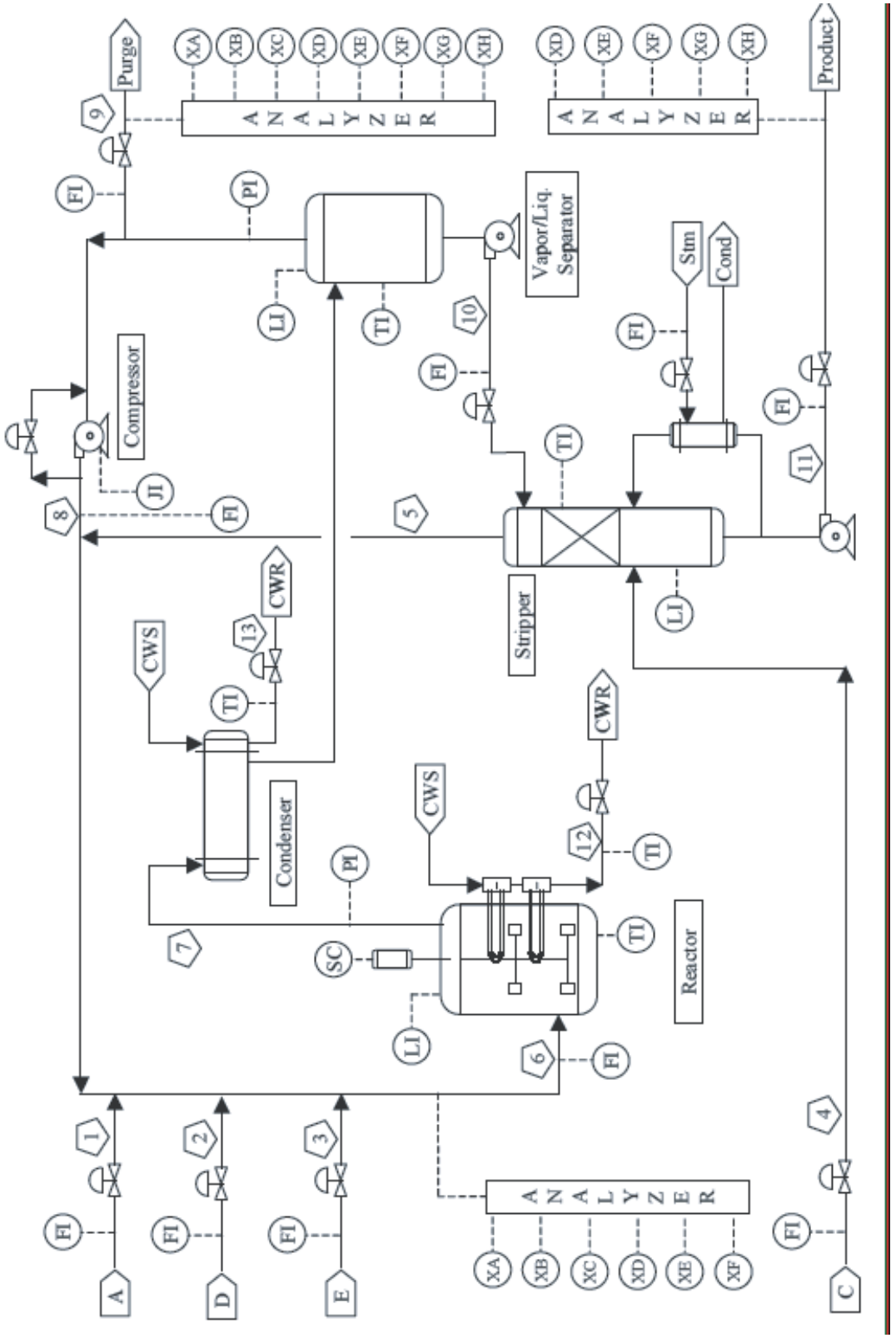

FIGURE 3.3 – Tennessee Eastman Process

Il comporte 53 variables en total dont 12 variables d'asservissement et 41 variables mesurables. Parmi les 41 variables mesurables, il y a 22 qui sont des variables mesurables en continu (ce sont les valeurs des capteurs du procédé) donnés par le tableau 3.9. Les autres variables sont des mesures de compositions telles que des concentrations, disponibles en échantillonné et listés dans le tableau 3.10. Les 12 variables d'asservissement sont données dans la table 3.11.

TABLE 3.9 – Variables de mesures en continu

Variable	Description (%)	Unité
XMES(1)	Débit d'alimentation en A	kscmh
XMES(2)	Débit d'alimentation en D	kg/hr
XMES(3)	Débit d'alimentation en E	kg/hr
XMES(4)	Débit d'alimentation total	kscmh
XMES(5)	Débit de recyclage	kscmh
XMES(6)	Débit d'alimentation du réacteur	kscmh
XMES(7)	Pression du réacteur	kPa
XMES(8)	Niveau du réacteur	%
XMES(9)	Température du réacteur	C
XMES(10)	Débit de purge	kscmh
XMES(11)	Température du séparateur	C
XMES(12)	Niveau du séparateur	%
XMES(13)	Pression du séparateur	kPa
XMES(14)	Débit du séparateur	m3/hr
XMES(15)	Niveau du décapeur	%
XMES(16)	Pression du décapeur	kPa
XMES(17)	Débit du décapeur	m3/hr
XMES(18)	Température du séparateur	C
XMES(19)	Débit de gaz du séparateur	kg/hr
XMES(20)	Puissance du compresseur	kW
XMES(21)	Température de ref.liq. en sortie de réacteur	C
XMES(22)	Température de ref.liq. en sortie de séparateur	C

TABLE 3.10 – Variables de mesures échantillonnées

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

TABLE 3.11 – Variables de contrôle du TEP

Variable	Description (%)	Unité
XC(1)	Débit d'alimentation en D	kscmh
XC(2)	Débit d'alimentation en E	kscmh
XC(3)	Débit d'alimentation en A	kscmh
XC(4)	Débit d'alimentation en A et C	kscmh
XC(5)	Valve de recyclage du compresseur	%
XC(6)	Valve de purge	%
XC(7)	Débit d'alimentation du séparateur	m3/hr
XC(8)	Débit d'alimentation du réacteur	m3/hr
XC(9)	Valve du décapeur	%
XC(10)	Débit de refroidissement liquide de condensateur	m3/hr
XC(11)	Débit de refroidissement liquide de condensateur	m3/hr
XC(12)	Vitesse de l'agitateur	tr/min

L'intérêt du TEP pour le diagnostic se manifeste via les 20 fautes différentes à lesquelles il peut être soumis. Ces fautes sont de diverses natures : saut en échelon de certaines variables internes, augmentation de la variabilité de certaines autres, ou bien fautes d'actionneurs tel qu'une vanne bloquée. La description de ces fautes est donnée par le tableau 3.12. On peut observer que les fautes de F16 à F20 sont inconnues.

TABLE 3.12 – Les différentes fautes du TEP

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

On peut remarquer que toutes ces variables possèdent un certain bruit, et que certaines d'entre elles suivent une certaine dynamique.

Afin de comprendre plus ce que présente une faute (une classe), nous prenons le cas de la faute F4. Cette faute est une augmentation de la température du liquide de refroidissement à l'entrée du réacteur. Comme pour les autres fautes, on remarque qu'elle agit sur une variable qui n'est pas pris en compte dans le procédé. En effet, aucune variable surveillée ne donne la température du liquide de refroidissement à l'entrée du réacteur. Cependant cette faute engendre des répercussions sur deux variables incluses dans la surveillance : les variables 9 et 51, respectivement la température du réacteur et le débit de son refroidissement liquide.

La figure 3.4 donne la comparaison des variables 9 et 51 pour le cas de fonctionnement normal et pour le cas de la faute $F4$. Sur les graphiques (c) et (d), la faute $F4$ est introduite à la $161^{éme}$ observation.

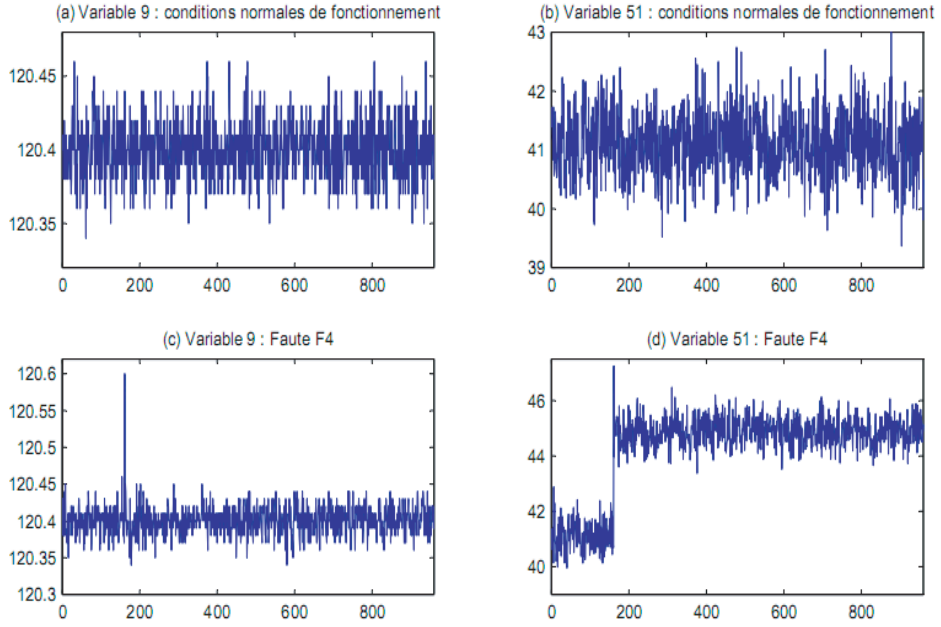

FIGURE 3.4 – Comparaison des variables 9 et 51 pour le fonctionnement normal et pour la faute F4

Il est possible de comprendre le comportement des variables 9 et 51 en réponse à l'introduction de la faute F4 dans le procédé. Dés son apparition (observation 161), la faute 4 (augmentation de la température d'entrée du liquide de refroidissement du réacteur) engendre une augmentation de la température du réacteur.
Ceci est bien visible sur le graphique(c). On observe que la température du réacteur augmente jusqu'à 120.6 ˚ C alors qu'elle oscillait normalement autour de 120.4 ˚ C. En effet, puisque la température du liquide de refroidissement est plus élevée, l'échange de chaleur entre le liquide et le réacteur est plus faible, engendrant alors une augmentation de la température dans celui-ci. Au vu de cette augmentation de température dans le réacteur, l'asservissement du TEP accroît alors le débit du refroidissement liquide d'environ $41m^3/hr$ à environ $45m^3/hr$ (visible sur la variable 51, graphique(d)). Puisque le débit augmente, la quantité de chaleur évacuée du réacteur redevient normale et la température du réacteur retourne à son niveau normal de fonctionnement(aux alentour de 120.4 ˚ C).
Sur la figure, on observe que la faute F4 engendre des conséquences très visibles sur les variables du procédé. Ce type de faute devrait donc être facilement détecté. Cependant, certaines fautes n'entraînent pas de changements si brutaux sur les variables du procédé, rendant la détection moins évidente.

3.4.2 Résultats et discussion

De nombreux tests de la méthode ont été réalisés sur différents problèmes de classification. Le diagnostic industriel peut être approché comme un problème de classification automatique [Chiang et al., 2004], [Verron et al., 2008], [Boucuelid, 2007]. Un classificateur peut alors être appliqué sur une base de données constitué des échantillons mesurées lors du fonctionnement du système. Il considère les fautes comme des classes d'observations. Chaque échantillon (observation) est décrit par un nombre de paramètres qui représentent les attributs ou les variables. Ainsi, comme tout problème de classification automatique, notre approche d'extraction peut être favorable puisqu'elle permet de réduire le nombre de variables et améliorer la précision de classification. De plus, l'extraction permettra ainsi de réduire la complexité du classificateur et le temps de traitement. Ceci peut induire une version on line plus rapide et efficace.
L'application de notre méthode d'extraction d'attributs FECM présentée au chapitre précédent requiert une base de données regroupant des observations de période de fonctionnement normal ainsi que des observations des différentes fautes déjà connues. Ce procédé complexe implique 53 variables et 20 types de fautes différentes.
Nous avons repris les données utilisées dans le livre de Chiang [Chiang et al., 2004]. Elles proviennent du TEP couplée à la structure d'asservissement de Lyman et Georgakis [Lyman and Georgakis, 1995]. L'intérêt de ces données est qu'elles sont disponibles en ligne à l'adresse suivante http ://brahms.scs.uiuc.edu. Ces données se présentent ainsi (voir tableau 3.13) : 480 observations d'apprentissage pour chaque type de faute ainsi que pour la période normale et 800 observations de test pour chaque type de faute ainsi que pour la période normale. Les données d'apprentissage ont été obtenues par simulation de chacune des fautes sur une période de 24 heures, alors que les données de test ont été obtenues sur une durée de 40 heures. La période d'échantillonnage de toutes a été fixée à 3 minutes. Il faut également préciser que les 53 variables n'ont pas été prises en comptes puisque la variable 12, la vitesse de l'agitateur, reste constante dans n'importe quelle situation (ceci étant dû au système d'asservissement). Ainsi seules 52 variables sont présentes dans les données utilisées.

3.4.2.1 Diagnostic supervisé

Afin de tester les performances en diagnostic supervisé de la méthode d'extraction de variables proposée, nous étudions tout d'abord les taux de classification initiaux

sans prendre en compte l'extraction des caractéristiques. Nous prenons en compte toutes les fautes en supposant que chaque observation est détectée. Les bases de données d'apprentissage et de validation sont celles décrites au tableau 3.13.

TABLE 3.13 – Données utilisées

Classe	Données d'apprentissage	Données de test
Normale	480	800
Faute 1	480	800
Faute 2	480	800
...	...	...
Faute k	480	800
...	...	...
Faute 20	480	800

Ainsi, nous avons 800 observations de chaque faute à classer, soit 16 000 observations. Nous allons nous intéresser en fait aux cas des F4, F9 et F11 qui constituent un problème de classification difficile.
Les résultats de cette classification sont présentés sous forme de trois matrices de confusion (matrice d'occurrence, matrice de précision et matrice de fiabilité). Les résultats de classification par différents classificateurs linéaires et non linéaires sont aussi présentés. En effet, la matrice d'occurrence donne pour chaque colonne testée (représentant 800 observations de la faute F_i) les différents classements du classificateur. Ainsi, la trace de cette matrice représente le nombre de bonnes classifications. Nous présentons également la matrice de précision. Cette matrice est construite en divisant chaque cellule de la matrice d'occurrence par la somme de colonne (ici 800 observations pour chaque colonne), et elle est approximé en pourcentage. Enfin, nous présentons la matrice de fiabilité, construite de la même manière que la matrice de précision, mais en prenant la somme de la ligne, et non plus de la colonne.
Par la suite, nous présenterons les taux de classification après l'extraction des variables ainsi que les trois matrices de confusion pour comparaison. Une deuxième comparaison portera sur les performances des différents classificateurs combiné avec deux différentes méthodes d'extraction LDA et ACP et notre méthode FECM.

3.4.2.2 Diagnostic sur les fautes 4, 9 et 11 sans extraction de variables

Nous avons choisit dans cette section de travailler sur 3 types de fautes (F4, F9, F11) car elles présentent un cas de chevauchement de données, et donc elles ne sont pas facilement séparables.

En effet, les fautes F4 et F11 sont sélectionnés car toutes les deux sont associées au problème "reactor cooling water inlet temperature", mais elles sont différentes en terme de type de faute (step change for F4, random variation for F11). Quand les variables fautives pour ces deux fautes (les variables qui se comportent anormalement par rapport aux données normales) sont dessinées l'une contre l'autre, le chevauchement est observé. La faute F9 est associée avec random variation in D feed temperature. Les variables fautives pour cette faute sont différentes de ceux des fautes F4 et F11. Ainsi, ces trois fautes sont une bonne représentation de données chevauchées présentées dans la figure3.5.

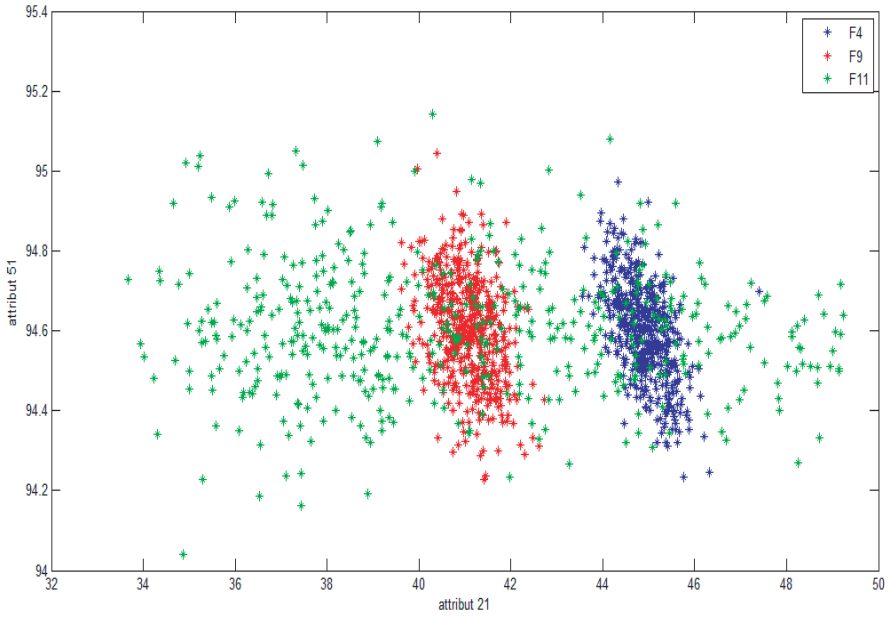

FIGURE 3.5 – Chevauchement entre les trois fautes : F4, F9 et F11 dans l'espace d'origine

Afin d'étudier l'influence de l'extraction des variables sur les performances du classificateur, nous évaluons tout d'abord le classificateur Séparateur à vaste marge (SVM) sur les faute (F4, F9 et F11) dans l'espace décrit par les 52 variables. Les résultats de cette évaluation sont donnés par le taux d'erreur de classification présentés dans le tableau 3.14.

TABLE 3.14 – Taux d'erreur de classification

Classificateur	T.d'erreur sur base d'apprentissage(%)	T.d'erreur sur base de test(%)
SVM	23.75	52.00
LDA	91.6	68.83
QDA	74.72	47.5
KNN	0.00	35.38

En guise de comparaison, on a présenté les résultats de classification des deux classificateur supervisés : Analyse discriminante linéaire (LDA) et Analyse quadratique discriminante (QDA) et un classificateur non supervisé k-plus proches voisins (KNN). De plus, on présente les matrices de confusion (précision, fiabilité et occurrence) données par le classificateur SVM.

TABLE 3.15 – Matrice d'occurrence exprimée en %

	F4	F9	F11
F4	607	79	114
F9	0	240	560
F11	178	317	305

TABLE 3.16 – Matrice de précision exprimée en %

	F4	F9	F11
F4	75.88	9.88	14.25
F9	0	30.00	70.00
F11	22.25	39.62	38.12

TABLE 3.17 – Matrice de fiabilité exprimée en %

	F4	F9	F11
F4	75.88	9.88	14.25
F9	0	30.00	70.00
F11	22.25	39.62	38.12

Sur la matrice de précision, on peut observer que les deux fautes (F9 et F11) sont très mal reconnues par le classificateur SVM avec un taux de précision inférieur à 40%. La faute F4 est moyennement reconnue puisque son taux de précision est de 75.88%. Il est cependant intéressant de remarquer que pour les fautes F9 et F11, les confusions portent principalement entre ces mêmes fautes : la faute F9 va être

classée comme F11 avec une fréquence de 70%. Le taux de reconnaissance global (moyenne de la diagonale) est de 38%. Cela signifie qu'en moyenne, si l'on présente cinq observations au classificateur, il trouve la bonne classe, et donc le bon diagnostic, pour seulement deux de ces observations.
Un autre critère important est la fiabilité du classificateur. Il s'agit du nombre de fois que le classificateur a bien classé une faute, sur le nombre de fois qu'il a déclaré cette faute. Prenons par exemple le cas de la faute F4, la précision du classificateur pour cette faute est de $\frac{607}{800} = 75.88\%$. Sa fiabilité vaut $\frac{607}{800} = 75.88\%$, montrant alors que si le classificateur attribue la classe F4 à une observation, alors on est certain (à 75.88%) que ceci est le bon diagnostic. Nous pouvons notamment observer que l'on ne peut pas accorder de confiance au diagnostic des fautes F9 et F11 puisque leur taux de fiabilité est d'environ 30%. Le taux de fiabilité globale (moyenne de la diagonale) étant de même de 38%.
Ceci indique bien que les caractéristiques de la base ne sont pas capturé par le modèle. Ainsi, pour simplifier le problème de classification et augmenter les performances du classificateur, une extraction d'attributs sera bénéfique.

3.4.2.3 Diagnostic sur les fautes 4, 9 et 11 avec extraction de variables

Dans cette section, on reprend le même problème initial qui est le diagnostic des fautes F4, F9 et F11. Vu que le classificateur SVM arrive difficilement à distinguer entre ces trois fautes comme évoquée dans la section précédente, on a procédé à effectuer une extraction d'attributs qui permettra d'améliorer les performances du classificateur.
Vu que les signaux chimiques sont plus complexes, on a dû changer l'analyse de tendance afin d'effectuer un filtrage au premier niveau et par la suite analyser le signe de la dérivée première et la dérivée seconde pour décrire le comportement de chaque attribut.
Nous avons comparé dans le tableau suivant 3.18 les taux d'erreurs après extraction pour les trois classificateurs : KNN avec $K = 1$, LDA et QDA, et le taux d'erreur sur base d'apprentissage et sur base de test pour le SVM avec noyau gaussien.

TABLE 3.18 – Taux d'erreur de classification avec extraction

Nombre d'attributs	Taux d'erreur (%)			
	SVM	KNN	LDA	QDA
21	53.96	59.25	49.67	43.42
10	49.17	58.88	47.50	35.50
4	44.20	54.50	48.17	45.79

D'après le tableau, on remarque que l'extraction d'attributs a considérablement amélioré la classification pour le SVM : on a passé d'un taux d'erreur de 54.62% à 44.20% et de 52 attributs à 4 attributs seulement. sur les deux figures suivantes 3.6 et 3.7, on retrouve les trois fautes projetées deux à deux sur les 4 axes retrouvés après extraction par FECM.

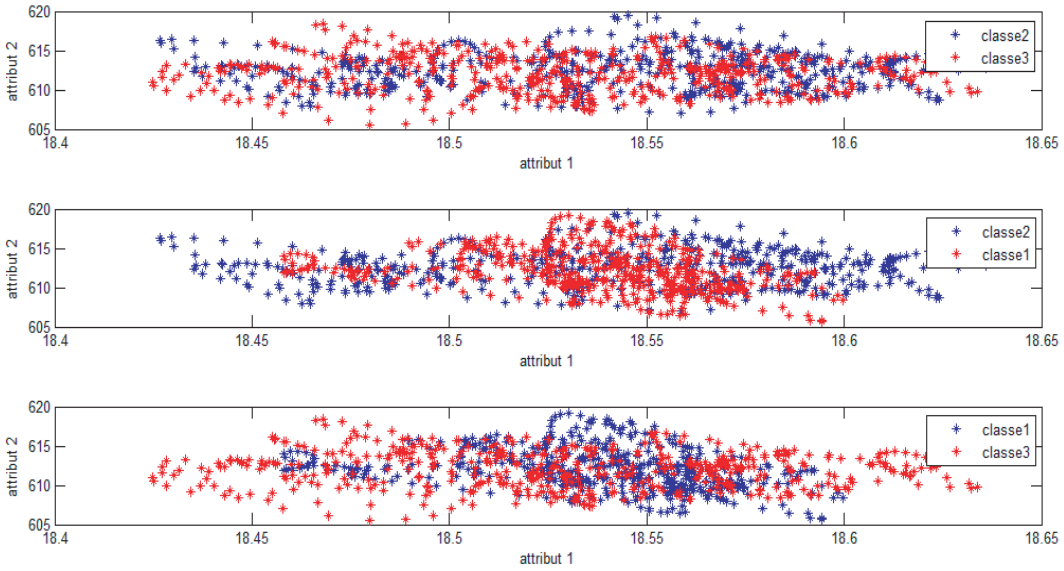

FIGURE 3.6 – Données projetées sur les deux premiers attributs extraits par notre approche

Le taux d'erreur sur la base d'apprentissage a passé de 0 à 21 %. En effet, on remarque que le taux d'erreur de classification sur la base d'apprentissage est de 0% malgré que le taux d'erreur sur la base de test est élevé 53.96%. Ceci indique que le classificateur est en sur-apprentissage. Une amélioration est bien noté à la dernière ligne où le taux d'erreur sur la base d'apprentissage est de 21% alors qu'on remarque que le taux d'erreur sur la base de test a diminué.

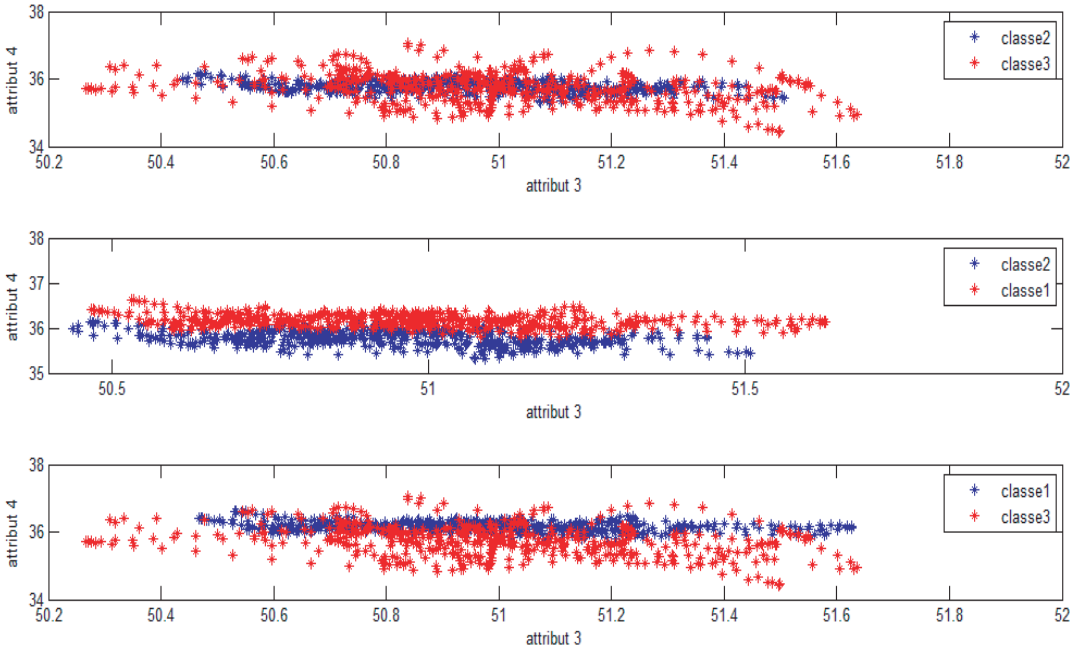

FIGURE 3.7 – Données projetées sur les deux derniers attributs extraits par notre approche

On remarque aussi que le processus d'extraction a amélioré les performances du classificateur linéaire et non supervisé KNN. Le taux d'erreur est passé de 59.25% à 54.50%. Dans le cas des classificateurs LDA et QDA, on a pu réduire la dimension mais avec taux d'erreur supérieur au taux initial.
Nous présentons par la suite les trois matrices de confusion données par les tableaux 3.19, 3.20 et 3.21, calculées pour le classificateur SVM avec extraction d'attributs.

TABLE 3.19 – Matrice d'occurrence exprimée en %

	F4	F9	F11
F4	539	70	161
F9	159	530	369
F11	102	200	270

TABLE 3.20 – Matrice de précision exprimée en %

	F4	F9	F11
F4	67.37	08.75	20.13
F9	19.88	66.25	46.12
F11	12.75	25.00	33.75

TABLE 3.21 – Matrice de fiabilité exprimée en %

	F4	F9	F11
F4	70.00	09.09	20.91
F9	15.03	50.09	34.88
F11	17.83	34.97	25.52

On remarque d'après la matrice d'occurrence que la reconnaissance de la faute F9 s'est améliorée et la confusion avec la faute F11 est diminué. En revanche on note bien que c'est la faute F4 qui a légèrement perdu de sa distinction de la faute F11 mais elle l'a légèrement amélioré par rapport à la faute F9.
Ainsi, sur la matrice de précision, le classificateur a augmenté la précision de sa classification pour la faute F9 et sa fiabilité qui a passé de 30% à 50%.
Dans le tableau 3.22, on a procédé à la comparaison de notre méthode d'extraction FECM avec les deux méthodes d'extraction LDA (supervisée) et ACP (non supervisée).

TABLE 3.22 – Taux d'erreur de classification pour différents méthodes d'extraction

N. attributs	Taux d'erreur LDA(%)			Taux d'erreur ACP(%)			Taux d'erreur FECM(%)		
	SVM	QDA	KNN	SVM	QDA	KNN	SVM	QDA	KNN
2	53.63	43.00	44.92	67.13	66.75	67.79	-	-	-
4	-	-	-	64.71	66.21	66.21	44.20	45.79	54.50
10	-	-	-	64.71	65.00	66.92	49.17	35.50	58.88
21	-	-	-	52.96	48.71	56.29	53.96	43.42	59.25

On a utilisé les trois classificateurs SVM, QDA et KNN pour effectuer la classification. Vu que le LDA ne peut produire que 2 attributs, on retrouve ses résultats à la première ligne seulement. Depuis le tableau, on retrouve que les performances de l'ACP se dégrade avec la diminution de la dimension. Son résultat optimal étant de 53% pour le classificateur SVM et la dimension 21. Les premiers axes de l'ACP ne porte pas l'information nécessaire pour pouvoir classer correctement les trois fautes. Pour notre méthode FECM, on remarque bien que la diminution de la dimension ne l'affecte pas surtout avec SVM. Elle arrive aussi à améliorer les performances du KNN : classificateur non supervisé et linéaire. Quant au QDA, on retrouve que le meilleur résultat est de 35% attribué à la dimension 10.

Influence de la taille des classes sur la classification et l'extraction des attributs

Dans cette section, on a choisit de travailler avec un nombre réduit d'observations dans la base d'apprentissage : 60 observations par classe. Dans la pratique, dans le domaine de diagnostic industriel les bases d'apprentissage ne contiennent pas autant de nombre d'observations. De plus, si nous envisageons d'implémenter notre approche d'extraction sur un procédé de diagnostic en ligne on a besoin de qualifier les performances de notre méthode sur un ensemble d'apprentissage plus petit.
Nous essayons ainsi, d'évaluer notre méthode sous ces contraintes proche de la réalité : nombre réduit d'observations pour faire l'apprentissage.
On remarque que pour le classificateur SVM, notre extraction lui a permit d'améliorer ses performances et passer de 66% à 55% avec la dimension 5. Le classifcateur QDA a obtenu son meilleur résultat 44% avec la dimension 10 comme précédemment. Quant aux classificateurs linéaires LDA et KNN, on remarque une amélioration considérable avec la diminution de la dimension. En effet, le LDA a passé de 63% à 51% et le classificateur non supervisé à passé de 66% à 61%.

TABLE 3.23 – Performance de classification avec 60 obs/classe

Nombre d'attributs	SVM	QDA	KNN	LDA
21	66.21	45.58	63.58	65.29
10	57.75	44.62	60.29	61.04
5	55.17	48.38	61.50	51.75

3.5 Conclusion

A travers ce chapitre, on est arrivé à illustrer notre approche pour l'extraction d'attributs FECM pour les problèmes de classifications. Pour cela, on a choisit différents problèmes de classification et plusieurs bases notamment les bases sonar, pima, iris.. Ces bases présentent différents types de distribution de données avec différents nombre d'attributs et d'observations et un nombre variable de classes (classification binaire et multi-classe).
On a aussi appliqué notre approche sur le problème de reconnaissance de visage. Outre son importance de point de vu industriel, ce problème nous intéresse pour la validation puisque les images ont une grande dimension. Il s'agit en plus d'une classification multi-classe dont il faut plus de précision sur le comportement des at-

tributs. On a travaillé avec deux bases ORL et Yale.
Dans ce chapitre, on a aussi abordé le problème de diagnostic industriel. On a choisit pour cela un procédé multivarié Tenessee easteman process (TEP). Il comporte 52 variables d'entrées et 20 types de fautes différents. On a travaillé sur trois fautes utilisées généralement dans la littérature et qui constituent un problème de classification difficile. On a pu constater les améliorations apportées par notre approche d'extraction pour le diagnostic. Ceci a permis de tester différents types de classificateurs et analyser l'effet apporté par notre extraction à chacun des classificateurs supervisé / non supervisé et linéaire / non linéaire.

Conclusions

Conclusion et perspectives

Conclusion

La classification automatique des données est un concept important qui fait partie du processus d'analyse de données. Elle vise à découvrir la structure intrinsèque d'un ensemble d'objets en formant de regroupements homogènes appelés clusters, et qui partagent des caractéristiques similaires.
La complexité de cette tâche s'est accrue suite à l'augmentation du volume de la masse des données disponibles. Pour cela, il est fondamental de mettre en place des outils de traitement de données qui permettent une meilleure compréhension de la qualité des connaissances disponibles dans ces données.
La réduction de dimension est l'une des approches les plus classiques permettant d'apporter des éléments de réponse à ce problème. Son objectif est de sélectionner ou d'extraire un sous-ensemble optimal de caractéristiques pertinentes selon un critère prédéfini. Cette sélection/extraction permet donc de réduire la dimension de l'espace des entrÃľes et de rendre l'ensemble des données plus représentatif du problème. Ceci permettra d'alléger le problème de classification et d'obtenir de meilleures performances.
Dans cette thèse, nous nous sommes intéressés à la réduction de dimension, dans le cadre de la classification automatique dans un but décisionnel, le diagnostic industriel entre autres. Notre objectif consiste au développement d'une approche d'extraction d'attributs qui peut être appliquée pour différents types de données et différentes distributions.
Elle est basée sur la classification des attributs en différents regroupements ou clusters. Dans chaque cluster, un centre sera extrait à partir de ces attributs et, sera considéré par la suite comme nouveau attribut représentatif. Les nouveaux attributs ainsi construits seront les nouveaux descripteurs des données et qui seront amenés à l'étape d'apprentissage et de classification.
Dans une première étape, nous avons développé une mesure de similarité pour l'al-

gorithme de clustering. Cette mesure de similarité entre les variables est un point essentiel et mérite toute attention. Nous avons remplacé la distance Euclidienne, généralement utilisée dans les algorithmes de clustering, et qui s'est révélée inadaptée par une distance à base de l'analyse de tendance. Deux variables sont similaires si elles se comportent d'une façon similaire le long de la base de données. L'application de notre approche sur différentes bases de données réelles a permis d'extraire les bons attributs pour la classification en comparaison avec l'ACP et l'ACI. De plus ces attributs dépassent en performance les attributs extraits d'une façon linéaire par ACP dans les faibles dimensions. Par ailleurs, pour les plus grandes dimensions, notre approche arrive à dépasser les performances des attributs extraits non linéairement par l'ACI.

Vu l'importance du rôle de chaque centre de cluster dans la représentation des données à classer, nous avons pensé à améliorer la qualité de la transformation appliquée sur chaque cluster. Traditionnellement dans les algorithmes de clustering, les centres de chaque cluster sont obtenus en appliquant la moyenne des données. Cette procédure n'est pas très fiable puisque un cluster peut avoir une forme quelconque avec des données localisées dans les coins. Un critère plus robuste doit être appliqué pour obtenir un centre le plus représentatif possible des attributs dans les clusters.

Nous avons alors exploité l'information mutuelle comme critère de recherche de la transformation optimale à appliquer sur chaque cluster pour obtenir un centre. L'information mutuelle est un critère puissant qui révèle toute forme de relation linéaire ou non linéaire entre les données. Ainsi, la mesure de l'IM est à maximiser entre les attributs dans un cluster et son centre. Un algorithme de descente du gradient stochastique a été mis en place pour la recherche des poids optimaux à appliquer sur les attributs de chaque cluster. La maximisation de l'IM sur différentes bases a permis d'améliorer la qualité de l'extraction. Les attributs extraits dans les faibles dimensions ont dépassé en performances les attributs extraits linéairement par ACP et non linéairement par l'ACI, la plupart de temps.

Dans le dernier chapitre, on a appliqué notre approche d'extraction sur différents benchmarks issus de différents problèmes de classification comme la base de diagnostic de cancer de sein (base breast cancer), diagnostic de diabètes (base Pima), reconnaissance de vin (base Wine), reconnaissance de plante (base Iris), reconnaissance de type de métal (base Sonar), et diagnostic de couche inosphere (base Ionosphere). Ces bases sont diversifiées en nombre d'observations, nombre d'attributs, nombre de classes, et nombre d'observations par classe.

Une deuxième application est la reconnaissance de visage à travers les deux bases Yale et ORL. Ce sont deux problèmes multi-classes avec un nombre énorme d'observations par rapport au nombre d'attributs.
Finalement, on a appliqué notre méthode d'extraction pour le diagnostic industriel perçu comme étant un problème de classification supervisée. On a utilisé le processus Tenessee Easteman Process (TEP), qui se distingue par un grand nombre de défauts et un grand nombre de variables.

Perspectives

Plusieurs améliorations peuvent être apportées aux travaux de cette thèse et qui sont à plusieurs niveaux à savoir : la maîtrise de la sensibilité au bruit et le déséquilibre entre les classes, l'incorporation d'informations supplémentaires pour gérer les problèmes semi-supervisés, l'utilisation d'autres algorithmes de clustering plus robuste tels que le spectral clustering, et enfin l'application de notre approche sur des problèmes totalement non supervisés et dynamiques (spécifiquement les données temporelles).

- La mesure de similarité conçue dans ce travail est basée sur l'analyse de la tendance des observations. La qualité des observations introduites au processus d'extraction définit la qualité de description de chaque attribut. Cette mesure peut devenir sensible au bruit. Dans le cas d'une version supervisée, le problème de déséquilibre de classe s'y ajoute. Une étude de robustesse de notre approche peut être envisagée.
- L'extraction d'attributs d'une manière non supervisée est moins prédisposée au sur-apprentissage. Et la plupart des problèmes réels sont des problèmes non supervisés vu que la labélisation des données est coûteuse. Toutefois, l'introduction d'une forme de supervision est nécessaire afin de traiter les problèmes de classification semi-supervisée. Pour adapter notre approche aux problèmes de classification semi-supervisée, une contrainte doit s'ajouter afin de modéliser les attributs dans chaque classe à part.
- L'application de notre approche d'extraction sur le problème de diagnostic de TEP nous a permis de constater que la réduction du nombre d'observations par classe a réduit les performances des différents classificateurs. Ceci montre la sensibilité de notre approche par rapport au nombre et qualité d'observations. Attaquer le problème de diagnostic industriel en ligne et la classification dynamique nécessite une adaptation de notre approche.

- Appliquer notre approche sur un problème réel totalement non supervisé serait intéressant pour évaluer l'apport effective de notre approche au problème de classification automatique. Toutefois, dans le traitement du TEP, on a dû diversifier les classificateurs pour évaluer l'effet de l'extraction. Nous avons utilisé le KNN avec un seul voisin qui est un classificateur linéaire non supervisé. Cependant, l'évaluation de la classification non supervisée s'est seulement reposé sur le taux d'erreur de classification.
- D'autres méthodes de clustering peuvent être exploitées. Le spectral clustering qui se base sur une matrice d'affinité pour définir les similarités entres les variables peut ainsi être considéré. Une régression peut aussi remplacer l'analyse de tendance pour caractériser chaque attribut par une fonction spécifique.

Annexe A :
Liste des publications

Publications dans des revues internationales

1. S.EL FERCHICHI, S. ZIDI, K.LAABIDI, M. KSOURI and S. MAOUCHE, A New Unsupervised Clustering-based Feature Extraction Method. International Journal of Computer Applications (IJCA), vol.57, No. 6, pages 43-49, 2012.

2. S.EL FERCHICHI, S. ZIDI, K.LAABIDI, M. KSOURI and S. MAOUCHE, Clustering based Mutual Information Maximization for Feature Extraction, soumis à Neural Network World, 2013.

Publications dans des congrès internationaux

1. S.EL FERCHICHI, S. ZIDI, K.LAABIDI, M. KSOURI and S. MAOUCHE. Feature extraction for atmospheric pollution detection (regular paper). In CCCA'11, 2011 international conference on Communications, Computing and Control applications, 3-5 Mars Hammamet, Tunisie, 2011.
2. S.EL FERCHICHI, S. ZIDI, K.LAABIDI, M. KSOURI and S. MAOUCHE (regular paper). A new feature extraction method based on clustering for face recognition. In EANN 2011, 12^{th} conference on Engineering Applications of Neural Networks, 15-18 Septembre Corfou, Grèce. (eds.) IFIP Advances in Information and Communication Technology, 2011, Volume 363/2011, pages 247-253, Springer-Verlag.
3. S.EL FERCHICHI, S. ZIDI, K.LAABIDI, M. KSOURI and S. MAOUCHE. Meanshift Clustering based trend analysis distance for fault diagnosis, accepté

à International Conference on Advances in Computer Science and Engineering (CSE 2013).

Chapitre de livre

1. S.EL FERCHICHI, S. ZIDI, K.LAABIDI, M. KSOURI and S. MAOUCHE. A new feature extraction method based on clustering for face recognition. L.Iliadis and C.Jayne (Eds.), Springer, pages 247-253, Vol. 363, IFIP Advances in Information and Communication Technology, 2011.

Bibliographie

[Aguilar, 2008] Aguilar, J. A. A. (2008). *Méthodes spectrales pour le traitement automatique de documents audio*. PhD thesis, Université Toulouse III- Paul Sabatier. [-]

[Baker and McCallum, 1998] Baker, L. and McCallum, A. (1998). Distributional clustering of words for text classification. In *Proceedings of the 21 st annual international ACM SIGIR conference on Resaerch and development in information retrieval*, pages 96–103. ACM. [-]

[Barbedor, 2006] Barbedor, P. (2006). *Analyse en composantes indépendantes par ondelettes*. PhD thesis, Université Paris 7 Denis Diderot. [-]

[Battiti, 1994] Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5. [-]

[Bekkerman, 2003] Bekkerman, R. (2003). *Distributional clustering of words for text categorization*. PhD thesis, Israel Institute of Technology. [-]

[Bekkerman et al., 2003] Bekkerman, R., El-Yaniv, R., Tishby, N., and Winter, Y. (2003). Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3 :1183–1208. [-]

[Belhumeur et al., 1997] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces : recognition using class specifc linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :711–720. [-]

[Bennani, 2001] Bennani, Y. (2001). Systèmes d'apprentissage connexionnistes : sélection de variables. *Revue d'Intelligence Artiticielle*, 15 :3–4. [-]

[Bisson, 2000] Bisson, G. (2000). *La similarité une notion symbolique/numérique*, volume 2 of *Apprentissage symbolique-numérique*. Cepadues. [-]

[Blake et al.,] Blake, C., Keogh, E., and Merz, C. Uci repository of machine learning databases. `http://www.ics.uci.edu/mlearn/MLRepository.html`, year = 1998. [-]

[Bonet et al., 2006] Bonet, I., Y.Saeys, Abalo, R., Garcia, M., Sanchez, R., and de Peer, Y. V. (2006). Feature extraction using clustering of protein. In *Proceedings of the 11 th Iberoamerican Congress on Pattren Recognition (CIARP)*, LNCS, pages 614–623. Springer. [-]

[Bouchon-Meunier et al., 1996] Bouchon-Meunier, B., Rifqi, M., and Bothorel, S. (1996). Towards general measures of comparaison of objects. *ACM Computing Surveys*, 84 :143–153. [-]

[Boucuelid, 2007] Boucuelid, M. (2007). *Contribution à l'application de la reconnaissance des formes et la théorie des possibilités au diagnostic adaptatif et prédictif des systèmes dynamiques*. PhD thesis, Université de Reims Champagne-Ardenne. [-]

[Bouveyron, 2006] Bouveyron, C. (2006). *Modélisation et classiifcation des données de grande dimension : application à l'analyse d'images*. PhD thesis, Université Joseph Fourier Grenoble 1. [-]

[Bouveyron and Girard, 2009] Bouveyron, C. and Girard, S. (2009). Classification supervisée et non supervisée des données de grande dimensions. *MODULAD*, 1 :81–102. [-]

[Carreira-Perpinan, 1997] Carreira-Perpinan, M. (1997). A review of dimension reduction techniques. Technical report, Technical Report CS-96-09, University of Sheffield. [-]

[Castella et al., 2003] Castella, M., Chevreuil, A., and Pesquet, J. C. (2003). Séparation aveugle d'un mélange convolutif de sources non linéaires par une approche hiérarchique. In *Proceedings of 19éme Colloque GRETSI*. [-]

[Cattell, 1966] Cattell, R. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, 1 :245–276. [-]

[Cellucci et al., 2005] Cellucci, C., Albano, A., and Rapp, P. (2005). Statistical validation of mutual information calculations : comparison of alternative numerical algorithms. *Physical Review E*, 71(6). [-]

[Chapelle et al., 2006] Chapelle, O., Schölkopf, B., and Zein, A. (2006). *Semi-supervised learning*. MIT Press. [-]

[Charbonnier, 2006] Charbonnier, S. (2006). *Surveillance de systèmes continus à l'aide de méthodes n'utilisant pas de modèles formels : Application aux systèmes médicaux*. PhD thesis, Institut National Polytechnique de Grenoble. [-]

[Charbonnier and Gentil, 2007] Charbonnier, S. and Gentil, S. (2007). A trend-based alarm system to improve patient monitoring in intensive care units. *Control Engineering Practice*, 15 :1039–1050. [-]

[Cheung and Stephanopoulos, 1990] Cheung, J. and Stephanopoulos, G. (1990). Representation of process trends. *Computers and Chemical Engineering*, 14 :495–510. [-]

[Chiang et al., 2004] Chiang, L. H., Kotanchek, M. E., and Kordon, A. K. (2004). Fault diagnosis based on fisher discriminant analysis and support vector machines. *Computers and Chemical Engineering*, 28 :1389–1401. [-]

[Cleuziou, 2004] Cleuziou, G. (2004). *Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information*. PhD thesis, Université d'Orléans. [-]

[Colomer et al., 2002] Colomer, J., Meléndez, J., and Gamero, F. I. (2002). Pattern recognition based on episodes and dtw, application to diagnosis of a level control system. In *Proceedings of The 16th international Workshop on Qualitative Reasoning*, pages 37–43. [-]

[Comon, 1994] Comon, P. (1994). Independent component analusis, a new concept ? *Signal Processing*, 36 :287–314. [-]

[Comon et al., 1994] Comon, P., luc Voz, J., and Verleysen, M. (1994). Estimation of performance bounds in supervised classification. In *Proceedings of European Symposium on Artificial Neural Networks (ESANN)*, pages 37–42. [-]

[Cornuéjols and Miclet, 2009] Cornuéjols, A. and Miclet, L. (2009). *Apprentissage artificiel, concepts et algorithmes*. Eyrolles. [-]

[Dash and Liu, 1997] Dash, M. and Liu, H. (1997). Feature selection for classification, in intelligent data analysis. *Intelligent Data Analysis*, 1 :131–156. [-]

[Daviet, 2009] Daviet, H. (2009). *ClassAdd, une procédure de sélection de variables basée sur une troncature k-additive de l'information mutuelle et sur une Classification Ascendante Hiérarchique en pré-traitement.* PhD thesis, Université Nantes. [-]

[Dhillon et al., 2003] Dhillon, I. S., Mallela, S., and Kumar, R. (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3 :1265–1287. [-]

[Donoho, 2000] Donoho, D. L. (2000). *Aide-Memoire. High-Dimensional Data Analysis : The Curses and Blessings of Dimensionality.* Department of Statistics, Stanford University, math challenges of the 21st century edition. [-]

[Evangelista et al., 2006] Evangelista, P. F., Embrechts, M. J., and Szymanski, B. K. (2006). Taming the curse of dimensionality in kernels and novelty detection. In *Applied Soft Computing Technologies : The Challenge of Complexity*, volume 34 of *Advances in Soft Computing*, pages 425–438. Springer Verlag. [-]

[Feraud et al., 1998] Feraud, R., B., M., Clérot, F., Fessant, F., and Lemaire, V. (1998). The orange customer analysis platform. In *Proceedings of the 10th International Conference on Data Mining*, Lecture Notes in Computer Science, pages 584–594. Springer. [-]

[Ferchichi, 2008] Ferchichi, S. (2008). Sélection de variables et de caractéristiques pour une méthode d'apprentissage. Master's thesis, Ecole Nationale d'Ingénieurs de Tunis. [-]

[Ferchichi et al., 2009a] Ferchichi, S. E., Laabidi, K., and Zidi, S. (2009a). Genetic algorithm and tabu search for feature selection. *Studies in Informatics and Control*, 18 :181–187. [-]

[Ferchichi et al., 2011] Ferchichi, S. E., Zidi, S., Laabidi, K., Ksouri, M., and Maouche, S. (2011). A new feature extraction method based on clustering for face recognition. In *Proceedings of The 12th Engineering Applications of Neural Networks (EANN)*, volume 363 of *IFIP Advances in information and communication Technology*, pages 247–253. Springer Berlin Heidelberg. [-]

[Ferchichi et al., 2009b] Ferchichi, S. E., Zidi, S., Laabidi, K., and Maouche, S. (2009b). Feature selection using an svm learning machines. In *Proceedings of the 3rd International Conference on Signals, Circuits and Systems (SCS)*, pages 1–6. IEEE Computer Society. [-]

[Fern and Brodley, 2006] Fern, X. and Brodley, C. (2006). Cluster ensembles for high dimensional clustering : an empirical study. Technical report, Oregon State University. [-]

[FisherIII et al., 1998] FisherIII, J. W., Jos, I., and Principe, J. C. (1998). A methodology for information theoretic feature extraction. In *Proceedings of World Congress on Computational Intelligence*, pages 1712–1716. [-]

[Fodor, 2002] Fodor, I. (2002). A survey of dimension reduction techniques. Technical report, Technical Report, Center for Applied Scientific Computing, Livermore, Canada. [-]

[Grandvalet and Canu, 2002] Grandvalet, Y. and Canu, S. (2002). Adaptive scaling for feature selection in svms. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 553–560. [-]

[Guérif, 2006] Guérif, S. (2006). *Réduction de dimension en apprentissage numérique non supervisée*. PhD thesis, Université Paris 13. [-]

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning*, 3 :1157–1182. [-]

[Guyon and Elisseeff, 2006] Guyon, I. and Elisseeff, A. (2006). *Introduction in Feature Extraction : Foundations, and Applications*. Springer. [-]

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning*. Springer. [-]

[He et al., 2005] He, X., Cai, D., and Niyogi, P. (2005). Laplacian score for feature selection. In *Proceedings of the 17 th International Conference on Pattern Recognition (ICPR)*, pages 335–338. [-]

[HildII et al., 2006] HildII, K. E., Torkkola, K., and Principe, J. C. (2006). Feature extraction using information-theoritic learning. In *Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 29, pages 1385–1392. [-]

[Hunter and McIntosh, 1999] Hunter, J. and McIntosh, N. (1999). Knowledge based event detection in complex time series data. In notes in artificial intelligence, L., editor, *Proceedings of the AIDM*, pages 271–280. Springer-Verlag. [-]

[Ishak, 2007] Ishak, A. B. (2007). *Sélection des variables par la machine à vecteurs supports pour la discrimination binaire et multiclasse en grande dimension*. PhD thesis, Université de la Méditerranée Aix-Marseille II. [-]

[Jain et al., 1999] Jain, A., Murty, M., and Flyn, P. (1999). Data clustering : a review. *ACM Computing Surveys*, 31 :264–323. [-]

[Janusz, 1991] Janusz, M. E. (1991). Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis. *Engineering Applications of Artificial Intelligence*, 4 :329–339. [-]

[Kano et al., 2000] Kano, M., Nagao, K., Hasebe, S., Hashimoto, I., Ohno, H., Strauss, R., and Bakshi, B. (2000). Comparison of statistical process monitoring methods : application to the eastman challenge problem. *Computers and Chemical Engineering*, 24 :175–181. [-]

[Kapur, 1994] Kapur, J. (1994). *Measures of information and their applications*. Wiley, New Delhi, India. [-]

[Kohavi and John, 1997] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 1 :273–324. [-]

[Kärnä, 2007] Kärnä, T. (2007). Functional data dimensionality reduction for machine learning. Master's thesis, Helsinki University of Technology. [-]

[Kulkarni et al., 2005] Kulkarni, A., Jayaraman, V., and Kulkarni, B. (2005). Knowledge incorporated support vector machines to detect faults in tenessee easteman process. *Computer and Chemical Engineering*, 29 :2128–2133. [-]

[Kwak, 2003] Kwak, N. (2003). *Feature selection and extraction based on mutual information for classification*. PhD thesis, School of Electrical and Computer Engineering, Seoul National University, Seoul, Korea. [-]

[Kwak, 2007] Kwak, N. (2007). Feature extraction based on direct calculation of mutual information. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(7) :1213–1231. [-]

[Kwak and Choi, 2002] Kwak, N. and Choi, C. H. (2002). Input feature selection by mutual information based on parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 :1667–1671. [-]

[Langley, 1994] Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall symposium on relevance*, pages 140–144. AAAI Press. [-]

[Lee and Verleysen, 2007] Lee, J. A. and Verleysen, M. (2007). *Non linear dimensionality reduction*. Springer. [-]

[Liu et al., 2008] Liu, X., Tang, J., Liu, J., and Feng, Z. (2008). A semi-supervised relief based feature extraction algorithm. In *Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking Symposia*, pages 3–6. IEEE Computer Society. [-]

[Luxburg, 2007] Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17 :395–416. [-]

[Luxburg et al., 2007] Luxburg, U., Bubeck, S., Jegelka, S., and Kaufman, M. (2007). Consistent minimization of clustering objective functions. In *Proceedings of The 21 st Annual Conference on Neural Information Processing Systems (NIPS)*. [-]

[Lyman and Georgakis, 1995] Lyman, P. and Georgakis, C. (1995). Plant-wide control of the tennessee eastman problem. *Computers and Chemical Engineering*, 19(3) :321–331. [-]

[Makivirta et al., 1998] Makivirta, A., Koski, E., Kari, A., and Sukuvaara, T. (1998). Robust signal to symbol transformation using median filters. In *Proceedings of IFAC workshop on decision support for patient management : measurement, modelling and control*, pages 91–104. [-]

[Mao and Jain, 1996] Mao, J. and Jain, A. (1996). Data clustering : a a self-organizing network for hyperellipsoidal clustering(hec). *IEEE Transactions in Neural Networks*, 7 :16–29. [-]

[Mika et al., 1999] Mika, S., Ratsch, G., Weston, J., Schblkopf, B., and Muller, K. (1999). Fisher discriminant analysis with kernels. In *Proceedings of the IEEE Signal Processing Society Workshop*, pages 41–48. [-]

[Pavlenko and Rosen, 2001] Pavlenko, T. and Rosen, D. V. (2001). Effect of dimensionality on discrimination. *Statistics*, 35 :191–213. [-]

[Payne and Edwards, 1998] Payne, T. R. and Edwards, P. (1998). Implicit feature selection with the value difference metric. In *European Conference on Artificial Intelligence*, pages 450–454. [-]

[Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2 :559–572. [-]

[Pereira et al., 1993] Pereira, F., Tishby, N., and Lee, L. (1993). Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190. [-]

[Pierce, 1980] Pierce, J. R. (1980). *An introduction to information theory : Symbols, signals and noise*. Dover Publications. [-]

[Principe et al., 2000] Principe, J., FisherIII, J., and Xu, D. (2000). *Information theoretic learning*. Unsupervised adaptive filtering. Wiley, New York, NY. [-]

[Rengaswamy et al., 2001] Rengaswamy, R., Hagglund, T., and Venkatasubramanian, V. (2001). A qualitative shape analysis formalism for monitoring control loop performance. *Engineering Applications of Artificial Intelligence*, 14 :23–33. [-]

[Ripley and Hjort, 1995] Ripley, B. D. and Hjort, N. L. (1995). *Pattern Recognition and Neural Networks*. Cambridge University Press. [-]

[Rodriguez et al., 2008] Rodriguez, Y., Baets, B., Garcia, M. M., Morell, C., and Grau, R. (2008). A correlation-based distance function for nearest neighbor classification. In *Proceedings of the 13th Iberoamerican congress on Pattern Recognition (CIARP) : Progress in Pattern Recognition, Image Analysis and Applications*, pages 284–291. Springer-Verlag. [-]

[Saul and Roweis, 2000] Saul, L. K. and Roweis, S. T. (2000). An introduction to locally linear embedding. Technical report, Technical Report. [-]

[Schaffernicht et al., 2010] Schaffernicht, E., Kaltenhaeuser, R., Verma, S. S., and Gross, H.-M. (2010). On estimating mutual information for feature selection. In *Proceedings of the 20th International Conference on Artificial Neural Networks (ICANN)*, LNCS, pages 362–367. Springer. [-]

[Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10 :1299–1319. [-]

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423. [-]

[Silverman, 1986] Silverman, B. (1986). *Density estimation for statistics and data analysis*. Chapman and Hall. [-]

[Slonim and Tishby, 1999] Slonim, N. and Tishby, N. (1999). Agglomerative information bottleneck. pages 617–623. MIT Press. [-]

[Srivastava et al., 2007] Srivastava, S., Gupta, M. R., and Frigyik, B. A. (2007). Bayesian quadratic discriminant analysis. *Journal of Machine Learning Research*, 8 :1277–1305. [-]

[Stanfill and Waltz, 1986] Stanfill, C. and Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, 29 :1213–1228. [-]

[Tahir, 2004] Tahir, M. (2004). Feature selection using tabu search for improving the classification rate prostate needle biopsies. In *Proceedings of the 17 th International Conference on Pattern Recognition (ICPR)*, pages 335–338. [-]

[Taleb and Jutten, 1997] Taleb, A. and Jutten, C. (1997). Séparation de sources dans les mélanges non linéaires. In *Proceedings of 6éme Colloque GRETSI*. [-]

[Torkkola, 2003] Torkkola, K. (2003). Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3 :1415–1438. [-]

[Tzanakou, 2001] Tzanakou, E. M. (2001). Supervised and unsupervised pattern recognition : Feature extraction and computational intelligence. *IEEE Transactions on Neural Networks*, 12(3) :644–647. [-]

[Vapnick, 1996] Vapnick, V. (1996). *The nature of statistical learning theory*. Springer-Verlag. [-]

[Verleysen, 2003] Verleysen, M. (2003). Learning high-dimensional data. In *Proceedings of Limitations and Future Trends in Neural Computation*, pages 141–162. IOS Press. [-]

[Verleysen and François, 2005] Verleysen, M. and François, D. (2005). The curse of dimensionality in data mining and time series prediction. In *Computational Intelligence and Bioinspired Systems, Lecture Notes in Computer Science 3512*, pages 758–770. Springer. [-]

[Verron, 2007] Verron, S. (2007). *Diagnostic et surveillance des processus complexes par réseaux bayésiens*. PhD thesis, Université d'Angers. [-]

[Verron et al., 2008] Verron, S., Tiplica, T., and Kobi, A. (2008). Fault detection and identification with a new feature selection based on mutual information. *Journal of Process Control*, 18 :479–490. [-]

[Wang et al., 2006] Wang, L., Jiang, M., Liao, S., and Lu, Y. (2006). A feature selection method based on concept extraction and som text clustering analysis. *International Journal of Computer Science and Network Security*, 6 :20–28. [-]

[Weston et al., 2000] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2000). Feature selection for svms. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 668–674. [-]

[Wilson and Martinez, 1996] Wilson, D. R. and Martinez, T. R. (1996). Value difference metrics for continuously valued attributes. In *Proceedings of the International Conference on Artificial Intelligence, Expert Systems and Neural Networks (AIE)*, pages 11–4. [-]

[Xiang and Huang, 2006] Xiang, C. and Huang, D. (2006). Feature extraction using recursive cluster-based linear discriminant with application to face recognition. *IEEE transactions on image processing*, 15 :3824–3832. [-]

[Xu et al., 2008] Xu, J., Paiva, A., and Principe, J. (2008). A reproducing kernel hilbert space framework for information-theoretic learning. *IEEE Transactions on Signal Processing*, 56(12) :5891–5902. [-]

[Yang and Moody, 1999] Yang, H. H. and Moody, J. (1999). Feature selection based on joint mutual information. In *Proceedings of International ICSC Symposium on Advances in Intelligent Data Analysis*, pages 22–25. [-]

[Zhao et al., 2008] Zhao, J., Lu, K., and He, X. (2008). Localy sensitive semi-supervised feature selection. *Neurocomputing*, 71 :1842–1849. [-]

[Zhu, 2006] Zhu, X. (2006). Semi-supervised learning literature survey. Technical report, Computer Sciences Technical Report 1530, University of Wisconsin Madison. [-]

Résumé

Résumé

Les progrès scientifiques réalisés ces dernières années ont produit des bases de données de plus en plus grandes et complexes. Ceci amène certains classificateurs à générer des règles de classification basées sur des attributs non pertinents, et dégrader ainsi la qualité de classification et la capacité de généralisation. Dans ce contexte, nous proposons une nouvelle méthode pour l'extraction d'attributs afin d'améliorer la qualité de la classification. Notre méthode consiste à effectuer une classification non supervisée des attributs afin de retrouver les groupements d'attributs similaires. Une nouvelle mesure de similarité à base d'analyse de tendance est alors conçue afin de retrouver les attributs similaires dans leur comportement. En effet, notre méthode cherche à réduire l'information redondante tout en identifiant les tendances similaires dans les vecteurs attributs tout au long de la base de données. Suite à la formation des clusters, une transformation linéaire sera appliquée sur les attributs dans chaque groupement pour obtenir un représentant unique. Afin de retrouver un centre optimal, nous proposons de maximiser l'Information Mutuelle (IM) comme mesure de dépendance entre les groupements d'attributs et leur centre recherché. Des expériences réalisées sur des bases de données réelles et artificielles montrent que notre méthode atteint de bonnes performances de classification en comparaison avec d'autres méthodes d'extraction d'attributs. Notre méthode a été également appliquée sur le diagnostic industriel d'un procédé chimique complexe Tennessee Eastman Process (TEP). **Mots clés :** Extraction d'attributs, sélection d'attributs, clustering, analyse de tendance, reconnaissance de visage, diagnostic industriel.

Absract

Scientific advances in recent years have produced databases increasingly large and complex. This brings some classifiers to generate classification rules based on irrelevant features, and thus degrade the quality of classification and generalization ability. In this context, we propose a new method for extracting features to improve the quality of classification. Our method performs a clustering of features to find groups of similar features. A new similarity measure based on trend analysis is then designed to find similarity between features in their behavior. Indeed, our method aims to reduce redundant information while identifying similar trends in features vectors throughout the database. Following the construction of clusters, a linear transformation is applied on each group to obtain a single representative. To find an optimal center, we propose to maximize the Mutual Information (MI) as a measure of dependency between groups of features and the desired center. Experiments on real and synthetic data show that our method achieved good classification performance in comparison with other methods of extracting features. Our method has also been applied to the industrial diagnosis of a complex chemical process Tennessee Eastman Process (TEP). **Keywords :** Feature extraction, feature selection, clustering, trend analysis, face

recongnition, industrial diagnosis.