Numéro d'ordre :40553

UNIVERSITÉ DE LILLE 1 SCIENCES ET TECHNOLOGIES

U.F.R d'Informatique, Électronique, Électrotechnique et Automatique

THÈSE

présentée pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LILLE 1

Discipline : Automatique et Informatique Industrielle

par

Mariam KALAKECH

le 8 juillet 2011

Titre :

Sélection semi-supervisée d'attributs : Application à la classification de textures couleur

JURY

Laurent HEUTTE	Professeur à l'Université de Rouen	*Rapporteur*
Gérard GOVAERT	Professeur à l'UTC Compiègne	*Rapporteur*
Carl FRELICOT	Professeur à l'Université de La Rochelle	*Président de Jury*
Alain TREMEAU	Professeur à l'Université de Saint-Etienne	*Examinateur*
Gilles VERSTRAETE	Directeur de R & D chez Castorama	*Invité*
Ludovic MACAIRE	Professeur à l'Université de Lille 1	*Co-directeur*
Denis HAMAD	Professeur à l'ULCO Calais	*Co-directeur*
Philippe BIELA	Enseignant-Chercheur à HEI	*Co-encadrant*

Table des matières

Notations

Notations

$\mathscr{X}$	Ensemble de n objets caractérisés par d attributs
n	Nombre de données
d	Nombre d'attributs
x_i	Donnée particulière d'indice i
f_r	Attribut particulier d'indice r
x_{ir}	Réalisation particulière de l'attribut f_r sur la donnée x_i
δ_{ij}	Fonction de distance entre deux données x_i et x_j
Δ	Matrice de distance (n$\times$n)
w_{ij}	Fonction de similarité entre deux données x_i et x_j
W	Matrice de similarité (n$\times$n)
σ	Paramètre de dispersion de la fonction gaussienne de similarité
$\mathscr{G}$	Graphe de similarité
$\mathscr{N}$	Ensemble des noeuds du graphe
$\mathscr{E}$	Ensemble d'arcs entre les différents noeuds du graphe
s_i	Noeud correspondant au point x_i
d_i	Degré correspondant au noeud s_i
D	Matrice des degrés
L	Matrice Laplacienne
y_i	Label de classe de la donnée x_i
Y	Vecteur des labels de classes des données
c	Nombre de classes de données
w	Classe de données
n_w	Nombre de données de la classe w

$\mu_{\omega r}$	Moyene de la classe *w* sur l'attribut f_r
$\sigma^2_{\omega r}$	Variance de la classe *w* sur l'attribut f_r
$\mathscr{X}_l$	Sous-ensemble de *l* données labellisées
$\mathscr{X}_u$	Sous-ensemble de *u* données non labellisées
$\mathscr{M}$	Sous-ensemble de contraintes must-link
$\mathscr{C}$	Sous-ensemble de contraintes cannot-link
$\mathscr{S}$	Sous-ensemble de contraintes
$\lvert\mathscr{M}\rvert$	cardinal de $\mathscr{M}$
$\lvert\mathscr{C}\rvert$	cardinal de $\mathscr{C}$
$G_{\mathscr{M}}$	Graphe des must-link
$G^{\mathscr{C}}$	Graphe des cannot-link
$W^{\mathscr{M}}$	Matrice des must-link
$W^{\mathscr{C}}$	Matrice des cannot-link
SV_r	Score de la Variance de l'attribut f_r
μ_r	Moyenne de l'ensemble des données sur l'attribut considéré f_r
$var(f_r)$	Score de la Variance sous-jacente à un graphe de l'attribut f_r
$\overline{f}_r$	Moyenne pondérée des données projetées sur l'attribut f_r
SL_r	Score Laplacien de l'attribut f_r
SF_r	Score de Fisher de l'attribut f_r
SM_r	Score Laplacien couplé à l'information mutuelle de l'attribut f_r
SC^1_r	Scores de contraintes 1 de Zhang de l'attribut f_r
SC^2_r	Scores de contraintes 2 de Zhang de l'attribut f_r
SC^3_r	Scores de contraintes de Zhao de l'attribut f_r
SC^4_r	Score de sélection semi-supervisé avec contraintes de l'attribut f_r
R	Matrice des rangs
K	Coefficient de Kendal
T	Total des rangs

Introduction générale

.1 Contexte

Dans les différents domaines des sciences de l'ingénieur, le développement technologique et le besoin de superviser des systèmes de plus en plus complexes nécessitent l'analyse de bases de données de taille importante (signaux, images, documents, ...).

Toutefois, si dans cette accumulation de données, on est sûr d'avoir une information complète et utile, celle-ci risque d'être "noyée" dans la masse. Ceci pose les problèmes de la structuration des données et de l'extraction des connaissances.

En effet, les bases de données sont en général définies par des tableaux à deux dimensions correspondant aux données et aux attributs caractérisant ces données. Ces deux dimensions peuvent prendre des valeurs très élevées, ce qui peut poser un problème lors du stockage, de l'exploration et de l'analyse de ces données.

Pour cela, il est important de mettre en place des outils de traitement des données permettant l'extraction des connaissances sous-jacentes. L'extraction des connaissances s'effectue selon deux directions, la catégorisation des données (par regroupement en classes) et/ou la réduction de la dimension de l'espace de représentation de ces données (par sélection ou extraction d'attributs).

La classification vise à découvrir la structure intrinsèque d'un ensemble de données en formant des groupements qui partagent des caractéristiques similaires. La réduction de la dimension se pose comme une étape primordiale dans le processus de prétraitement des données (filtrage, nettoyage, élimination des points aberrants, etc.). En effet, pour des données appartenant à un

espace de grande dimension, certains attributs n'apportent aucune information voire expriment du bruit, d'autres sont redondants ou corrélés. Ceci rend les algorithmes de décision complexes, inefficaces, moins généralisables et d'interprétation délicate.

Les méthodes de réduction de la dimension de l'espace de représentation peuvent être divisées en méthodes d'extraction d'attributs et méthodes de sélection d'attributs. L'extraction d'attributs transforme l'espace d'attributs de départ en un nouvel espace formé de la combinaison linéaire ou non linéaire des attributs initiaux. La sélection d'attributs choisit les attributs les plus pertinents selon un critère donné. Les données sont alors analysées après projection dans un espace de représentation composé des attributs les plus pertinents. Toutefois, l'interprétation des attributs extraits est plus délicate que l'interprétation des attributs sélectionnés.

Le point clé de la sélection d'attributs est la définition d'un score mesurant la pertinence de chacun des attributs. Cette sélection s'appuie sur la connaissance explicite et implicite sur les données. Quand on ne dispose d'aucune information a priori sur le regroupement des données en classes, le contexte d'apprentissage est dit non supervisé. La pertinence d'un attribut est alors mesurée en évaluant ses capacités à préserver la structure des données [DB04].

Pour de nombreuses applications, on dispose des informations a priori sur la répartition des données en classes. Ainsi, pour ces données, les labels des classes ont été fournis. Dans ce cas, la sélection supervisée consiste à mesurer la corrélation entre l'attribut et les labels des classes des données [YL04].

Les algorithmes de sélection supervisée nécessitent de définir des labels de toutes les données. Par conséquent, la procédure de labellisation réalisée par un expert humain peut s'avérer fastidieuse et couteuse en temps de travail. C'est pour cette raison que, pour des applications réelles, on est généralement en présence de bases de données formées de nombreuses données non labellisées et de peu de données labellisées. Ce contexte d'apprentissage est dit semi-supervisé car l'analyste exploite à la fois les données non labellisées et les quelques données labellisées.

Il est aussi possible de modéliser la connaissance a priori de l'expert grâce à des contraintes por-

tant sur des couples de données ([WCR+01]). Il s'agit de mentionner pour quelques données prises deux à deux si elles sont similaires et doivent alors être regroupées ensemble, ce sont des contraintes "must-link", ou si elles sont dissimilaires et donc ne doivent pas être regroupées ensemble, ce sont des contraintes "cannot-link".

La définition de contraintes demande moins de connaissance et d'efforts de la part de l'expert que la définition des labels de données qui nécessitent d'avoir des informations précises sur les classes d'appartenance des données,

Les scores qui évaluent les attributs grâce à ces contraintes négligent l'information apportée par les données non contraintes [ZCZ08] [ZLH08]. Par ailleurs, ces scores dépendent fortement des contraintes fournies par l'expert. Un changement de ces contraintes provoque un changement des scores des différents attributs et donc de leur classement. Ceci nous améne alors à introduire un nouveau score de sélection semi-supervisée avec contraintes, qui prend en considération à la fois les propriétés locales des données ainsi que les contraintes fournies par l'expert.

Afin de mesurer la sensibilité de chacun des scores de contraintes vis à vis du choix des contraintes, nous aurions pu estimer la dispersion des taux de bonne classification obtenus par chacun de ces scores avec différents sous-ensembles de contraintes. Cependant, cette évaluation est très dépendante du classifieur utilisé. Nous proposons alors d'examiner la dispersion des rangs des attributs fournis par les divers scores en utilisant le coefficient de Kendall [Grz06]. Cette comparaison des différents scores de sélection grâce au coefficient de Kendall est une nouvelle approche basée uniquement sur les rangs des attributs fournis par chacun des scores sans considérer les performances de classification.

Les travaux antérieurs évaluent les performances des différents scores en considérant les taux de bonne classification obtenus par un classifieur supervisé opérant dans l'espace des attributs sélectionnés par chacun de ces scores. Cette méthode d'évaluation des performances des scores ne prend pas en considération le contexte d'apprentissage. Nous proposons alors une méthode d'évaluation semi-supervisée de façon à ce que la sélection d'attributs et la classification de

données soient réalisées dans le même contexte.

Afin de comparer les performances de différents scores proposés dans la littérature, des expérimentations sont dans un premier temps menées avec des bases de données de référence. Nous avons ensuite appliqué notre approche à une problématique scientifique concrète, à savoir la classification d'images couleur présentant des textures. La description du contenu des images se fait par le biais d'attributs qui doivent permettre de caractériser les classes en présence. Dans le cas de textures couleur, il est opportun d'utiliser des attributs de textures qui sont non seulement capables de caractériser la distribution des couleurs mais qui tiennent compte également de l'arrangement spatial de ces couleurs au sein des images. Alice Porebski a montré qu'il est intéressant de calculer des attributs de textures à partir d'images codées dans différents espaces couleur [Por09]. Comme le nombre d'attributs est si élevé qu'il peut pénaliser la qualité de discrimination, il est nécessaire de procéder à une réduction de la dimension de l'espace d'attributs afin de ne conserver que les plus pertinents.

.2 Organisation du manuscrit

Le premier chapitre de ce mémoire s'intéresse à la représentation des données et de la connaissance à priori sur la structure de ces données. Ainsi, nous détaillons les données et les attributs qui les caractérisent. Ensuite, nous nous intéressons aux mesures de similarité entre données basées dans la plupart des cas sur la notion de distance. Cette relation de similarité entre données est représentée sous la forme d'un graphe de similarité. Enfin, nous représentons aussi sous forme de graphes la formalisation des connaissances a priori apportées par l'expert et qui définissent le contexte d'apprentissage. Nous nous intéressons au contexte semi-supervisé où nous disposons de connaissances incomplètes.

Le nombre élevé d'attributs nécessite une étape de réduction de la dimension qui sera présentée dans le second chapitre. Nous nous sommes intéressée aux méthodes de sélection d'attributs et plus particulièrement aux scores de classement des attributs à base de graphes.

Nous commençons alors par détailler les différentes étapes d'une procédure de sélection. Ensuite, nous exposons les divers scores de classement d'attributs utilisés dans un contexte non-supervisé, supervisé ou semi-supervisé.
Quand la connaissance a priori est formalisée grâce à des contraintes reliant les données prises deux à deux, les attributs sont évalués grâce à des scores de contraintes. Comme ces scores négligent l'information apportée par les données non-contraintes, nous proposons un nouveau score pour la sélection semi-supervisée avec contraintes. Ce score prend en compte à la fois l'ensemble des données et les contraintes disponibles.

Dans le troisième chapitre, nous abordons un problème important de la sélection d'attributs par les divers scores de contraintes, à savoir, la dépendance des attributs sélectionnés vis à vis des contraintes. Cette dépendance est tout d'abord illustrée par un exemple, puis mesurée à l'aide du coefficient de Kendall basé sur la matrice des rangs des attributs. Des résultats expérimentaux sur des bases de données réelles montrent que notre score de contraintes est moins sensible au jeu de contraintes que les scores existants.

Dans le quatrième chapitre, nous appliquons la procédure d'évaluation classique sur les bases de données utilisées dans le troisième chapitre afin de comparer les performances des différents scores considérés. Les résultats obtenus montrent que les attributs sélectionnés par notre score fournissent des taux de bonne classification comparables à ceux obtenus par les attributs sélectionnés par les scores existants. Cette procédure d'évaluation classique utilise un classifieur supervisé des données tests. Ainsi, les données d'apprentissage utilisées comme prototypes sont labellisées. Cependant, ces labels n'ont pas été exploités par les scores de sélection. En effet, ces scores utilisent uniquement un nombre restreint de contraintes entre données et/ou l'ensemble de données non contraintes. Par conséquent, les données test sont classifiées dans un contexte supervisé, alors que les attributs ont été sélectionnés dans un contexte semi-supervisé. Cela nous a amené à proposer une nouvelle procédure d'évaluation qui garantit que la sélection et la classification opèrent dans le même contexte d'apprentissage.

Le cinquiéme chapitre est une application de l'évaluation et la comparaison des scores pour la sélection d'attributs de textures couleur. En effet, parmi les nombreux attributs de texture pouvant être extraits des images couleur, il est nécessaire de sélectionner les plus pertinents afin d'améliorer la qualité de classification. Nous utilisons les attributs d'Haralick extraits des matrices de co-occurences pour caractériser les images de texture codées dans différents espaces couleur. Nous montrons des résultats obtenus sur les divers bases de texture de référence avec lesquelles nous avons mené nos expériences.

.3 Contributions de la thèse

Les contributions de la thèse se situent au niveau des points suivants :

- Etat de l'art sur les scores de classement d'attributs utilisés dans un contexte non-supervisé, supervisé ou semi-supervisé. Ainsi, nous présentons les scores basés sur la représentation des données par graphes de similarité. Ces scores sont illustrés avec des exemples pédagogiques.
- Proposition d'un nouveau score de sélection d'attributs semi-supervisé avec contraintes. Ce score utilise à la fois l'ensemble des données et le sous-ensemble de contraintes must-link et cannot-link disponibles. Ceci permet d'intégrer les propriétés locales des données en plus de l'information a priori fournie par l'expert.
- Nouvelle comparaison des différents scores utilisant un ensemble de contraintes basée sur le coefficient de Kendal. A notre connaissance, c'est le premier travail de ce genre qui compare les scores de contraintes en se basant uniquement sur les rangs des attributs fournis par chacun des scores.
- Proposition d'une nouvelle méthode d'évaluation basée sur le taux de bonne classification dans un contexte non-supervisé ou semi-supervisé. Cette méthode d'évaluation de la performance des scores consiste à réaliser la sélection d'attributs ainsi que la classification dans le même contexte d'apprentissage.

– Application des scores de contraintes à la sélection d'attributs de textures couleur. L'expert construit des contraintes must-link et cannot-link entre les différentes images de textures. Ces contraintes sont ensuite utilisées pour sélectionner les attributs de textures les plus pertinents pour la classification de ces images.

Chapitre 1

Représentation des données et de la connaissance a priori

1.1 Introduction

Dans les domaines de la reconnaissance des formes, l'apprentissage des machines et la fouille des données, on dispose de plus en plus de bases de données de très grandes dimensions (images, signaux, documents, etc.) qu'on cherche à analyser dans l'objectif d'extraire des connaissances utiles. Cette analyse passe par le regroupement en classes des données qui respectent des propriétés similaires.

Lorsque la seule connaissance disponible est la base de données elle-même, le contexte de décision, à savoir l'assignation de chaque donnée à une classe, est dit non-supervisé. En plus des données, il peut être utile de disposer de connaissances a priori du domaine (nombre de classes, labels des classes, similarité entre données, ...). L'introduction de ces connaissances dans le processus de décision ne peut qu'améliorer les performances des algorithmes de décision.

Le contexte de décision est dit supervisé lorsqu'en plus des données, une connaissance complète du domaine sous forme de labels de classes de toutes les données est disponible. Entre ces deux contextes, le contexte semi-supervisé correspond à des situations courantes où on est en présence de connaissances partielles sous forme de labels de classes de quelques données, contre un grand nombre de données non labellisées.

Ce chapitre est alors consacré à la représentation des données et de la connaissance a priori apportée par l'utilisateur. Nous définissons tout d'abord les données ainsi que les attributs qui les caractérisent (cf. section 1.2). Puis, nous détaillons les mesures de comparaison entre paires de données en évaluant notamment leur similarité (cf. section 1.3). Ensuite, nous exposons la représentation de ces données et de cette notion de similarité sous forme d'un graphe, ainsi que toutes les mesures caractéristiques et les notions qui peuvent en découler (cf. section 1.4). Enfin, selon les connaissances a priori dont nous disposons sur ces données, nous distinguons trois contextes d'analyse à savoir, le contexte supervisé, non supervisé ou semi-supervisé (cf. section 1.5).

1.2 Données et attributs

Nous disposons d'un ensemble $\mathscr{X}$ de n objets, donnant naissance à n observations ou données. Chaque objet i est caractérisé par d attributs $\{f_1, ..., f_r, ..., f_d\}$. La réalisation particulière de l'attribut f_r associée à la donnée x_i est désignée par x_{ir}. L'ensemble des réalisations des attributs sur les données définit une matrice X de n lignes et d colonnes appelée matrice de données, représentée comme suit :

$$X = \begin{bmatrix} x_{11} & \dots & x_{1r} & \dots & x_{1d} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{ir} & \dots & x_{id} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nr} & \dots & x_{nd} \end{bmatrix} \tag{1.1}$$

Chacune des lignes $i = 1, ..., n$ de la matrice, représente un objet et définit un vecteur appelé donnée dans $\mathbb{R}^d$ et noté $x_i = (x_{i1}, ..., x_{ir}, ..., x_{id})^T$. Chacune des colonnes $r = 1, ..., d$, correspondant aux réalisations d'un attribut sur l'ensemble des objets, définit un vecteur dans $\mathbb{R}^n$, noté $f_r = (x_{1r}, ..., x_{ir}, ..., x_{nr})^T$ et appelé vecteur attribut des objets.

D'une manière générale, les valeurs des attributs peuvent être qualitatives ou quantitatives. Dans le cadre de nos travaux, nous nous intéressons exclusivement aux attributs quantitatifs.
Pour se placer selon un point de vue géométrique, on représente chaque donnée par un point dans un espace $\mathbb{R}^d$. L'ensemble des données définit des nuages de points dans cet espace appelé espace d'attributs ou espace d'entrée.

1.3 Distance et Similarité

L'idée intuitive de l'analyse de données s'appuie sur le principe : " ceux qui se ressemblent s'assemblent ". En faisant une analogie entre similarité des données et proximité des points dans l'espace d'attributs, on peut chercher dans la structure des données telles qu'elles se présentent, des groupements naturels selon l'idée : il est très probable que, dans l'espace d'attributs, des points proches représentent des données d'un même groupe et que des points lointains représentent des données qui appartiennent à des groupes différents [Bis00].
Il est alors nécessaire d'évaluer les ressemblances ou les dissemblances qui existent au sein de ces données [Hus10]. Le terme de fonction de similarité ou plus simplement celui de similarité est alors utilisé.
La similarité a pour objet de quantifier la ressemblance entre deux données. En faisant l'analogie avec la proximité, elle est basée dans la plupart des cas sur la notion mathématique de distance. En effet, il est admis que deux points séparés, dans l'espace des attributs, par une grande distance correspondent à deux données non similaires, tandis que deux points proches (au sens de cette distance) correspondent à deux données qui sont similaires. Pour cette raison, nous allons tout d'abord introduire la notion de distance pour ensuite définir la fonction de similarité.

1.3.1 Distance

La fonction de distance utilisée pour définir la distance δ_{ij} entre les points représentant les données (x_i, x_j) est une application de $\mathbb{R}^d \times \mathbb{R}^d$ dans $\mathbb{R}^+$. Elle doit respecter les propriétés suivantes [Los98] :

- Non négativité : $\delta_{ij} \geq 0$
- Symétrie : $\delta_{ij} = \delta_{ji}$
- Séparation : $\delta_{ij} = 0 \Rightarrow i = j$
- Minimalité : $\delta_{ii} = 0$
- Inégalité triangulaire : $\delta_{ij} \leq \delta_{ik} + \delta_{kj}$

Plusieurs fonctions de distance ont été définies dans la littérature. Ces fonctions ont une valeur proche de 0 pour un couple de points proches dans l'espace des attributs et une valeur qui tend vers $+\infty$ pour un couple de points éloignés dans l'espace des attributs.
Les distances les plus courantes sont :

- **Distance Euclidienne**

  La distance Euclidienne, qui est la distance la plus utilisée, est définie comme suit :

$$\delta_{ij} = [\sum_{r=1}^{d} (x_{ir} - x_{jr})^2]^{1/2} \tag{1.2}$$

- **Distance de Manhattan**

  La distance de Manhattan est définie par :

$$\delta_{ij} = \sum_{r=1}^{d} \left| x_{ir} - x_{jr} \right| \tag{1.3}$$

- **Distance de Minkowski**

  La distance de Minkowski est une généralisation de la distance Euclidienne et de la distance de Manhattan. Elle est définie par :

$$\delta_{ij} = [\sum_{r=1}^{d} (x_{ir} - x_{jr})^q]^{1/q} \tag{1.4}$$

où q est un entier positif non nul.

Une forme plus générale est la distance pondérée :

$$\delta_{ij} = [\sum_{r=1}^{d} a_r (x_{ir} - x_{jr})^q]^{1/q} \tag{1.5}$$

a_r étant un coefficient de pondération associé à chaque attribut.

A partir de la matrice des données X, on construit la matrice de distances $\Delta(n \times n)$ de terme général δ_{ij} caractérisant la distance séparant chaque paire de points (x_i, x_j) :

$$\Delta = \begin{bmatrix} 0 & \dots & \delta_{1i} & \dots & \delta_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ \delta_{i1} & \dots & 0 & \dots & \delta_{in} \\ \dots & \dots & \dots & \dots & \dots \\ \delta_{n1} & \dots & \delta_{ni} & \dots & 0 \end{bmatrix} \tag{1.6}$$

1.3.2 Similarité

La similarité notée w_{ij} exprime la ressemblance entre les données (x_i, x_j). C'est une application de $\mathbb{R}^d \times \mathbb{R}^d$ dans $[0,1]$ telle que :

- Symétrie : $w_{ij} = w_{ji}$,
- Normalisation : $w_{ij} \in [0,1]$,
- $w_{ii} \geq w_{ij}$.

Une similarité proche de 1 indique que les données sont similaires, tandis qu'une valeur proche de 0 indiquent qu'elles sont différentes.

Les fonctions de similarité peuvent être exprimées sous des formes multiples (cosinus, coefficient de corrélation de Pearson, Gaussienne, voire floue....) [MRB96].

Les fonctions de similarité les plus courantes sont :

– **La fonction cosinus**

La fonction cosinus est surtout utilisée dans l'analyse des documents [SB88]. Elle est définie comme suit :

$$w_{ij} = \left|\cos(x_i,x_j)\right| = \frac{\left|x_i^T x_j\right|}{\|x_i\| \left\|x_j\right\|} \tag{1.7}$$

Deux données sont d'autant plus similaires que leurs points associés sont placés sur une même droite passant par l'origine de l'espace d'attributs. Cette fonction de similarité est donc sensible à la direction des données projetées dans l'espace d'attributs. Son principal inconvénient est l'impossibilité de différencier des données qui ont des formes ou des directions similaires et qui sont très éloignées les unes des autres. Cette mesure n'est pas couramment utilisée en analyse de données où les différences entre données sont plus liées à leur amplitude qu'à leur direction dans l'espace d'attributs.

– **La fonction Gaussienne basée sur la distance Euclidienne**

En général, la fonction de similarité doit prendre en compte les relations de voisinage entre les données. C'est pour cette raison que la fonction Gaussienne basée sur la distance Euclidienne entre points est souvent utilisée. Elle est définie par :

$$w_{ij} = \exp(-\frac{1}{2\sigma^2}\delta_{ij}^2) \tag{1.8}$$

où δ_{ij} est la distance Euclidienne entre les points associés aux données x_i et x_j définie dans l'équation (1.2).

Le paramètre de dispersion σ doit être choisi de telle sorte qu'il soit adapté à la dispersion locale des données disponibles [MJ01]. En effet, quand la distance Euclidienne séparant x_i et x_j est inférieure à $\sqrt{2}\sigma$, le terme au sein de l'exponentielle est inférieur à -1. La

mesure de similarité entre ces deux points s'approche alors de la valeur 1. Par contre, quand la distance séparant x_i et x_j est nettement supérieure à $\sqrt{2}\sigma$, le terme au sein de l'exponentielle est supérieur à -1. La mesure de similarité est alors proche de 0.

Cette fonction de similarité prend ses valeurs dans l'intervalle continu [0,1]. La valeur 0 signifie une similarité nulle entre données (x_i, x_j) associées à des points éloignés dans l'espace des attributs (δ_{ij} tend vers $+\infty$), tandis que la valeur 1 correspond à une grande similarité entre données associées à des points proches dans l'espace des attributs (δ_{ij}=0). Plus la distance séparant deux points dans l'espace d'attributs est grande, plus la similarité entre les données associées est petite.

A partir de la matrice de distances Δ, on construit alors la matrice de similarité $W(n \times n)$ de terme général w_{ij} caractérisant la similarité entre chaque paire de données (x_i, x_j) à partir de leurs représentations dans l'espace d'attributs :

$$W = \begin{bmatrix} 1 & \dots & w_{1i} & \dots & w_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{i1} & \dots & 1 & \dots & w_{in} \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & \dots & w_{ni} & \dots & 1 \end{bmatrix} \tag{1.9}$$

Voici un exemple simple qui illustre les notions de données, attributs ainsi que le calcul de distance et de similarité entre ces données.

Exemple : Soit un ensemble de 4 données $\mathscr{X} = \{x_1, x_2, x_3, x_4\}$, caractérisées par 2 attributs f_1 et f_2 ($n = 4$, $d = 2$). Ces données sont définies par la matrice X comme suit :

Matrice de données :

$$X=\begin{bmatrix} 4 & 4 \\ -0.5 & -0.5 \\ 4 & 2.5 \\ -0.5 & 0.5 \end{bmatrix} \tag{1.10}$$

La figure 1.1 montre la représentation de ces données dans l'espace $\mathbb{R}^2$.

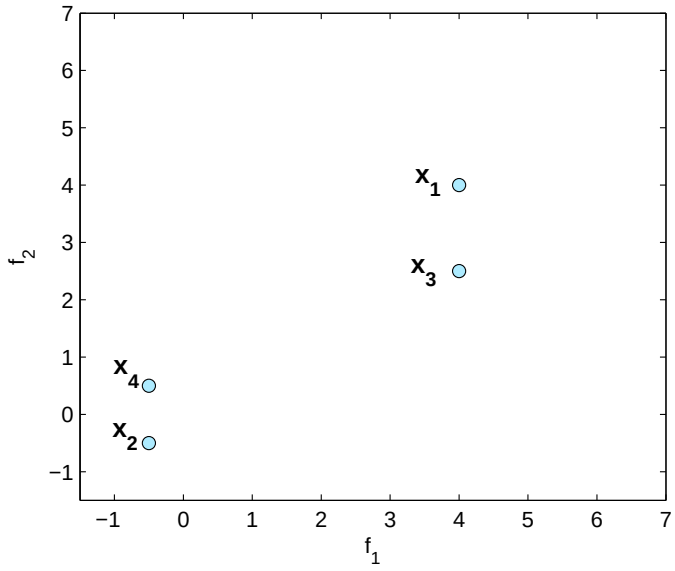


Figure 1.1 : Représentation des données dans l'espace $\mathbb{R}^2$.

Matrice de distances :

La matrice de distance Δ ($n \times n$) calculée entre ces données dans l'espace $\mathbb{R}^2$ en utilisant la distance Euclidienne de l'équation (1.2) est :

$$\Delta = \begin{bmatrix} 0 & 6.364 & 1.5 & 5.7009 \\ 6.364 & 0 & 5.4083 & 1 \\ 1.5 & 5.4083 & 0 & 4.9244 \\ 5.7009 & 1 & 4.9244 & 0 \end{bmatrix} \tag{1.11}$$

Nous pouvons ainsi remarquer que les points x_2 et x_4 les plus proches dans l'espace d'attributs sont séparés par la distance minimale ($\delta_{24} = \delta_{42} = 1$), tandis que les points x_1 et x_2 les plus éloignés dans l'espace d'attributs sont séparés par la distance maximale ($\delta_{12} = \delta_{21} = 6.364$).

Pour illustrer l'influence du choix de la fonction de similarité, nous allons calculer la similarité de deux façons différentes : tout d'abord, en utilisant la fonction cosinus, puis en utilisant la fonction Gaussienne basée sur la distance Euclidienne.

Matrice de similarité cosinus :

En utilisant la fonction cosinus de l'équation (1.7), la matrice de similarité W sera alors :

$$W = \begin{bmatrix} 1 & 1 & 0.9744 & 0 \\ 1 & 1 & 0.9744 & 0 \\ 0.9744 & 0.9744 & 1 & 0.2249 \\ 0 & 0 & 0.2249 & 1 \end{bmatrix} \tag{1.12}$$

Nous pouvons remarquer que les points x_1 et x_2 séparés par la distance la plus grande dans l'espace d'attributs ($\delta_{12} = \delta_{21} = 6.364$) correspondent à la paire de données la plus similaire ($w_{12} = w_{21} = 1$), puisque ces deux points appartiennent à une même droite passant par l'origine. L'angle x_1Ox_2 étant égal à 0 degré, son cosinus est donc égal à 1. Par contre, les points x_2 et x_4 séparés par la distance la plus faible dans l'espace d'attributs ($\delta_{24} = \delta_{42} = 1$) correspondent à la paire de données la moins similaire ($w_{24} = w_{42} = 0$).

Cet exemple illustre que la fonction de similarité basée sur le cosinus n'est pas sensible aux distances séparant les points dans l'espace des attributs.

Matrice de similarité Gaussienne :

En utilisant la fonction de similarité Gaussienne de l'équation (1.8) et en fixant σ à 2, la matrice de similarité W sera alors :

$$W = \begin{bmatrix} 1 & 0.0063 & 0.7548 & 0.0172 \\ 0.0063 & 1 & 0.0258 & 0.8825 \\ 0.7548 & 0.0258 & 1 & 0.0483 \\ 0.0172 & 0.8825 & 0.0483 & 1 \end{bmatrix} \tag{1.13}$$

Nous pouvons remarquer que les points x_1 et x_2 séparés par la distance la plus grande dans l'espace d'attributs correspondent à la paire de données la moins similaire ($w_{12} = w_{21} = 0.0063$), tandis que les points x_2 et x_4 séparés par la distance la plus faible dans l'espace d'attributs correspondent à la paire de données la plus similaire ($w_{24} = w_{42} = 0.8825$).

Cet exemple met en évidence que la similarité basée sur la fonction Gaussienne dépend de la distance séparant les points associés aux données dans l'espace d'attributs.

La similarité basée sur la distance entre les points dans l'espace d'attributs semble donc être mieux adaptée au regroupement des données. C'est pour cette raison que la fonction de similarité Gaussienne basée sur la distance Euclidienne a été retenue le long de ce manuscrit.

1.4 Représentation des données par graphe de similarité

Le concept de graphe peut être utilisé comme modèle de représentation des données dès que celles-ci sont intrinsèquement liées entre elles. Il permet d'exprimer les relations et de révéler les dépendances entre ces données [Lop05]. L'analyse de ces graphes a pour objectif de concevoir des représentations synthétiques qui puissent exprimer l'interaction entre les différentes données représentées [Big93].

Après avoir défini les données et la notion de similarité entre ces données, nous allons montrer comment représenter ces notions sous la forme d'un graphe.

1.4.1 Définition

D'une manière générale, les données sont représentées sous forme d'un graphe de similarité non orienté et pondéré de façon à modéliser la relation de voisinage de ces différentes données. Ce graphe est défini comme suit : $\mathscr{G} = (\mathscr{N}, \mathscr{E})$ où :

- $\mathscr{N}$ est l'ensemble des noeuds : à chaque point x_i on associe un noeud s_i.
- $\mathscr{E}$ est l'ensemble d'arcs entre les différents noeuds. Il correspond au produit cartésien $\mathscr{N} \times \mathscr{N}$.

A chaque arc reliant deux noeuds s_i et s_j $(i \neq j)$ est attribué un poids. Ce poids n'est autre qu'une fonction de similarité w_{ij} $(0 \leq w_{ij} \leq 1)$ calculée entre les données x_i et x_j comme précisé dans le paragraphe 1.3.

Exemple :

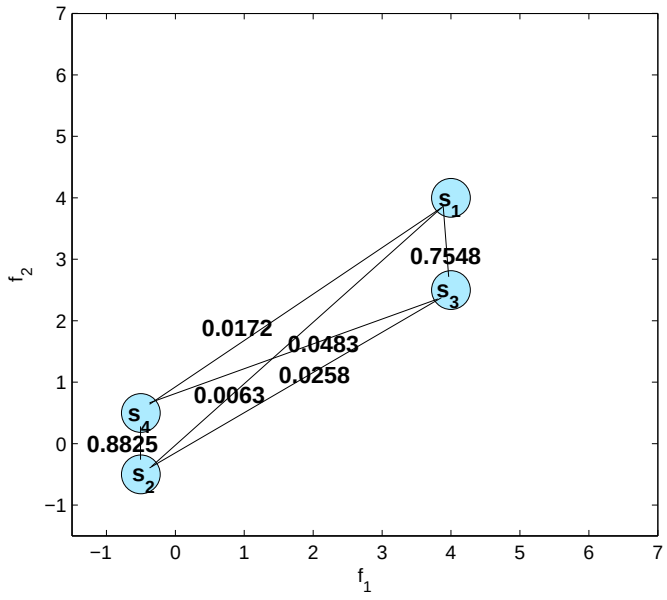


Figure 1.2 : Représentation des données et du graphe correspondant.

Afin de mieux comprendre la représentation de données par un graphe de similarité, nous allons reprendre l'exemple précédent avec les données issues de la matrice (1.10). Pour construire le graphe représentatif de ces données (voir figure 1.2), on associe à chacun des points x_i dans l'espace $\mathbb{R}^2$ un noeud noté s_i. Les 4 points seront ainsi représentés par 4 noeuds. Ensuite, comme

un arc relie chaque paire de noeuds deux à deux, nous aurons un total de 6 arcs. Nous pondérons chaque arc reliant deux noeuds s_i et s_j par la similarité w_{ij} entre leurs données correspondantes x_i et x_j en utilisant la matrice de similarité (1.13) basée sur la distance Euclidienne de l'équation (1.8) construite ci-dessus. Par exemple, l'arc reliant les noeuds s_3 et s_4 est pondéré par la valeur 0.0483 puisque la similarité entre les points x_3 et x_4 est $w_{34} = w_{43} = 0.0483$.
Il est important de signaler que la similarité au sein d'un même noeud n'est pas représentée dans ce graphe car elle est toujours égale à 1.

1.4.2 Types de graphes

Il existe principalement deux types de graphes qui se distinguent par le nombre d'arcs reliant les noeuds entre eux [Lux07].

- **Graphe complètement connecté :** Tous les noeuds du graphe sont connectés entre eux. Cela induit alors un très grand nombre d'arcs même pour un petit nombre de noeuds. En effet, pour n données représentées par n noeuds du graphe, correspondent $n(n-1)/2$ arcs. Connecter tous les noeuds du graphe entre eux n'est pas toujours utile. Dans ce cas, il est préférable d'utiliser un graphe partiellement connecté.
- **Graphe partiellement connecté :** il se caractérise par des connexions partielles entre les noeuds liée à la notion de voisinage. On en distingue 2 catégories :
  - $\varepsilon-$**voisinage** : seuls les noeuds s_i, s_j associés aux points x_i et x_j, dont la distance δ_{ij} dans l'espace d'attributs est inférieure à un certain seuil ε sont connectés, ε étant un réel fixé par l'utilisateur.
  - $k-$ **voisinage** : le noeud s_i associé au point x_i est connecté au noeud s_j associé au point x_j si x_j est parmi les k plus proches voisins de x_i au sens d'une distance dans l'espace d'attributs, k étant un nombre entier fixé par l'utilisateur.

Il faut noter que la construction du graphe partiellement connecté est uniquement basée sur la distance séparant les points dans l'espace d'attributs.

Il est intéressant d'étudier les coûts de calcul de ces 3 types de graphes en distinguant le coût de construction du graphe, le coût de stockage du graphe ainsi que le coût de parcours du graphe. Il est vrai que le coût de stockage ainsi que le coût de parcours du graphe partiellement connecté ($\varepsilon-$voisinage et $k-$ voisinage) sont inférieurs respectivement au coût de stockage et au coût de parcours du graphe complètement connecté, puisque seule une partie des arcs de connexion est représentée dans ce type de graphe.

Cependant, la construction du graphe complètement connecté nécessite $n(n-1)/2$ opérations pour le calcul des différentes valeurs de similarité. Pour un graphe $\varepsilon-$voisinage , à ce coût seront ajoutées $n(n-1)/2$ opérations afin de seuiller les différentes valeurs de similarité et d'éliminer les valeurs inférieures au seuil. Tandis que, pour un graphe de $k-$voisinage, à ce coût sera ajouté le coût du tri des $(n-1)$ similarités, à savoir, $(n(n-1)/2)log(n(n-1)/2)$ opérations et le coût d'extraction des $k-$plus proches voisins de chacune des n données qui atteint kn opérations.

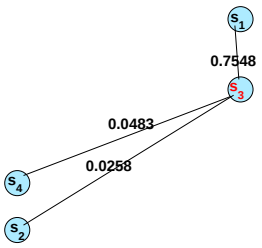
Exemple :

Afin de mieux comprendre la différence entre le graphe complètement connecté, le graphe ε-voisinage et le graphe k-voisinage, nous allons reprendre l'exemple précédent de la matrice (1.10). Nous illustrons les différents types de graphe en prenant en considération le noeud s_3 du graphe représentatif du point x_3 et en nous appuyant sur la matrice de distance Δ (1.11) et sur la matrice de similarité W (1.13).

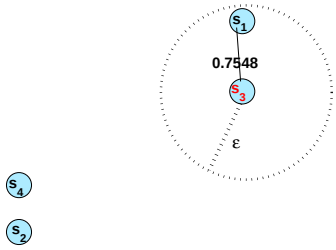
La figure 1.3(a) illustre le cas du graphe complètement connecté où le noeud s_3 est connecté à tous les autres noeuds du graphe.

La figure 1.3(b) illustre le cas du graphe de $\varepsilon-$ voisinage ($\varepsilon = 2$). Puisque seule $\delta_{31} = 1.5$ est inférieure à ε d'après la matrice Δ (1.11), le noeud s_3 est connecté au noeud s_1 situé dans un disque centré en s_3 et de rayon ε.

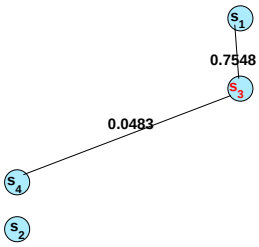
La figure 1.3(c) illustre le cas du graphe de k-voisinage pour k=2. D'après la matrice de distance



(a) Graphe totalement connecté



(b) Graphe de ε voisinage (ε=2)



(c) Graphe des k plus proches voisins (k=2)

Figure 1.3 : Les différents types de graphes de similarité en représentant uniquement les arcs du noeud s_3.

Δ, $\delta_{31} = 1.5$, $\delta_{32} = 5.4083$, $\delta_{34} = 4.9244$. Les points x_1 et x_4 sont alors les 2 points les plus proches de x_3 dans l'espace d'attributs. Le noeud s_3 est connecté aux noeuds s_1 et s_4 qui sont associés aux 2 plus proches voisins de x_3 dans l'espace d'attributs.

Il est important de noter que les valeurs de k et ε ont une influence directe sur le nombre d'arcs du graphe. Plus k et ε sont grands, plus le nombre d'arcs reliant chaque noeud à ses voisins augmente.

1.4.3 Caractérisation d'un graphe

Matrice de similarité :

On associe à un graphe, sa matrice de similarité définie à partir des poids des arcs de connexion de ses noeuds.

La matrice de similarité $W(n \times n)$ entre les n noeuds du graphe est une matrice symétrique et positive. Elle est définie par :

$$W = \begin{bmatrix} 1 & \dots & w_{1i} & \dots & w_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{i1} & \dots & 1 & \dots & w_{in} \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & \dots & w_{ni} & \dots & 1 \end{bmatrix} \tag{1.14}$$

- $w_{ij} \neq 0$ si les deux noeuds du graphe s_i et s_j sont connectés
- $w_{ij} = 0$ si les deux noeuds s_i et s_j ne sont pas connectés
- $w_{ii} = 1$
- $w_{ij} = w_{ji}$. Par conséquent, $W^T = W$.

A partir de la matrice de similarité, on peut exprimer un certain nombre de caractéristiques du graphe associé. La définition de ces caractéristiques est importante pour les chapitres suivants :

- **Degré :**

  Le degré d_i d'un noeud $s_i \in \mathcal{N}$ est défini par la somme des éléments de la $i^{ème}$ ligne de W :

$$d_i = \sum_{j=1}^{n} w_{ij} \tag{1.15}$$

Dans le cas d'un graphe partiellement connecté, le degré d_i peut être considéré comme une mesure de la densité au voisinage du point x_i représenté par le noeud s_i dans le graphe G.

– **La matrice des degrés**

La matrice des degrés $D(n \times n)$ est une matrice diagonale définie par $D_{ii} = d_i$,

$$D = \begin{bmatrix} d_1 & 0 & \dots & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_i & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_n \end{bmatrix} \tag{1.16}$$

Exemple :

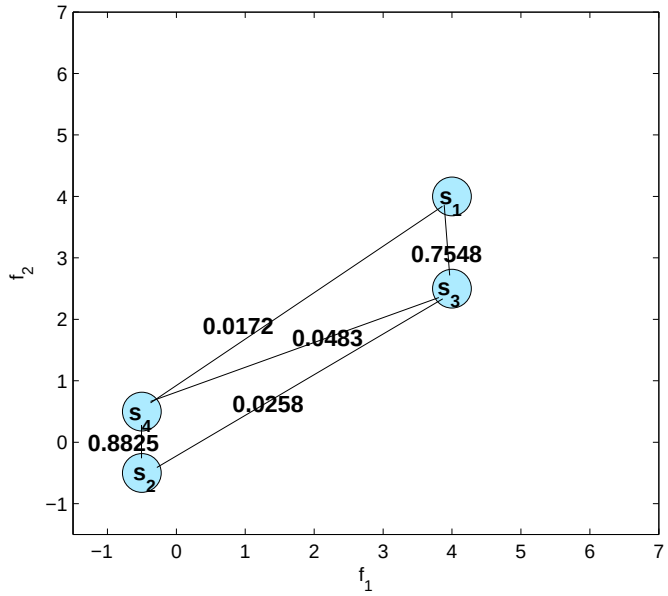


Figure 1.4 : Graphe de similarité type 2-voisinage.

Nous reprenons le même exemple des données de la matrice (1.10) et nous étudions le cas du graphe partiellement connecté. La figure 1.4 représente le graphe de 2-voisinage de ces don-

nées. Ce graphe est partiellement connecté, chaque noeud du graphe étant connecté à ses deux plus proches voisins en utilisant la matrice de distance Δ (1.11). Ainsi, le noeud s_1 est connecté aux noeuds s_3 et s_4, le noeud s_2 est connecté aux noeuds s_3 et s_4, le noeud s_3 est connecté aux noeuds s_1 et s_4 et le noeud s_4 est connecté aux noeuds s_2 et s_3.

La matrice de similarité W associée à ce graphe est alors :

$$W = \begin{bmatrix} 1 & 0 & 0.7548 & 0.0172 \\ 0 & 1 & 0.0258 & 0.8825 \\ 0.7548 & 0.0258 & 1 & 0.0483 \\ 0.0172 & 0.8825 & 0.0483 & 1 \end{bmatrix} \tag{1.17}$$

On peut noter que cette matrice de similarité doit être symétrique. C'est pour cette raison que, par exemple, w_{32} atteint la valeur 0.0258 alors que le noeud s_2 n'est pas parmi les 2 plus proches voisins du noeud s_3.

Nous remarquons que comme les noeuds s_1 et s_2 ne sont pas connectés, leur similarité correspondante est alors $w_{12} = w_{21} = 0$.

La matrice de degré D qui en résulte est la suivante :

$$D = \begin{bmatrix} 1.7720 & 0 & 0 & 0 \\ 0 & 1.9083 & 0 & 0 \\ 0 & 0 & 1.8289 & 0 \\ 0 & 0 & 0 & 1.9480 \end{bmatrix} \tag{1.18}$$

1.5 Représentation contextuelle de la connaissance a priori

Dans la première partie de ce chapitre, nous nous sommes intéressée à la représentation des données et à leur similarité grâce à l'utilisation de graphes sans pour autant s'intéresser à la représentation des connaissances disponibles pouvant caractériser ces données ainsi que leur

contexte d'étude. Mais avant d'introduire ces connaissances, il est essentiel de définir la notion de classification.

Le problème de la classification de données est identifié comme l'une des problématiques majeures en extraction des connaissances à partir des données. La classification consiste à regrouper les données similaires en sous-ensembles, appelés classes. Les données qui ne sont pas similaires doivent être regroupées dans des classes différentes.

En général, en plus des données, on dispose de connaissances a priori sur le domaine [CCM03]. Ces connaissances du domaine peuvent être diverses : nombre maximal ou minimal de classes, nombre maximal ou minimal de données dans une classe, distance maximale ou minimale inter-classes, dispersion minimale ou maximale des classes, ... Tenir compte de ces informations dans le processus de décision est d'une importance capitale pour améliorer les performances de classification.

Dans notre travail, nous nous intéressons à la connaissance a priori disponible sous forme de labels de classes de données.

Par ailleurs, un autre de type de connaissances, moins exigeant que les labels des classes a été récemment introduit par Wagstaff et al. [WC00]. Cela consiste à spécifier pour un couple de données si elles doivent être regroupées ou non au sein d'une même classe.

Selon l'absence ou la présence de cette connaissance, le contexte de représentation et d'interprétation est dit non-supervisé, supervisé, ou semi supervisé. Dans cette partie, nous allons présenter les principales caractéristiques de chacun de ces contextes.

1.5.1 Contexte non supervisé

Dans un contexte non supervisé, nous ne disposons que de la matrice de données X. La construction du graphe de similarité ainsi que de la matrice de similarité ne s'appuie sur aucune information concernant la structure des classes en présence.

Notons que, tous les exemples de graphes et de matrices de similarité précédents ont été abor-

dés dans le contexte non supervisé puisque les labels des classes des données n'étaient pas renseignés.

1.5.2 Contexte supervisé

Dans un contexte supervisé, nous disposons de toute l'information sur les labels des classes des données. A la matrice de données X est associé le vecteur Y des labels des classes des données défini comme suit :

$$Y = \begin{bmatrix} y_1 \\ \dots \\ y_i \\ \dots \\ y_n \end{bmatrix} \tag{1.19}$$

où $y_i \in \{1,...,c\}$, c étant le nombre de classes des données noté ω. A chaque donnée x_i est associée un label de classe y_i.

Le graphe de similarité ainsi que la matrice de similarité du graphe peuvent être définis d'une manière supervisée en fonction des labels disponibles. Le graphe G est un graphe coloré : les noeuds représentant des données de la même classes ont la même couleur. De même, une connexion est établie entre deux noeuds s_i et s_j si leurs données correspondantes x_i et x_j ont le même label (y_i=y_j).

La matrice de similarité $W(n \times n)$ est définie comme suit :

$$w_{ij} = \begin{cases} 1 \text{ si } y_i = y_j \\ 0 \text{ sinon} \end{cases} \tag{1.20}$$

Les éléments de la matrice W sont égaux à 1 (similarité entre couples de données appartenant à la même classe) ou égaux à 0 (similarité entre couples de données appartenant à deux classes différentes). Nous pouvons noter que cette matrice de similarité est dite binaire, car elle ne tient

pas compte des distances séparant les points associés aux données dans l'espace des attributs.

Zhao et al. proposent une autre façon de définir la matrice de similarité dans un contexte supervisé comme suit [ZL07c] :

$$w_{ij} = \begin{cases} \frac{1}{n_\omega} \text{ si } y_i = y_j = \omega \\ 0 \text{ sinon} \end{cases} \tag{1.21}$$

n_ω est le nombre de données de la classe ω. La similarité entre deux données appartenant à une même classe est alors la probabilité a priori qu'une donnée appartienne à cette classe.

Exemple : Pour bien illustrer le contexte supervisé, nous reprenons l'exemple de la section 1.3. Supposons qu'en plus de la matrice de données X (1.10), nous disposons du vecteur Y de labels de classes des données défini comme suit :

$$Y = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix} \tag{1.22}$$

Ces données sont alors divisées en deux classes à savoir la classe 1 contenant les données x_1 et x_3 et la classe 2 contenant les données x_2 et x_4.

La figure 1.5 représente ces données labellisées dans l'espace $\mathbb{R}^2$. Les données de la classe 1 sont représentées par un carré rouge tandis que les données de la classe 2 sont représentées par une croix bleue.

La figure 1.6 illustre le graphe de similarité représentatif de ces données. Seuls des arcs de connexion sont établis entre les noeuds s_1 et s_3 d'une part ainsi qu'entre les noeuds s_2 et s_4 d'autre part puisque leurs données correspondantes appartiennent aux mêmes classes. Les noeuds représentatifs de données appartenant à des classes différentes ne sont pas connectés.

La matrice de similarité binaire associée à ce graphe construit de manière supervisée est définie

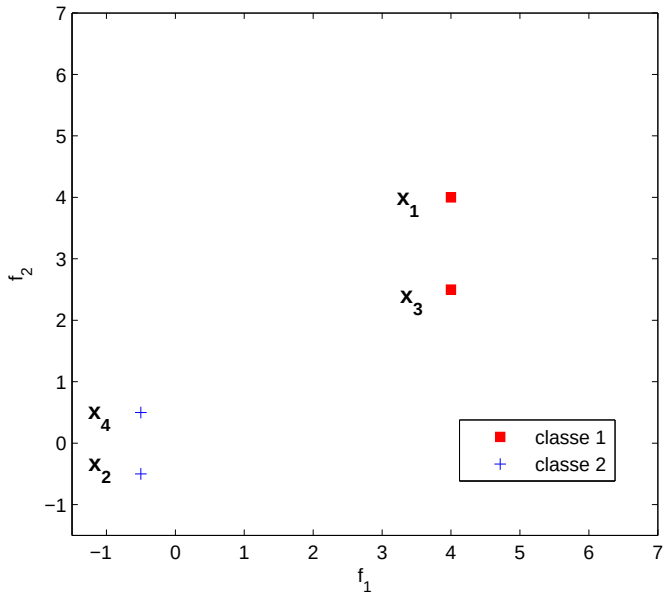


Figure 1.5 : Données supervisées projetées dans l'espace des attributs.

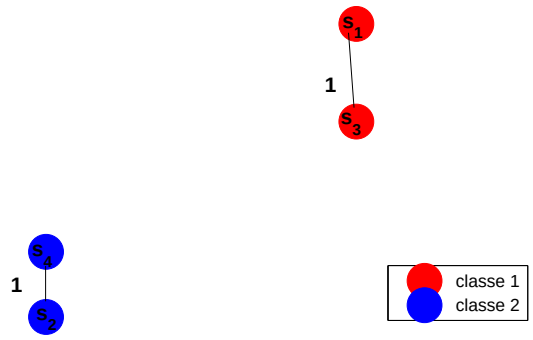


Figure 1.6 : Graphe de similarité supervisé.

par :

$$W = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{1.23}$$

1.5.3 Contexte semi-supervisé

La labellisation a priori de toutes les données nécessite l'intervention d'un expert humain. C'est une opération difficile voire fastidieuse lorsque le nombre de données est important.
Dans des applications concrètes, il est souvent impossible que l'expert puisse assigner toutes les données d'apprentissage aux classes en présence.
Le contexte semi-supervisé qui se situe à l'intersection entre le contexte supervisé et le contexte non supervisé, est alors une solution alternative [JZ97]. Il se caractérise par la présence de quelques informations disponibles sur l'ensemble des données. Ces informations sont représentées soit sous la forme de quelques données labellisées, soit sous la forme de ressemblance ou dissemblance au sein de couples de données.

1.5.3.1 Présence de sous ensembles de données labellisées

Le contexte semi-supervisé peut se caractériser par la présence de quelques labels disponibles sur les données. L'ensemble de données $\mathscr{X}$ est alors divisé en deux sous-ensembles $\mathscr{X} = \{\mathscr{X}_l \cup \mathscr{X}_u\}$. $\mathscr{X}_l$ est le sous-ensemble des l données labellisées pour lequel le vecteur Y_l est disponible. $\mathscr{X}_u$ est le sous-ensemble des u données pour lesquelles aucune information sur leurs labels n'est disponible. Les nombres l et u respectent la relation $l+u=n$.
Le vecteur Y_l est alors défini comme suit :

$$Y_l = \begin{bmatrix} y_1 \\ \dots \\ y_i \\ \dots \\ y_l \end{bmatrix}. \tag{1.24}$$

où $y_i \in (1,\dots,c)$, c étant le nombre de classes disponibles.
Le graphe de similarité ainsi que la matrice de similarité prennent alors en considération les données labellisées et les données non labellisées. Le graphe G est un graphe coloré, une couleur

par classe. Une connexion est établie entre deux noeuds s_i et s_j si leurs données correspondantes x_i et x_j ont le même label (y_i=y_j) ou si les données x_i ou x_j ne sont pas labellisées. Dans ce dernier cas, la similarité sera évaluée par la fonction Gaussienne de l'équation (1.8).

La matrice de similarité $W(n \times n)$ est définie comme suit :

$$w_{ij} = \begin{cases} 1 \text{ si } (x_i, x_j) \in (\mathscr{X}_l \times \mathscr{X}_l) \text{ et } y_i = y_j \\ 0 \text{ si } (x_i, x_j) \in (\mathscr{X}_l \times \mathscr{X}_l) \text{ et } y_i \neq y_j \\ \exp(-\frac{1}{2\sigma^2}\delta_{ij}^2) \text{ (voir équation 1.8) sinon} \end{cases} \tag{1.25}$$

Exemple : Pour illustrer le contexte semi-supervisé, reprenons l'exemple de la section 1.3. Considérons le cas où l'expert a assigné les données x_1, x_2, x_3 à deux classes, tandis que, nous ne disposons d'aucune information sur la classe d'appartenance de x_4. $\mathscr{X}_l = \{x_1, x_2, x_3\}$ et $\mathscr{X}_u = \{x_4\}$.

Les matrices sont alors :

$$X_l = \begin{bmatrix} 1 & 2 \\ -1 & -0.5 \\ 2 & 2 \end{bmatrix} \tag{1.26}$$

$$Y_l = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \tag{1.27}$$

$$X_u = \begin{bmatrix} -1 & -0 \end{bmatrix} \tag{1.28}$$

Dans ce cas, les données x_1, x_2 et x_3 sont labellisées, les données x_1 et x_3 ont le label de la classe 1, la donnée x_2 a le label de la classe 2, tandis que la donnée x_4 n'est pas labellisée. Les données sont représentées dans l'espace $\mathbb{R}^2$ par la figure 1.7.

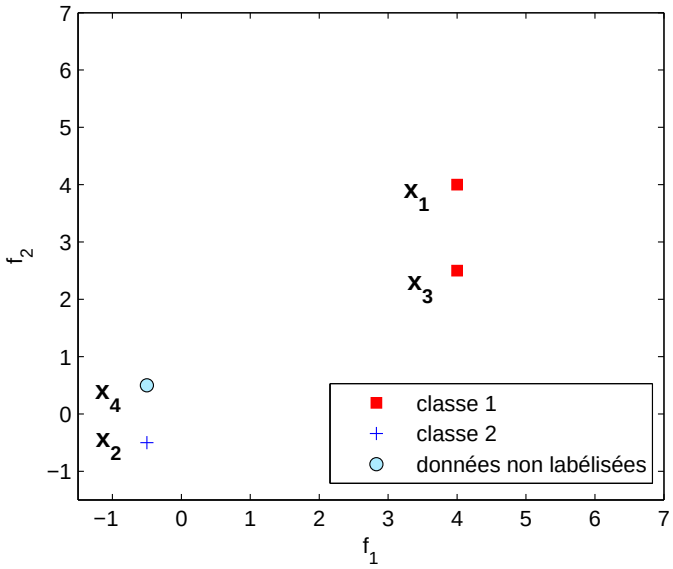


Figure 1.7 : Données dans un contexte semi-supervisé projetées dans l'espace des attributs.

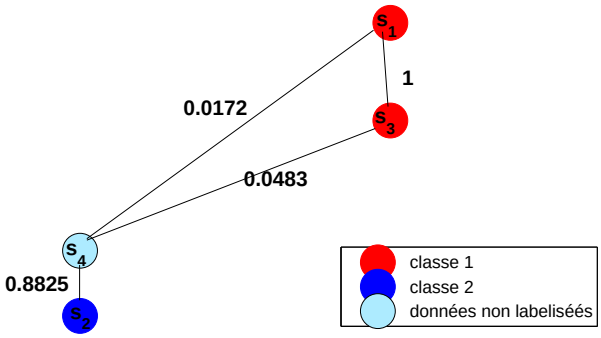


Figure 1.8 : Graphe de similarité semi-supervisé.

La figure 1.8 illustre le graphe de similarité représentatif de ces données. Puisque seules les données x_1 et x_3 sont assignées à la même classe, une connexion existe entre les noeuds s_1 et s_3, tandis que les noeuds s_1 et s_2 et les noeuds s_2 et s_3 ne sont pas connectés. De même, le point x_4 n'étant pas labellisé, le noeud correspondant s_4 est connecté à tous les autres noeuds du graphe. La matrice de similarité associé à ce graphe semi-supervisé est définie par :

$$W = \begin{bmatrix} 1 & 0 & 1 & 0.0172 \\ 0 & 1 & 0 & 0.8825 \\ 1 & 0 & 1 & 0.0483 \\ 0.00172 & 0.8825 & 0.0483 & 1 \end{bmatrix} \tag{1.29}$$

La valeur 1 correspond à une similarité entre données de la même classe et la valeur 0 correspond à une similarité entre données de classes différentes, les autres similarités étant calculées en utilisant l'équation (1.8).

Notons que dans cet exemple, l'expert a attribué le même label de classe pour des données proches dans l'espace d'attributs considéré ($\delta_{13} = 1.5$) et des labels différents pour des données éloignées ($\delta_{14} = 5.7009$, $\delta_{34} = 4.9244$). En d'autres termes, la labellisation des données par l'expert est en accord avec la distance séparant ces données dans l'espace considéré. Cependant, il existe des cas où la labellisation des données par l'expert n'est pas en accord avec le calcul de cette distance.

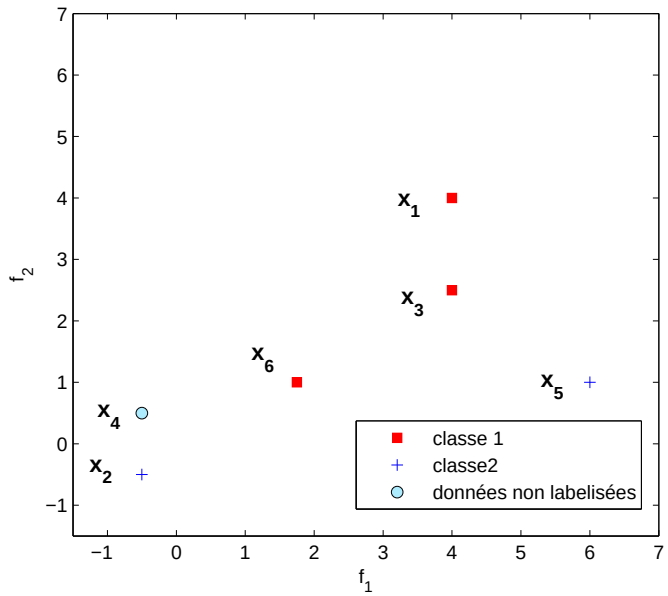


Figure 1.9 : Données d'apprentissage projetées dans l'espace des attributs.

Pour illustrer cela, nous allons garder le même exemple précédent, en ajoutant deux données supplémentaires x_5 (6,1) et x_6 (1.75,1). L'expert décide alors d'attribuer à la donnée x_5 le label 2

désignant qu'elle appartient à la classe 2, et à la donnée x_6 le label 1 désignant qu'elle appartient à la classe 1.

La totalité des données peuvent alors être représentées dans l'espace $\mathbb{R}^2$ par la figure 1.9. Nous pouvons ainsi remarquer que, dans l'espace d'attributs, le point x_5 est plus proche des points x_1 et x_3 de la classe 1 que du point x_2, alors que la donnée x_5 a été affectée à la classe 2 (même label de classe de la donnée x_2). Concernant le point x_6, il est à égale distance des points x_2 et x_3 dans l'espace d'attributs. L'expert a décidé d'affecter la donnée x_6 à la classe 1 (même label de classe de la donnée x_3).

La labellisation des données par l'expert est donc une opération indépendante de la distance entre leurs points correspondants dans l'espace d'attributs.

Cet exemple souligne le soin qui doit être apporté à la mesure de similarité entre les données labellisées. Cette dernière étant déduite des labels de classes, elle est totalement indépendante de la distance séparant les données dans l'espace considéré. Le fait que la distance calculée dans cet espace d'attributs ne respecte pas la décision a priori de l'expert, montre que la structure des données au sein de cet espace d'attributs n'est pas concordante avec les informations de labels fournies par cet expert.

1.5.3.2 Présence de sous ensembles de données contraintes

Un autre de type de connaissances, connu sous le nom de contraintes, moins exigeant que les labels des classes a été récemment introduit par Wagstaff et al. [WC00]. Il s'agit de mentionner pour quelques paires de données si elles sont similaires et doivent alors être regroupées ensemble, ces liens sont appelés contraintes must-link, ou si elles sont dissimilaires et donc ne doivent pas être regroupées ensemble, ces liens étant appelés contraintes cannot-link.

La définition de contraintes demande moins de connaissances et d'efforts de la part de l'expert que la labellisation des données qui nécessite d'avoir des informations précises sur leurs classes d'appartenance .

L'expert doit construire le sous-ensemble $\mathscr{M}$ des contraintes "must-link" de cardinal $|\mathscr{M}|$ et le

sous-ensemble $\mathscr{C}$ des contraintes "cannot-link" de cardinal $|\mathscr{C}|$ définis comme suit :

$\mathscr{M} = \{(x_i,x_j) \in \mathscr{X} \times \mathscr{X}$, *tel que* x_i *et* x_j *doivent être regroupées ensemble* $\}$.

$\mathscr{C} = \{(x_i,x_j) \in \mathscr{X} \times \mathscr{X}$, *tel que* x_i *et* x_j *ne doivent pas être regroupées ensemble* $\}$.

Il est important de signaler que les contraintes must-link sont symétriques et transitives, tandis que les contraintes cannot-link sont symétriques mais ne sont pas transitives.

Les cardinaux de ces deux sous-ensembles ($|\mathscr{M}|$) et ($|\mathscr{C}|$) sont en général plus petits que C_n^2 le nombre de toutes les contraintes possibles générées par les *n* données.

Il est clair que ces contraintes sont beaucoup plus faciles à obtenir par l'expert que les labels des données. Ils formalisent pour deux données si elles doivent être regroupées ensemble ou non, sans fournir d'informations sur les classes présentes (nombre, structure, ...). En effet, les données partiellement labellisées peuvent être transformées en contraintes must-link et cannot-link mais pas l'inverse. Cela consiste à imposer la contrainte must-link entre les données ayant le même label et la contrainte cannot-link pour deux données ayant des labels différents.

A partir des deux sous-ensembles de contraintes, Zhang et al. [ZCZ08] proposent de représenter les relations entre les données avec deux graphes spécifiques : le graphe des must-link $G^{\mathscr{M}}$ et le graphe des cannot-link $G^{\mathscr{C}}$ comme suit :

- Le graphe des must-link $G^{\mathscr{M}}$: une connexion est établie entre deux noeuds s_i et s_j si $(x_i, x_j) \in \mathscr{M}$.
- Le graphe des cannot-link $G^{\mathscr{C}}$: une connexion est établie entre deux noeuds s_i et s_j si $(x_i, x_j) \in \mathscr{C}$.

A partir des deux graphes $G^{\mathscr{M}}$ et $G^{\mathscr{C}}$, les matrices $W^{\mathscr{M}}$ $(n \times n)$ et $W^{\mathscr{C}}$ $(n \times n)$ peuvent être définies par :

$$w_{ij}^{\mathscr{M}} = \begin{cases} 1 & \text{si } (x_i, x_j) \in \mathscr{M} \text{ ou } (x_j, x_i) \in \mathscr{M} \\ 0 & \text{sinon} \end{cases} \tag{1.30}$$

$$w_{ij}^{\mathscr{C}} = \begin{cases} 1 & \text{si } (x_i, x_j) \in \mathscr{C} \text{ ou } (x_j, x_i) \in \mathscr{C} \\ 0 & \text{sinon} \end{cases} \tag{1.31}$$

Exemple : Pour bien illustrer les données partiellement contraintes, nous allons reprendre notre exemple de la section 1.3. Nous supposons qu'en plus de la matrice de données X, l'utilisateur définit les deux sous ensembles de contraintes $\mathscr{M}$ et $\mathscr{C}$ suivants :

$\mathscr{M} = \{(x_2, x_4)\}$,

$\mathscr{C} = \{(x_1, x_4); (x_2, x_3)\}$.

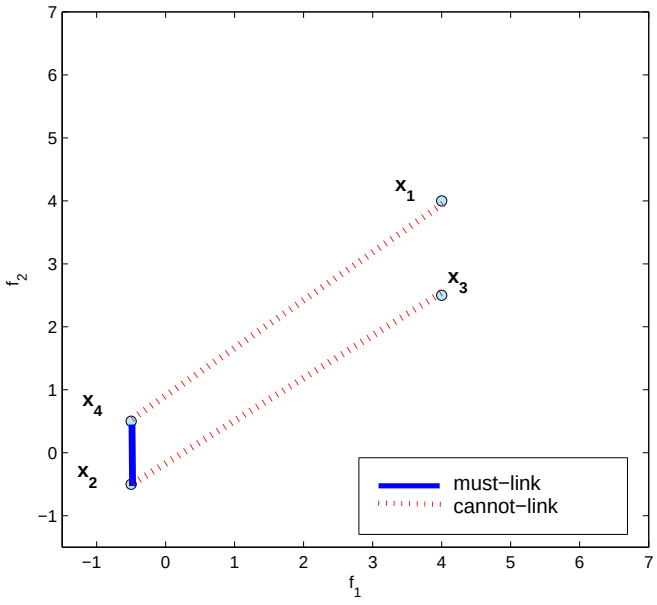


Figure 1.10 : Données partiellement contraintes projetées dans l'espace d'attributs.

Les données peuvent être représentées dans l'espace $\mathbb{R}^2$ par la figure 1.10. L'utilisateur a donc défini une contrainte must-link entre les données x_2 et x_4 représentée par un trait plein sur la figure et deux contraintes cannot-link entre les données x_1 et x_4 d'une part et les données x_2 et x_3 d'autre part représentées par deux traits pointillés.

A partir de ces données nous pouvons construire le graphe des must-link $G^{\mathscr{M}}$ et le graphe des cannot-link $G^{\mathscr{C}}$ représentées par les figures 1.11(a) et 1.11(b). Dans le graphe $G^{\mathscr{M}}$ juste les noeuds s_2 et s_4 sont connectés puisqu'il existe une seule contrainte must-link entre leurs données correspondantes x_2 et x_4. Pour $G^{\mathscr{C}}$, les noeuds s_1 et s_4 d'une part et les noeuds s_2 et s_3 sont connectés puisqu'il existe une contrainte cannot-link entre leurs données correspondantes.

Aux deux graphes $G^{\mathscr{M}}$ et $G^{\mathscr{C}}$, nous associons les matrices $W^{\mathscr{M}}$ (4×4) et $W^{\mathscr{C}}$ (4×4) :

(a) Graphe des must-link $G^{\mathscr{M}}$ (b) Graphe des cannot-link $G^{\mathscr{C}}$

Figure 1.11 : Les graphes des must-link $G^{\mathscr{M}}$ et des cannot-link $G^{\mathscr{C}}$.

$$W^{\mathscr{M}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{1.32}$$

$$W^{\mathscr{C}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{1.33}$$

Comme dans le cas des données labellisées, la définition des contraintes est indépendante de la distance entre leur points correspondants dans l'espace d'attributs. Reprenons notre exemple, en supposant cette fois que l'utilisateur définit les deux sous ensembles de contraintes $\mathscr{M}$ et $\mathscr{C}$ suivants :

$\mathscr{M} = \{(x_1, x_2)\}$,

$\mathscr{C} = \{(x_2, x_4); (x_3, x_4)\}$.

Les données peuvent être représentées dans l'espace $\mathbb{R}^2$ par la figure 1.12. L'utilisateur a défini

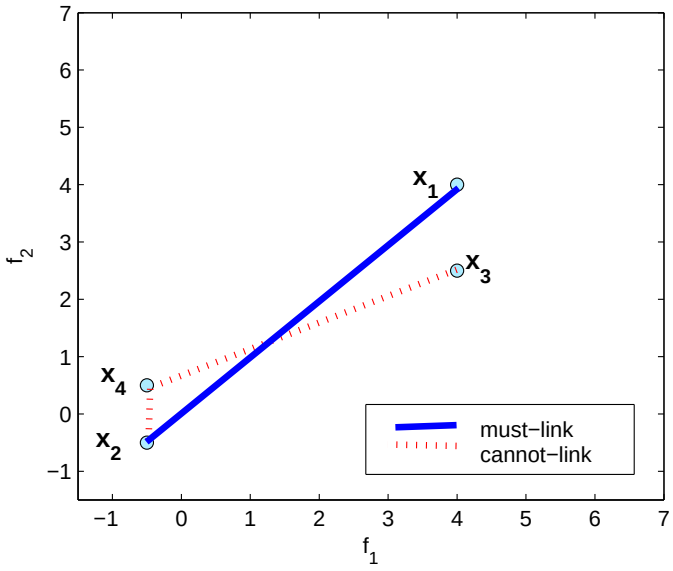


Figure 1.12 : Données partiellement contraintes projetées dans l'espace d'attributs.

une contrainte must-link entre les données x_1 et x_2, bien que leur points correspondants soient éloignés dans l'espace d'attributs. Par contre, l'utilisateur a défini deux contraintes cannot-link entre les données x_2 et x_4 d'une part et les données x_3 et x_4 d'autre part, bien que leurs points correspondants soient proches dans l'espace d'attributs.

On peut alors en conclure que l'espace d'attributs ne permet pas de respecter le choix de l'utilisateur en terme de regroupement de données. Cet exemple met en évidence la nécessité de trouver un autre espace de représentation qui respecte au mieux les connaissances a priori fournies par l'utilisateur.

1.6 Conclusion

Dans ce chapitre, nous avons formalisé sous forme de matrices et de graphes la représentation des données et de la connaissance a priori sur leur regroupement en classes. Dans une première partie, nous avons défini la notion de données caractérisées par des attributs. Pour comparer les données entre elles, nous avons ensuite distingué la notion de distance calculée dans l'espace des attributs de celle de similarité qui ne respecte pas forcément la propriété d'inégalité triangulaire. La fonction de similarité que nous avons retenue est la fonction Gaus-

sienne basée sur la distance Euclidienne. Cette mesure de similarité entre deux données peut être plus riche qu'une simple distance car elle prend en compte la structure du voisinage centrée en chaque donnée.

Nous nous sommes appuyée sur des exemples simples pour mettre en évidence les différentes propriétés de cette fonction de similarité Gaussienne. Nous avons également montré que certaines fonctions de similarité, telle que celle basée sur la fonction cosinus, ne respectent pas forcement la distance Euclidienne calculée dans l'espace d'attributs.

La structure des données peut être représentée par une matrice collectant leurs coordonnées dans l'espace d'attributs ou par un graphe qui s'appuie uniquement sur leur similarité. Dans cette dernière représentation, la localisation des points représentant les données dans l'espace des attributs est perdue au profit d'une mesure de similarité au sein de chaque paire de données. La construction d'un graphe de similarité partiellement connecté met alors en évidence la structure locale des données, ce qui permettra de contribuer à une analyse locale du regroupement des données en classes.

Cette analyse ne peut être pertinente que si l'espace de représentation des données est correctement choisi. En effet, pour de nombreuses applications concrètes, le nombre d'attributs est plus élevé que le nombre de données. Dans ce cas, l'espace d'attributs est peu occupé par les données qui forment des structures éparses. Il est alors peu opportun d'analyser la structure locale des données. Ce problème, connu sous le nom de "la malédiction de la dimension" nécessite une étape de réduction de la dimension de l'espace d'attributs qui sera abordée dans le chapitre suivant.

La connaissance a priori apportée par l'utilisateur définit le contexte d'apprentissage dans lequel opère la réduction de la dimension. Le contexte supervisé nécessitant une définition complète sur les labels des données se révèle être finalement peu utilisé dans les applications concrètes. Ceci nous a amené a retenir le contexte semi-supervisé où l'utilisateur doit fournir une connaissance a priori partielle sur le regroupement des données. Plutôt que fournir les la-

bels de certaines données, ce qui nécessite de définir les classes à retrouver, nous proposons de suivre la démarche proposée par Wagstaff et al. [WC00], à savoir de formaliser la connaissance a priori sous la forme de contraintes dites 'must-link' (deux données doivent être regroupées au sein d'une même classe) et 'cannot-link' (deux données ne doivent pas être regroupées au sein de la même classe).

Ces contraintes seront utilisées pour réduire la dimension de l'espace d'attributs selon les techniques décrites au chapitre suivant.

Chapitre 2

Sélection d'attributs

2.1 Introduction

Les données utilisées par les applications réelles (images, signaux...) sont souvent caractérisées par un grand nombre d'attributs qui peuvent dépasser le nombre de données elles-même. Ce problème connu sous le nom de "la malédiction de la dimension" constitue un défi pour les différents algorithmes de décision. Considérer un nombre élevé d'attributs d'une part augmente le risque de prendre en considération des attributs redondants ou corrélés ce qui rend ces algorithmes plus complexes et parfois moins performants.

Il est alors nécessaire de procéder à une étape de réduction de la dimension de l'espace des attributs d'entrée. Cette réduction de dimension permet de rendre l'ensemble des données plus représentatif du problème, de réduire l'espace de stockage nécessaire de ces données, ainsi que le temps d'apprentissage et d'exploitation des algorithmes de décision.

Les méthodes de réduction de la dimension peuvent être divisées en deux grandes catégories : l'extraction d'attributs et la sélection d'attributs [LM08], [YL03].

- Les méthodes d'extraction d'attributs consistent à transformer l'ensemble d'attributs de départ en un nouvel ensemble d'attributs, généralement plus petit, tout en conservant autant que possible la structure originale des données. On peut distinguer les méthodes linéaires non-supervisées comme l'Analyse en Composantes Principales (ACP) [Pea01]

[Cas96], les méthodes linéaires supervisées comme l'Analyse Factorielle Discriminante (AFD) [Sap06], les méthodes non linéaires non supervisées comme l'ACP à noyau [SSM96], Locally Linear Embedding (LLE) [RS00], Isometric Feature Mapping (Isomap) et les méthodes non linéaires supervisées comme l'Analyse Factorielle Discriminante à noyau...

Le principal inconvénient de ces méthodes est leur temps de calcul. Une méthode d'extraction d'attributs nécessite le calcul des d attributs initiaux pour ensuite extraire les $\widehat{d}$ attributs pertinents ($\widehat{d} < d$), ces derniers étant obtenus en combinant, linéairement ou non, les d attributs initiaux. Un autre inconvénient des méthodes d'extraction est qu'elles imposent un effort important à l'utilisateur pour interpréter et comprendre la nouvelle représentation des données : il est difficile de donner une interprétation sémantique des attributs extraits, ces derniers étant une combinaison des attributs initiaux.

– Les méthodes de sélection d'attributs [YL03] permettent de choisir un sous-ensemble pertinent d'attributs à partir de l'ensemble original d'attributs selon un critère de performance. Ces méthodes [KS96] permettent alors de caractériser plus rapidement les données et sont donc utilisées pour les applications où les coûts en temps de calcul doivent être minimisés, comme les applications de traitement d'images en temps réel. La sélection d'attributs ne modifie pas la représentation originale des données : les attributs sélectionnés gardent leur sémantique de départ et peuvent alors être interprétés plus facilement par l'utilisateur.

C'est pour cette raison que nous nous sommes intéressée dans ce chapitre aux méthodes de sélection d'attributs et plus particulièrement à celles à base de graphes.

Ainsi dans la section 2.2, nous énumérons les différentes étapes de sélection d'attributs. Ensuite nous détaillons, dans la section 2.3, le score de la Variance et le score Laplacien qui opèrent dans un contexte non supervisé. Les scores de sélection supervisée à savoir, le score de Fisher, le score de l'information mutuelle ainsi que le score Laplacien supervisé sont présentés dans la section 2.4. Dans un contexte semi supervisé, caractérisé par la présence de quelques labels de

données disponibles, le score Laplacien semi-supervisé de la section 2.5 est utilisé. Cependant en présence de contraintes portant sur les paires de données, des scores de contraintes proposés récemment dans la littérature sont présentés dans la section 2.6. Comme ces scores négligent l'information apportée par les données non contraintes, nous introduisons notre score pour la sélection semi-supervisée avec contraintes dans la section 2.7.

Ce chapitre dédié au coeur du problème auquel nous nous attaquons est long car il intègre une étude bibliographique conséquente, complétée d'exemples pédagogiques qui mettent en évidence les subtilités des différentes approches.

2.2 Etapes de sélection d'attributs

La sélection d'attributs est un domaine de recherche actif et en cours de développement dans diverses applications (indexation et recherche d'images, analyse génomique, analyse de documents...). Un grand nombre d'algorithmes ont été proposés dans la littérature pour la sélection d'attributs non-supervisée, supervisée ou semi-supervisée.

Selon Dash [DL97], une procédure de sélection d'attributs est généralement composée de quatre étapes illustrées par la figure 2.1 :

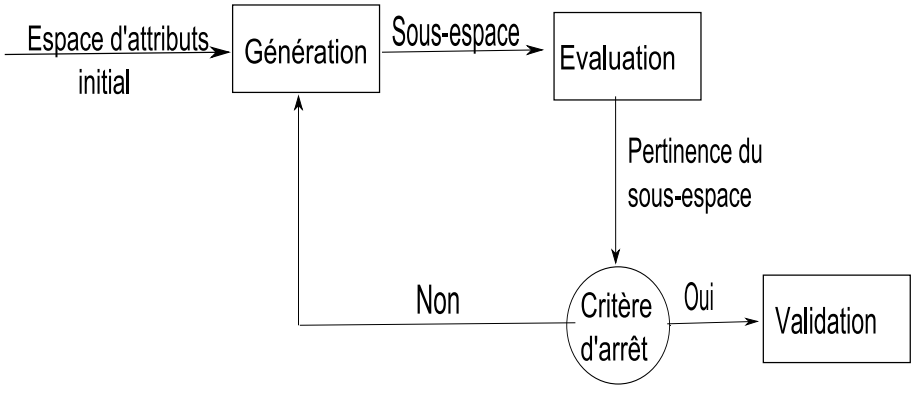


Figure 2.1 : Les différentes étapes d'une procédure de sélection d'attributs.

- la procédure de génération,
- la fonction d'évaluation,
- le critère d'arrêt,

– la procédure de validation.

Avant de détailler chacune de ces étapes, notons que les algorithmes de sélection d'attributs peuvent être divisés en deux catégories : Les algorithmes de classement des attributs (Feature ranking) [BDK+05] et les algorithmes de recherche de sous-ensembles (Subset selection) [YL03]. La première catégorie d'algorithmes consiste à ordonner l'ensemble d'attributs de départ selon un critère d'évaluation et à sélectionner ensuite les $\widehat{d}$ attributs les plus pertinents vis-à-vis du critère utilisé.

La deuxième catégorie recherche le sous-ensemble d'attributs le plus pertinent selon un certain critère de sélection. Ces algorithmes doivent alors trouver le meilleur sous-ensemble d'attributs parmi 2^d sous-ensembles candidats.

La première étape de la procédure de sélection, à savoir la génération des attributs, est donc caractéristique des algorithmes de recherche de sous-ensembles, tandis que les trois autres s'appliquent également aux algorithmes de classement des attributs.

2.2.1 La procédure de génération

La procédure de génération permet, à chaque itération, de générer un sous-ensemble d'attributs qui va être évalué lors de la seconde étape de la procédure de sélection.

Cette procédure de génération peut soit commencer avec un ensemble vide d'attributs, soit avec l'ensemble de tous les attributs, soit avec un sous-ensemble d'attributs choisis aléatoirement. Dans les deux premiers cas, les attributs sont itérativement ajoutés (Forward selection) ou retirés (Backward selection). Dans le troisième cas, soit on ajoute, ou on retire des attributs comme dans les deux premiers cas, soit un nouveau sous-ensemble d'attributs est créé de manière aléatoire à chaque itération (Random generation).

Trois grandes approches de génération ont été proposées dans la littérature, la génération complète, la génération aléatoire et la génération séquentielle [DL97], [Bla06].

En effet, pour un ensemble d'attributs initial de dimension d, le nombre total de sous-ensembles candidats qui peuvent être générés par la procédure de génération est 2^d. Ce nombre est généralement très élevé surtout lorsque le nombre d d'attributs est élevé.

2.2.1.1 La génération complète

Dans la procédure de génération complète, une recherche complète du sous-ensemble d'attributs optimal au sens de la fonction d'évaluation utilisée est effectuée. Une recherche exhaustive est complète, cependant la recherche ne doit pas être exhaustive pour qu'elle soit complète. C'est pour cela, au lieu d'évaluer les 2^d sous-ensemble candidats, différentes fonctions heuristiques peuvent être utilisées afin de réduire l'espace de recherche sans pour autant compromettre les chances de trouver le sous-ensemble optimal. Il s'agit d'utiliser un processus de backtracking permettant de revenir en arrière si la sélection s'engage dans une mauvaise direction de génération.

2.2.1.2 La génération aléatoire

Les procédures de génération aléatoire parcourent au hasard l'ensemble des 2^d sous-ensembles candidats, le sous-ensemble courant n'est alors pas issu d'une augmentation ou diminution d'attributs du sous-ensemble précédent. Cela permet de ne pas arrêter la recherche lorsque la fonction d'évaluation d'un sous-ensemble atteint un optimum local. Cependant, les 2^d sous-ensembles candidats ne sont pas tous évalués, contrairement aux procédures de génération complète. Un nombre maximal d'itérations est imposé afin que les temps de calcul restent raisonnables.

Les algorithmes génétiques (AG), initiés par Holland en 1975 [Hol75], sont les méthodes de génération aléatoire les plus couramment utilisées [Gol89].

L'avantage de la procédure de génération aléatoire est qu'elle ne nécessite pas l'utilisation de fonction d'évaluation monotone. D'autre part, contrairement aux méthodes de génération complète dont la complexité est exponentielle vis-à-vis de la dimension initiale d de l'espace d'attri-

buts, la complexité de calcul des méthodes basées sur une génération aléatoire est quadratique [DLM00] [KS00]. C'est également le cas des méthodes de sélection basées sur les procédures de génération séquentielle.

2.2.1.3 La génération séquentielle

Le principe des procédures de génération séquentielle est d'ajouter ou de supprimer un ou plusieurs attributs au fur et à mesure des itérations. On distingue alors deux approches de génération séquentielle :

- L'approche de type Forward ou Ascendante : cette approche part d'un ensemble vide d'attributs auquel, à chaque itération sont ajoutés un ou plusieurs attributs.
- L'approche de type Backward ou Descendante : c'est l'approche inverse, elle part de l'ensemble total des attributs. Chaque itération permet de supprimer un ou plusieurs attributs.

Les algorithmes utilisant ces approches de génération sont connus par leur simplicité de mise en oeuvre et leur rapidité. Cependant, comme ils n'explorent pas tous les sous-ensembles possibles d'attributs et ne permettent pas de retour arrière pendant la recherche ; ils sont donc sous-optimaux. Il est alors possible d'ajouter (ou de retirer) itérativement les attributs. C'est notamment le cas de l'algorithme plus l-take away r. Cet algorithme consiste tout d'abord à élargir le sous-ensemble d'attributs en répétant l fois la procédure Forward, puis à éliminer des attributs en répétant r fois la procédure Backward. Notons que le choix des paramètres l et r influe sur la qualité des résultats ainsi que sur le temps de calcul.

Cet algorithme est très performant lorsque l'on connait a priori la dimension du sous-espace discriminant, mais il reste cependant très couteux en temps de calcul. Pour réduire les coûts de calcul, tout en tentant de conserver un niveau élevé de performance, on peut utiliser les méthodes flottantes, qui sont une extension de l'algorithme plus l-take away r.

L'algorithme SFFS (Sequential Forward Floating Selection) consiste à appliquer après chaque étape Forward des étapes Backward tant que le sous-espace d'attributs correspondant améliore la fonction d'évaluation. L'algorithme SBFS (Sequential Backward Floating Selection)

applique le même principe à la différence que les deux étapes sont inversées.

2.2.2 La fonction évaluation

La fonction évaluation permet d'évaluer les attributs ou les sous-ensembles d'attributs générés à l'étape précédente. Elle est utilisée pour mesurer :

- la pertinence des attributs en les appréciant de manière individuelle, lorsqu'on utilise un algorithme de sélection par classement des attributs,
- la pertinence des sous-ensembles d'attributs générés par l'une des différentes méthodes de génération présentées ci-dessus, lorsqu'un algorithme de recherche de sous-ensembles est utilisé.

En effet, la sélection d'un sous-ensemble d'attributs optimal est toujours relative au critère utilisé car différents critères ne permettent pas de sélectionner le même sous-ensemble d'attributs optimal.

Différentes fonctions d'évaluation ont été proposées pour évaluer un attribut ou un sous-ensemble d'attributs dans un contexte de sélection. Elles peuvent être classées en cinq approches distinctes [DL97] :

- **Les mesures d'erreur de classification**

  L'attribut ou les sous-ensembles d'attributs considérés sont évalués en fonction de la qualité de la classification obtenue en utilisant ces attributs. Le sous-ensemble d'attributs le plus discriminant est celui pour lequel le taux d'erreur de classification est le plus faible [DLPT82].

- **Les mesures d'information**

  Les mesures d'information déterminent le gain d'information pour un attribut considéré, le gain d'information apporté par un attribut étant estimé à partir des probabilités *a posteriori*. Un attribut f_r est préféré à un attribut f_v si le gain d'information apporté par l'attribut f_r est plus grand que celui apporté par l'attribut f_v [CT91].

– **Les mesures de consistance**

Les mesures de consistance cherchent à évaluer si l'attribut (ou le sous-ensemble d'attributs) étudié contient les informations nécessaires à la discrimination des classes [Sem04]. La figure 2.2 illustre un attribut consistant (f_2) et un attribut non consistant (f_1) : on voit

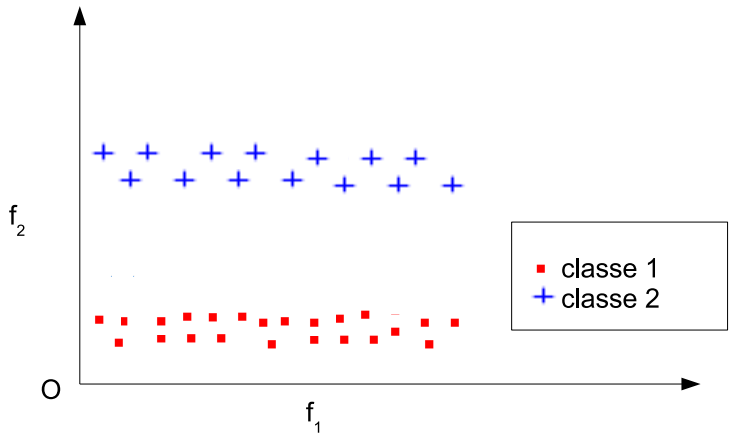


Figure 2.2 : Représentation d'un attribut consistant (f_2) et d'un attribut non consistant (f_1).

aisément dans cette figure que contrairement à l'attribut f_1, l'attribut f_2 permet de discriminer les deux classes en présence.

– **Les mesures de dépendance**

Les mesures de dépendance peuvent être divisées en deux catégories : la première est une mesure de corrélation qui quantifie la dépendance des attributs les uns par rapport aux autres. La deuxième catégorie est une mesure de dépendance qui caractérise la corrélation entre un attribut ou un sous-ensemble d'attributs et une classe [DL97].

– **Les mesures de distance**

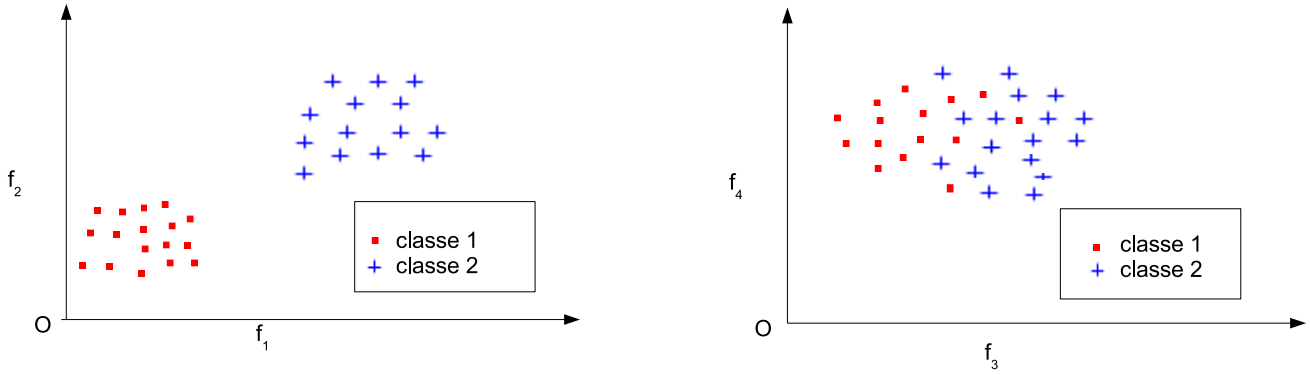
Les mesures de distance sont aussi nommées mesures de séparabilité, divergence ou de discrimination. Un attribut ou un sous-ensemble d'attributs est sélectionné s'il permet une meilleure séparabilité et cohérence des classes.

En effet, le but est de :

– maximiser la dispersion inter-classes (séparabilité), afin que les points représentatifs

des différentes classes forment dans l'espace d'attributs des nuages les plus séparés possibles les uns des autres

– minimiser la dispersion intra-classe (cohérence), afin que les nuages de points représentatifs de chaque classe soient les plus compacts possible.



(a) Sous-espace discriminant des classes 1 et 2 (b) Sous-espace non discriminant des classes 1 et 2

Figure 2.3 : Illustration d'un sous-espace discriminant et d'un sous-espace non discriminant vis-à-vis d'une mesure de distance.

La figure 2.3 illustre un ensemble de points provenant de deux classes représentées dans deux sous-espaces différents. Dans le premier sous-espace formé des attributs f_1 et f_2, les nuages de points correspondant aux deux classes sont compacts et séparés (voir figure 2.3(a)), tandis que dans le second sous-espace formé des attributs f_3 et f_4, ces nuages de points sont proches et étendus (voir figure 2.3(b)). Le premier sous-espace est donc plus discriminant vis-à-vis des deux classes que le second sous-espace.

Selon le critère d'évaluation utilisé dans le processus de sélection d'attributs, nous pouvons distinguer entre les approches de type "wrapper" et les approches de type "filter".

Les approches "wrappers" utilisent le taux d'erreur de classification comme critère d'évaluation (mesures d'erreur de classification) [RJ97]. Ils incorporent alors l'algorithme de classification dans la procédure de recherche et sélection d'attributs. Ces méthodes permettent d'obtenir de bonnes performances. Cependant, l'utilisation de telles méthodes nécessite pour chaque sous-

espace d'attributs candidats d'effectuer la classification, ce qui peut devenir coûteux en temps de calcul surtout lorsque la dimension *d* de l'espace d'entrée est grande. De même, ces méthodes sont très dépendantes de l'algorithme de classification utilisé comme critère d'évaluation. Ce dernier, s'il est mal adapté, pourrait contribuer à une sélection de mauvais attributs.

Les approches "filter" utilisent une fonction d'évaluation basée sur les caractéristiques de l'ensemble des données, indépendamment de tout algorithme de classification, afin de sélectionner certains attributs ou sous-ensemble d'attributs (mesures d'information, mesures de consistance, mesures de dépendance et mesures de distance) [HCN05] [Tal99]. Ces méthodes sont rapides, plus générales et moins coûteuses en temps de calcul, ce qui leur permet d'opérer plus facilement avec des bases de données de très grandes dimensions. Cependant, comme elles sont indépendantes de l'étape de classification, elles ne permettent pas de garantir que le meilleur taux de classification soit obtenu dans l'espace retenu.

Pour combiner les avantages des deux méthodes, des algorithmes hybrides "embedded" ont été proposés. Le processus de sélection d'attributs est effectué conjointement au processus de classification. Une fonction d'évaluation de type "filter" est tout d'abord utilisée pour présélectionner les sous-espaces d'attributs les plus discriminants. Puis les taux d'erreurs de classification obtenus en considérant chaque sous-espace discriminant précédemment sélectionné sont comparés afin de déterminer le sous espace final [Das01].

A cause de leur efficacité de calcul et leur indépendance de tout algorithme de classification, les approches de type "filter" sont plus populaires et d'utilisation courante.

2.2.3 Le critère d'arrêt

Le nombre optimal d'attributs n'étant pas connu a priori, il sera fixé grâce à un critère d'arrêt du processus de sélection. L'utilisation d'une règle pour contrôler la procédure de sélection permet d'arrêter la recherche lorsqu'aucun nouvel attribut n'est suffisamment informatif. C'est un choix souvent défini en fonction de la procédure de recherche [Sem04] et/ou du critère

d'évaluation [PPL97].

Les critères d'arrêt les plus fréquents sont :

- basés sur l'algorithme de génération [ZOD07] : on peut par exemple décider d'arrêter la recherche en fixant un seuil sur le nombre d'attributs à sélectionner ou sur le nombre d'itérations. Cependant, dans de nombreuses applications, le nombre d'attributs à sélectionner est très difficile à fixer au préalable. De même, un critère fondé sur un nombre maximal d'itérations peut s'avérer brutal et arrêter trop tôt ou trop tard la sélection [Sem04].
- basés sur l'évaluation [PPL97] : dans ce cas, on arrête la recherche en fixant un seuil soit sur la fonction d'évaluation, soit sur la différence entre la valeur d'évaluation à l'étape d et la valeur d'évaluation à l'étape $d-1$, c'est-à-dire lorsque l'ajout ou la suppression d'un attribut n'apporte pas un gain de discrimination suffisant. Par exemple, lorsque l'approche "wrapper" ou l'approche "embedded" est utilisée, les taux de bonne classification obtenus par les différents sous-espaces sont comparés pour mesurer le gain d'information. On peut ainsi décider d'arrêter la procédure de sélection dès que ce taux diminue ou alors dès qu'il atteint un certain seuil.

Le choix du critère d'arrêt est ainsi un choix délicat qui reste un problème non résolu dans de nombreux algorithmes de sélection. On ne connait pas à l'avance la dimension $\widehat{d}$ du sous-espace sélectionné.

2.2.4 La validation

La validation ne fait pas partie de la procédure de sélection d'attributs mais elle permet de tester la validité du sous-ensemble d'attributs sélectionnés en effectuant plusieurs tests sur des exemples de données générées artificiellement et/ou sur des données réelles.

L'ensemble des données est généralement divisé en deux sous-ensembles distincts : le sous-ensemble d'apprentissage constitué des prototypes des classes (données avec leurs labels) et le sous-ensemble de test dont on ne connait pas les labels de classes de ses données. Selon

la répartition des données entre ces deux sous-ensembles, il existe différentes approches de validation.

Nous citons :

- La méthode Holdout : les données sont divisées en deux sous-ensembles : le sous-ensemble d'apprentissage et le sous-ensemble de test dans des proportions $\frac{1}{2}$, $\frac{1}{2}$ pour chacun de ses deux sous-ensembles ou $\frac{2}{3}$ pour l'ensemble d'apprentissage et $\frac{1}{3}$ pour l'ensemble de test.
- La méthode de resubstitution : l'ensemble d'apprentissage est utilisé comme ensemble de test.
- La méthode V-validation croisée : l'ensemble des données est partitionné en V parties de tailles à peu près égales. Nous réalisons ainsi V fois la procédure de validation et à chaque fois une des parties constitue l'ensemble test et les V-1 parties restantes sont réunies pour former l'ensemble d'apprentissage.

  La méthode Leaving-one-out est un cas particulier de la méthode V-validation croisée où l'ensemble des données est divisé en *n* parties. La base de test contient alors à chaque fois une seule donnée.

Une fois que les données sont divisées en un ensemble d'apprentissage et un ensemble de test, l'erreur de classification est mesurée sur l'ensemble de test en utilisant les prototypes de classes de l'ensemble d'apprentissage.

Plusieurs méthodes de classification (k-plus proches voisins, réseaux de neurones,...) sont généralement utilisées dans la littérature [LM68] [Roj96].

Ci-dessus, nous avons énuméré les différentes étapes de sélection d'attributs sans donner beaucoup de détails sur chacune d'elles. Plus de détails peuvent être trouvés dans [Por09].

Par la suite, nous allons exposer quelques algorithmes de sélection d'attributs dans le contexte d'apprentissage supervisé, non-supervisé ou semi-supervisé. Ces algorithmes sont des algorithmes de classement individuel des attributs, aucune procédure de génération de sous-espaces

d'attributs n'est donc utilisée. De plus, ces algorithmes sont de type "filter". Ils sont ainsi indépendants de tout algorithme de classification et sont relativement peu coûteux en temps de calcul.

Dans cette partie, nous allons détailler ces différents algorithmes en mettant l'accent sur la fonction d'évaluation utilisée par chacune d'elle. Le critère d'arrêt et la validation sont détaillés dans le chapitre 3.

2.3 Sélection non supervisée

Dans un contexte d'apprentissage non supervisé où nous ne disposons pas des labels de classes des données, il est difficile de définir un critère d'évaluation performant des attributs candidats. Plusieurs algorithmes de sélection d'attributs opérant dans un contexte non-supervisé ont été proposés dans la littérature. Ces algorithmes utilisent les différents critères d'évaluation cités ci-dessus (mesure de corrélation [Hal00], mesure de consistance [TV72], ...)

Parmi les méthodes de sélection d'attributs dans un contexte non supervisé, nous nous sommes particulièrement intéressée au score de la Variance et au score Laplacien qui sont parmi les plus utilisés.

2.3.1 Score de la Variance

Le score de la Variance est le score le plus simple pour évaluer un attribut dans un contexte non supervisé. Il consiste à examiner la dispersion des données projetées sur chaque vecteur attribut. Un attribut est considéré comme pertinent lorsque les données projetées sur son axe sont les plus dispersées possible.

Soit la matrice X $(n \times d)$ de données définie par l'équation (1.1) de la section 1.2 :

$$X = \begin{bmatrix} x_{11} & \dots & x_{1r} & \dots & x_{1d} \\ \dots & \dots & \dots & \dots & \\ x_{i1} & \dots & x_{ir} & \dots & x_{id} \\ \dots & \dots & \dots & \dots & \\ x_{n1} & \dots & x_{nr} & \dots & x_{nd} \end{bmatrix} \tag{2.1}$$

En supposant que x est une variable aléatoire respectant une loi de distribution Gaussienne et que les réalisations sont équiprobables, le score de la variance SV_r est défini pour chaque attribut f_r, r=1...d, comme suit :

$$SV_r = \frac{1}{n}\sum_{i=1}^{n}(x_{ir} - \mu_r)^2 \tag{2.2}$$

où μ_r est la moyenne de l'ensemble de données sur l'attribut considéré f_r définie par :

$$\mu_r = \frac{\sum_{i=1}^{n} x_{ir}}{n} \tag{2.3}$$

Les attributs sont ordonnés par ordre décroissant de leurs score SV_r afin de sélectionner les premiers attributs les plus pertinents.

Ci-dessous un exemple permettant d'illustrer le score de la variance.

Exemple :

Soit l'ensemble composé de 4 données $\mathscr{X} = \{x_1, x_2, x_3, x_4\}$ caractérisées par 3 attributs f_1, f_2 et f_3 ($n = 4$, $d = 3$) :

$$X = \begin{bmatrix} 0.5 & 5 & 1 \\ 5 & 1 & 2 \\ 5 & 0.5 & 2 \\ 1 & 6 & 1 \end{bmatrix} \tag{2.4}$$

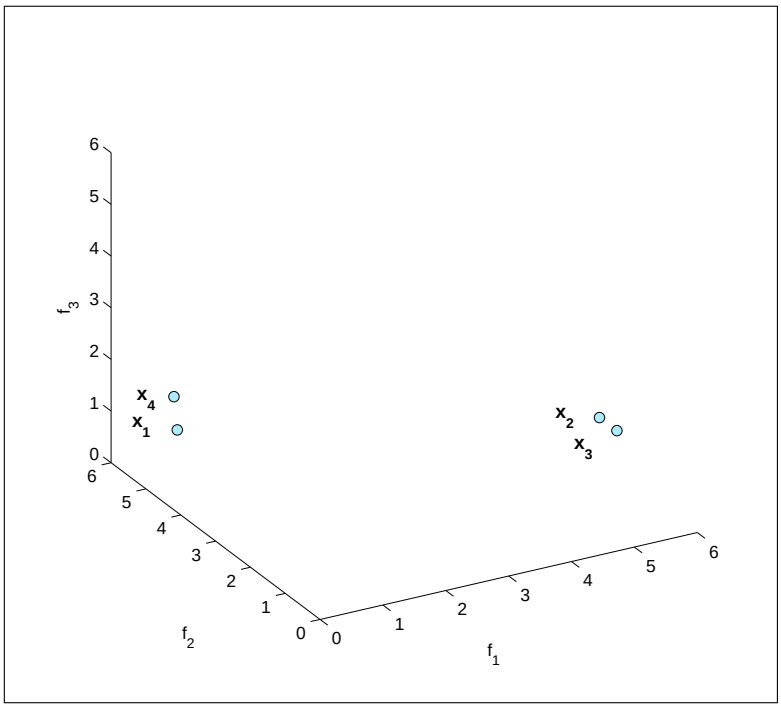


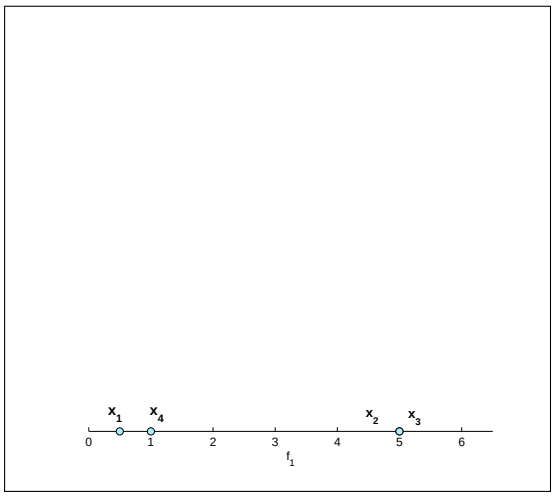
Figure 2.4 : Représentation des données dans l'espace $\mathbb{R}^3$.

La représentation de ces données par des points dans l'espace $\mathbb{R}^3$ est donnée par la figure 2.4.

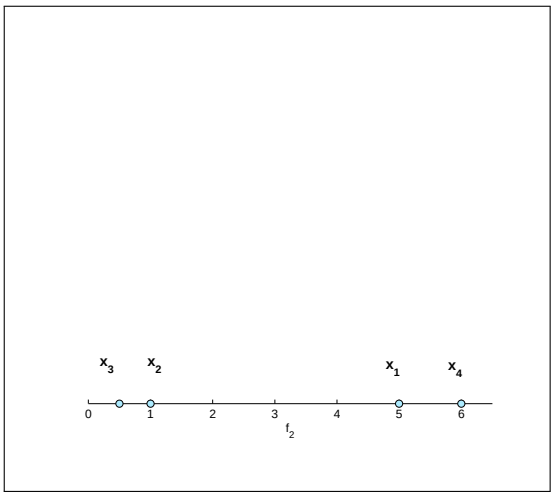
En examinant la projection des données sur l'attribut f_1 représentée par la figure 2.5(a), nous observons que les points x_2 et x_3 sont confondus. De même, lorsque ces données sont projetées sur l'attribut f_3 (cf. figure 2.5(c)), les points x_1 et x_4, d'une part et les points x_2 et x_3 d'autre part sont confondus. Par contre, la projection de ces données sur l'attribut f_2 donne lieu à des points dispersés (cf. figure 2.5(b)).

Ainsi, en utilisant le score de la variance de l'équation (2.2), l'attribut f_1 a un score $SV_1 = 4.5468$, l'attribut f_2 a un score $SV_2 = 5.7969$ et l'attribut f_3 a un score $SV_3 = 0.25$. L'attribut f_2 ayant le score le plus élevé est alors le plus pertinent. Les attributs sont ainsi classés par ordre de pertinence selon leur score SV : f_2, f_1, f_3.

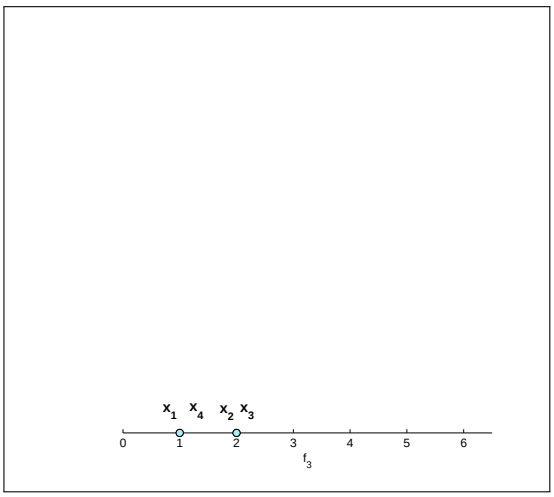
Ce score s'appuie sur une hypothèse forte de distribution Gaussienne sur la réalisation des données. Dans le paragraphe suivant, nous allons voir comment mesurer la dispersion des données projetées sur un vecteur d'attribut en relevant cette hypothèse.



(a) Projection sur l'attribut f_1



(b) Projection sur l'attribut f_2



(c) Projection sur l'attribut f_3

Figure 2.5 : La projection des données sur les différents attributs

2.3.2 Score de la Variance sous-jacente au graphe de similarité

En considérant que x est une variable aléatoire, la variance d'un attribut f_r peut alors être exprimée comme suit :

$$var(f_r) = \sum_{i=1}^{n} (x_{ir} - \overline{f}_r)^2 p_i \tag{2.5}$$

où p_i est la densité de probabilité que x_i constitue la réalisation de x ($p_i = p(x = x_i)$) et $\overline{f}_r$ est l'espérance ou la moyenne pondérée des données projetées sur l'attribut f_r. Elle correspond à

la moyenne des valeurs possibles de x_{ir} pondérées par les probabilités associées à ces valeurs :

$$\overline{f}_r = \sum_{i=1}^{n} x_{ir} p_i \tag{2.6}$$

Le score de la variance de l'équation (2.2) suppose que les réalisations des différentes données sont équiprobables, à savoir $p_i = \frac{1}{n} \forall i$. Il peut être intéressant d'affecter une densité de probabilité spécifique à chaque donnée selon sa densité locale. Cette densité va pouvoir être estimée par le biais d'un graphe de similarité.

En représentant les données par un graphe de similarité (cf. section 1.4), le degré d_i (défini équation (1.15)) du noeud s_i associé à la donnée x_i peut être considéré comme une densité au voisinage de x_i. On peut alors l'utiliser pour estimer la densité de probabilité p_i :

$$p_i = \frac{d_i}{\sum_{i=1}^{n} d_i} \tag{2.7}$$

La variance d'un attribut f_r de l'équation (2.5) sous-jacente au graphe de similarité peut alors être exprimée comme suit :

$$var(f_r) = \sum_{i=1}^{n} (x_{ir} - \overline{f}_r)^2 \frac{d_i}{\sum_{i=1}^{n} d_i} \tag{2.8}$$

et la moyenne pondérée de l'equation (2.6) sera :

$$\overline{f}_r = \frac{\sum_{i=1}^{n} x_{ir} d_i}{\sum_{i=1}^{n} d_i} \tag{2.9}$$

En définissant le vecteur $\mathbf{1}$ ($n \times 1$) dont les coordonnées sont des 1, $\overline{f_r}$ peut être représentée sous forme vectorielle :

$$\overline{f_r} = \frac{f_r^T D \mathbf{1}}{\mathbf{1}^T D \mathbf{1}} \tag{2.10}$$

avec D la matrice des degrés décrite par l'équation (1.16)

Soit $\tilde{x}_{ir} = x_{ir} - \overline{f_r}$, les coordonnées du vecteur f_r centrées sur la moyenne, on a alors

$$\tilde{f}_r = f_r - \frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1} \tag{2.11}$$

$var(f_r)$ de l'équation 2.8 peut être écrit vectoriellement comme suit :

$$\begin{aligned} var(f_r) &= \sum_{i=1}^{n} \tilde{x}_{ir}^2 d_i \\ var(f_r) &= \tilde{f}_r{}^T D \tilde{f}_r \end{aligned} \tag{2.12}$$

Exemple :

En reprenant l'exemple des données représentées par la matrice (2.4). Nous calculons le score de la variance sous-jacente au graphe pour les 3 attributs en présence (f_1, f_2 et f_3). Nous obtenons ainsi les scores suivants : $var(f_1) = 34.9597$, $var(f_2) = 44.5243$ et $var(f_3) = 1.9221$.
Les attributs sont classés par ordre de pertinence de leurs scores *var* : f_2, f_1, f_3. Ce classement est ici identique à celui obtenu par le score de la variance.

Il est vrai que le score de la variance permet de sélectionner les attributs sur lesquels la projection des données présente la plus grande dispersion. Cependant, rien ne garantit que les attributs sélectionnés permettent de discriminer les données de différentes classes. C'est pour cette raison, qu'au lieu de mesurer les propriétés globales des données à travers la dispersion selon chaque vecteur attribut, He et al [HCN05] proposent d'examiner les propriétés locales de ces données en utilisant le score Laplacien.

2.3.3 Score Laplacien

L'hypothèse sous-jacente au score Laplacien est que la structure des données dans l'espace d'attributs d'entrée est localement préservée dans l'espace d'attributs de sortie. En représentant cette structure par la notion de graphe et de similarité, des données similaires dans l'espace d'entrée doivent alors aussi l'être quand elles sont projetées sur un vecteur attribut pertinent.
Dans un contexte non supervisé, He et al. [HCN05] supposent que les données, initialement similaires dans l'espace d'attributs d'entrée, doivent être les plus proches possible, une fois projetées sur le vecteur attribut examiné. Ce principe est formalisé par le score SNL_r basé sur la similarité w_{ij} décrite par l'équation (1.8) et défini par :

$$SNL_r = \sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2 w_{ij} \tag{2.13}$$

Le score SNL_r tient compte de la structure du graphe représentée par la similarité (w_{ij}) entre les données dans l'espace d'entrée et de la distance ($(x_{ir}-x_{jr})^2$) entre les données projetées sur le vecteur attribut considéré . Le respect de la structure du graphe implique que les données similaires (w_{ij} proche de 1) dans l'espace d'entrée, doivent être proches une fois projetées sur l'attribut considéré ($(x_{ir}-x_{jr})^2$ proche de 0). Il s'agit donc de minimiser le score SNL_r.
Pour une meilleure lisibilité, le score SNL_r de l'équation (2.13) peut être réécrit sous la forme vectorielle comme suit :

$$\begin{aligned}
SNL_r &= \sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2 w_{ij} \\
SNL_r &= \sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}^2+x_{jr}^2-2x_{ir}x_{jr})w_{ij} \\
SNL_r &= 2\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ir}^2 w_{ij} - 2\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ir}w_{ij}x_{jr} \\
SNL_r &= 2\sum_{i=1}^{n}x_{ir}^2 d_i - 2\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ir}w_{ij}x_{jr} \\
SNL_r &= 2f_r^T D f_r - 2f_r^T W f_r \\
SNL_r &= 2f_r^T L f_r
\end{aligned} \tag{2.14}$$

Où W est la matrice de similarité et D est la matrice des degrés définies respectivement par les équations (1.14) et (1.16). L $(n \times n)$ est la matrice Laplacienne définie par :

$$L = D - W \tag{2.15}$$

En normalisant le score SNL_r par la variance sous jacente à un graphe définie par l'équation (2.8), He et al. [HCN05] proposent de calculer le score Laplacien SL_r pour un attribut f_r comme suit :

$$SL_r = \frac{SNL_r}{var(f_r)} \tag{2.16}$$

En remplaçant le numérateur et le dénominateur du score Laplacien de l'équation (2.16) par leurs écritures vectorielles correspondantes représentées respectivement par les équations (2.14) et (2.12), ce score peut être écrit comme suit :

$$SL_r = 2\frac{f_r^T L f_r}{\tilde{f}_r{}^T D \tilde{f}_r} \tag{2.17}$$

Comme $f_r^T L f_r = \tilde{f}_r{}^T L \tilde{f}_r$, le score Laplacien de l'équation (2.17) peut être écrit comme suit :

$$SL_r = 2\frac{\tilde{f}_r{}^T L \tilde{f}_r}{\tilde{f}_r{}^T D \tilde{f}_r} \tag{2.18}$$

En effet en utilisant l'équation (2.11) :

$$\begin{aligned} \tilde{f}_r{}^T L \tilde{f}_r &= (f_r - \frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1})^T L (f_r - \frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1}) \\ \tilde{f}_r{}^T L \tilde{f}_r &= (f_r^T L f_r) - ((\frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1})^T L f_r) - (f_r^T L (\frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1})) + ((\frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1})^T L (\frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1})) \end{aligned} \tag{2.19}$$

Soit $o = \frac{f_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}$, valeure réelle. Nous pouvons alors écrire :

$$\tilde{f_r}^T L \tilde{f_r} = f_r^T L f_r - o\mathbf{1}^T L f_r - f_r^T Lo\mathbf{1} + o\mathbf{1}^T Lo\mathbf{1} \tag{2.20}$$

Nous pouvons facilement vérifier que $\mathbf{1}^T L = L\mathbf{1} = \mathbf{0}$, où $\mathbf{0}$ $(n \times 1)$ est le vecteur dont les coordonnées sont des 0.

$\tilde{f_r}^T L \tilde{f_r}$ est alors égal à $f_r^T L f_r$:

$$\tilde{f_r}^T L \tilde{f_r} = f_r^T L f_r \tag{2.21}$$

L'équation (2.18) montre que le score Laplacien dépend de la matrice Laplacienne L et de la matrice des degrés D calculées à partir de la matrice W de similarité des données.

Les attributs sont classés par ordre croissant de leur score Laplacien SL_r pour sélectionner les attributs les plus pertinents.

Exemple : Pour illustrer la sélection d'attributs en utilisant le score Laplacien, nous allons reprendre l'exemple de données représentées par la matrice (2.4). En fixant la paramètre σ à 2, nous pouvons ainsi construire la matrice de similarité W entre ces données par l'équation (1.8) définie par :

$$W = \begin{bmatrix} 1 & 0.0095 & 0.0055 & 0.8553 \\ 0.0095 & 1 & 0.9692 & 0.0052 \\ 0.0055 & 0.9692 & 1 & 0.0027 \\ 0.8553 & 0.0052 & 0.0027 & 1 \end{bmatrix}$$

En utilisant le score Laplacien de l'équation (2.17), l'attribut f_1 obtient un score $SL_1 = 0.0185$, l'attribut f_2 a un score $SL_2 = 0.0354$ et l'attribut f_3 a un score $SL_3 = 0.012$. L'attribut f_3 ayant le score le plus bas est alors le plus pertinent pour la sélection. Il permet de bien représenter la

structure des données telles qu'elles se présentent dans leur espace d'entrée représentée par la figure 2.4.

Le classement des attributs par ordre de pertinence selon leur score *SL* est alors : f_3, f_1, f_2. Ce classement diffère du classement obtenu par le score de Variance car il tient compte de la similarité entre données.

Zhao et al. effectuent une analyse fine du score Laplacien en énumérant les principales propriétés mathématiques de la matrice Laplacienne *L* [ZL07c].

Ainsi, à partir du score Laplacien SL_r, Zhao et al. génèrent d'autres scores qui découlent de l'analyse spectrale du graphe de similarité.

Nous avons choisi de garder le score Laplacien classique représenté par l'équation (2.18) qui est le plus couramment utilisé.

2.4 Sélection supervisée

Les algorithmes de sélection supervisée évaluent la pertinence d'un attribut suivant sa corrélation avec les labels des classes de données. Une procédure de sélection qui opère dans ce contexte consiste à examiner la corrélation entre les données projetées sur chaque vecteur d'attribut et leurs labels de classes. Il s'agit de sélectionner les attributs sur lesquels les différentes classes sont les plus éloignées les unes des autres et les plus compactes possible. Le score de Fisher est l'un des scores les plus utilisés pour évaluer la pertinence de chacun des attributs dans ce contexte [Bis96].

2.4.1 Score de Fisher

En considérant les coordonnées des données projetées sur l'attribut f_r, chaque classe ω (ω=1,...,c), d'effectif n_ω est caractérisée par sa moyenne $\mu_{\omega r}$ et sa variance $\sigma^2_{\omega r}$ définies par :

$$\mu_{\omega r} = \frac{\sum_{i \in \omega} x_{ir}}{n_\omega} \tag{2.22}$$

$$\sigma^2_{\omega r} = \frac{\sum_{i \in \omega} (x_{ir} - \mu_{\omega r})^2}{n_\omega} \tag{2.23}$$

Le score de Fisher utilisé pour évaluer l'attribut f_r est alors défini par :

$$SF_r = \frac{\sum_{\omega=1}^{c} n_\omega (\mu_{\omega r} - \mu_r)^2}{\sum_{\omega=1}^{c} n_\omega \sigma^2_{\omega r}}. \tag{2.24}$$

Une valeur élevée du numérateur de l'équation (2.24) révèle que les différentes classes sont les plus éloignées possible, tandis qu'une valeur faible du dénominateur montre que les classes sont compactes. Plus le score SF_r est élevé, plus l'attribut f_r est considéré comme étant pertinent. Les différents attributs sont ainsi classés par ordre décroissant de leur score de Fisher afin de sélectionner les attributs les plus pertinents.

Exemple : Reprenons l'exemple de données représentées par la matrice (2.4). Supposons qu'en plus de la matrice de données X, nous disposons du vecteur Y de labels de classes des données défini comme suit :

$$Y = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

Ces données sont divisées en deux classes à savoir la classe 1 contenant les données x_1 et x_4 et la classe 2 contenant les données x_2 et x_3. Elles sont représentées dans l'espace $\mathbb{R}^3$ par la figure 2.6.

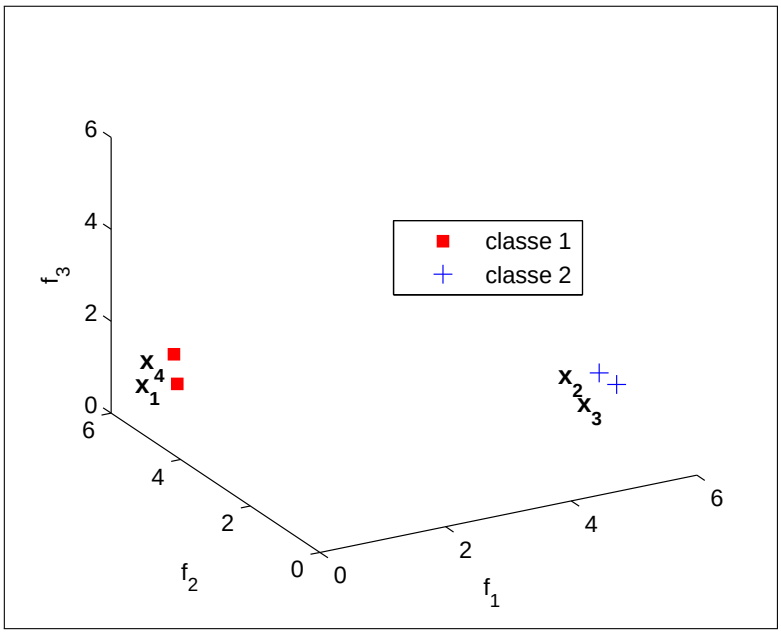


Figure 2.6 : Représentation des données labellisées dans l'espace $\mathbb{R}^3$.

Score de la Variance	f_2, f_1, f_3
Score Laplacien	f_3, f_1, f_2
Score de Fisher	f_1, f_2, f_3

Tableau 2.1 : Les attributs ordonnées selon différents algorithmes de sélection

En utilisant le score de Fisher de l'équation (2.24), l'attribut f_1 a un score $SF_1 = 144.5$, l'attribut f_2 a un score $SF_2 = 36.1$ et le score de l'attribut f_3 a été fixé à 0 car le dénominateur est égal à 0. L'attribut f_1 ayant le score le plus élevé est alors le plus pertinent pour la sélection. Il permet la meilleure représentation des classes en présence.

Les attributs sont ainsi classés par ordre de pertinence selon leur score F : f_1, f_2, f_3.

Le tableau 2.1 reprend les différents résultats de classement des attributs de l'exemple des données de la matrice (2.4) par les différents algorithmes de sélection utilisés (Score de la variance, Score Laplacien, Score de Fisher).

Nous pouvons ainsi remarquer que sur le même exemple, les trois différents scores de sélection débouchent sur trois résultats différents. Cet exemple met en évidence que la procédure de sélection est très dépendante de la fonction d'évaluation utilisée ainsi que du contexte d'apprentissage.

Le score de sélection supervisé énoncé ci-dessus ne prend pas en considération la structure des

données représentée par le graphe de similarité (voir section 1.4). C'est pour cette raison que nous allons introduire dans le paragraphe suivant le score Laplacien supervisé.

2.4.2 Score Laplacien supervisé

Le score Laplacien de l'équation (2.18) peut être utilisé pour évaluer l'attribut f_r dans un contexte supervisé. Dans ce contexte, la matrice de similarité W utilisée est définie en fonction du vecteur Y des labels de classes de données.
Il existe plusieurs façons de définir la matrice de similarité dans un contexte supervisé [KKM03]. He et al. la définissent comme suit [HCN05] :

$$w_{ij} = \begin{cases} \frac{1}{n_\omega} & \text{si } y_i = y_j = \omega \\ 0 & \text{sinon,} \end{cases} . \tag{2.25}$$

Les auteurs démontrent alors que, dans un contexte supervisé, en utilisant la matrice de similarité de l'équation (2.25), le score Laplacien supervisé devient [HCN05] :

$$SL_r = \frac{1}{1 + SF_r} \tag{2.26}$$

où SF_r est le score supervisé de Fisher définie par l'équation (2.24).

2.4.3 Information mutuelle

L'information mutuelle, introduite par Shanon et al. [SW49] [CT91], mesure la dépendance de deux variables aléatoires discrètes X et Y de densité de probabilité $p(x)$ et $p(y)$. Elle est définie par :

$$MI(X/Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \ log \frac{p(x,y)}{p(x).p(y)} \tag{2.27}$$

où $p(x,y)$ est la densité de probabilité conjointe des deux variables X et Y.
L'information mutuelle est nulle si les variables sont indépendantes et croit lorsque la dépen-

dance augmente.

En divisant l'information mutuelle définie par l'équation (2.27) par la valeur maximale des entropies H(X) et H(Y), on obtient l'information mutuelle normalisée définie par :

$$NMI(X/Y) = \frac{\sum_{x\in X}\sum_{y\in Y} p(x,y)\ log\frac{p(x,y)}{p(x).p(y)}}{max(H(X),H(Y))} \tag{2.28}$$

avec

$$H(X) = -\sum_{x\in X} p(x)\ log\ p(x) \tag{2.29}$$

L'information mutuelle normalisée de l'équation (2.28) est ainsi utilisée pour mesurer la dépendance entre un attribut f_r et le vecteur Y de labels des classes des données.

Les coordonnées du vecteur Y sont discrètes, tandis que celles du vecteur attribut f_r sont continues. C'est pour cette raison qu'un algorithme de classification des données projetées sur le vecteur d'attribut f_r est généralement utilisé. Il permet de transformer le vecteur attribut f_r en un vecteur de labels de classes estimés $\hat{Y}_r$, pour ainsi calculer $NMI(\hat{Y}_r/Y)$ [JMGW10].

Dans un contexte supervisé, les attributs sont ainsi classés par ordre décroissant de leur score NMI afin de sélectionner les attributs les plus pertinents.

Exemple

Pour illustrer la sélection d'attributs dans un contexte supervisé en utilisant l'information mutuelle, nous allons considérer l'exemple de données labellisées représentées dans l'espace $\mathbb{R}^2$ par la figure 2.7(a). Les points appartenant à la classe 1 (noté ω_1) sont représentés par des carrés rouges et les points de la classe 2 (noté ω_2) sont représentés par des croix bleues.

$$\omega_1 = \{x_i \in \mathscr{X}/y_i = 1\} \tag{2.30}$$

$$\omega_2 = \{x_i \in \mathscr{X} / y_i = 2\} \tag{2.31}$$

En utilisant un algorithme de classification des données projetées sur le vecteur d'attribut f_1, les données sont ainsi classées en deux classes (voir figure 2.7(b))

$$\widehat{\omega}_1 = \{x_i \in \mathscr{X} / \widehat{y}_{1i} = 1\} \tag{2.32}$$

$$\widehat{\omega}_2 = \{x_i \in \mathscr{X} / \widehat{y}_{1i} = 2\} \tag{2.33}$$

où $\widehat{y}_{1i}$ est le label estimé da la donnée x_i en tenant compte de l'attribut f_1.

Nous pouvons remarquer sur cette figure que des données de labels différents ont été affectées à la même classe ($\exists x_i \in X / y_i \neq \widehat{y}_{1i}$).

Cependant, en projetant ces données sur le vecteur d'attribut f_2 et en utilisant le même classifieur qui opère en utilisant cette fois f_2, les données sont classées en deux classes (voir figure 2.7(c)).

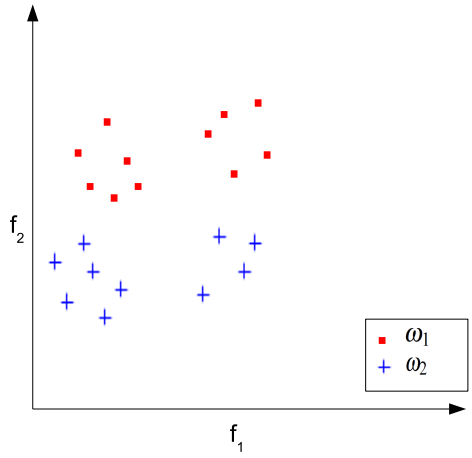
$$\widehat{\omega}_1 = \{x_i \in \mathscr{X} / \widehat{y}_{2i} = 1\} \tag{2.34}$$

$$\widehat{\omega}_2 = \{x_i \in \mathscr{X} / \widehat{y}_{2i} = 2\} \tag{2.35}$$

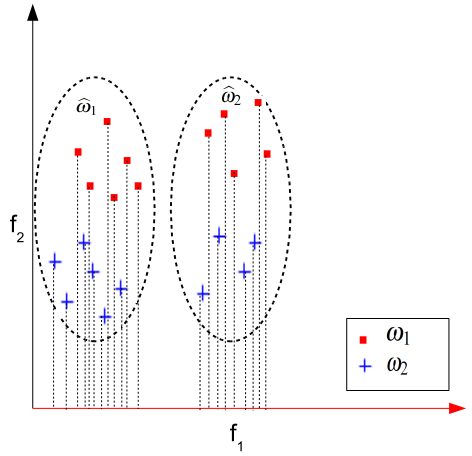
Dans cette figure, le résultat de classification est cohérent avec le label des données ($\forall x_i \in X / y_i = \widehat{y}_{2i}$)

Ainsi, l'information mutuelle entre le vecteur $\widehat{Y}_2$ et le vecteur Y est supérieure à l'information mutuelle entre le vecteur $\widehat{Y}_1$ et le vecteur Y (NMI($\widehat{Y}_2$/Y)>NMI($\widehat{Y}_1$/Y)).

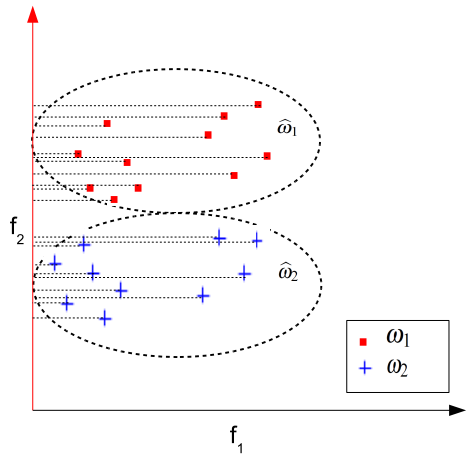
Comme l'information mutuelle normalisée utilise les labels estimés, elle ne peut être utilisée que par une procédure de sélection d'attributs de type "wrapper". Comme pour toutes les méthodes de ce type, la sélection des attributs dépend de la procédure de décision utilisée. Nous avons tenu toutefois à la présenter car elle est utilisée par un critère de sélection semi-supervisée abordé ultérieurement.



(a) Représentation des données labellisées dans $\mathbb{R}^2$



(b) Résultats de classification (classes entourées par des traits en pointillé) en tenant compte de la projection sur le vecteur d'attribut f_1



(c) Résultats de classification (classes entourées par des traits en pointillé) en tenant compte de la projection sur le vecteur d'attribut f_2

Figure 2.7 : Comparaison de l'information mutuelle.

2.5 Sélection semi-supervisée avec labels

Dans le contexte d'apprentissage semi-supervisé avec labels, le nombre de données labellisées est trop faible pour apporter suffisamment d'informations nécessaires à la sélection supervisée d'attributs. Un algorithme de sélection non supervisée peut être alors envisagé mais il ignore l'information fournie par les labels disponibles. Il est donc préférable que la pertinence des attributs soit évaluée en tenant compte à la fois des données labellisées et non labellisées. C'est pour cette raison que des approches récentes de sélection d'attributs dans un contexte semi-supervisé ont été développées.

Rappelons que dans ce contexte, l'ensemble de données $\mathscr{X}$ est divisé en deux sous-ensembles $\mathscr{X} = \{\mathscr{X}_l \cup \mathscr{X}_u\}$. $\mathscr{X}_l$ est le sous-ensemble des l données labellisées pour lequel le vecteur Y_l est disponible. $\mathscr{X}_u$ est le sous-ensemble des u données non labellisées pour lequel aucune information n'est disponible. Les nombres l et u respectent l'équation $l+u=n$.

Dans ce contexte, Zhao et al. proposent de coupler le score Laplacien et l'information mutuelle normalisée (NMI) pour introduire un score de sélection semi-supervisée d'attributs [ZL07a] [ZL07b]. Ce score est basé sur une comparaison entre les labels des données issus du sous-ensemble $\mathscr{X}_l$ et leurs labels estimés en utilisant un algorithme de classification opérant avec l'attribut considéré. Ce sore SM_r est défini comme suit :

$$\begin{aligned} SM_r &= \alpha \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2 w_{ij}}{var(f_r)} + (1-\alpha)\left(1 - NMI\left(\widehat{Y}_{lr}/Y_l\right)\right) \\ SM_r &= \alpha SL_r + (1-\alpha)\left(1 - NMI\left(\widehat{Y}_{lr}/Y_l\right)\right) \end{aligned} \tag{2.36}$$

où $\widehat{Y}_{lr}$ est le vecteur indicateur de classe des données généré par un algorithme de classification opérant avec l'attribut f_r [ZL07a].

Ce score est formé de deux parties pondérées par le terme α :

– Le premier terme de l'équation (2.36) révèle la structure locale des données en calculant le score Laplacien de l'attribut f_r (SL_r à minimiser). Il permet donc de s'assurer que les données projetées sur le vecteur attribut f_r respectent la similarité des données estimée

dans l'espace de départ.

– Le second terme de l'équation (2.36) estime l'erreur d'estimation de $\widehat{Y}_{lr}$ en utilisant les données labellisées ($(1 - NMI\left(\widehat{Y}_{lr}/Y_l\right)$ à minimiser). Il permet de s'assurer que les données projetées sur l'attribut considéré f_r forment des classes en accord avec l'information de labels fournie par l'expert.

Le terme α de l'équation (2.36) est un paramètre de régularisation pour pondérer la contribution des deux termes de cette équation. Zhao et al. favorisent la contribution de la cohérence avec les données labellisées en fixant ce paramètre à 0.1.

Les attributs sont classés par ordre croissant de leur score SM_r pour sélectionner les plus pertinents.

Ce scores est utilisé par un algorithme de sélection de type "embedded". Il tient compte des données non labellisées à travers un critère de type "filter" basé sur le score Laplacien. De même, il tient compte des données non labellisées à travers un critère type "wrapper" basé sur l'information mutuelle normalisée (NMI).

Il est vrai que ce score permet de prendre en considération à la fois les données labellisées et les données non labellisées. Cependant, les attributs sélectionnés en utilisant l'équation (2.36) sont très dépendants de la règle de décision retenue par le classifieur utilisé.

Exemple :

Soit l'ensemble des données $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ représentées par la matrice $X(6 \times 2)$

suivante :

$$X=\begin{bmatrix} 1.5 & 1.5 \\ 3.5 & 3 \\ 4 & 2.5 \\ 3 & 1.2 \\ 1 & 1 \\ 3 & 2.5 \end{bmatrix}. \tag{2.37}$$

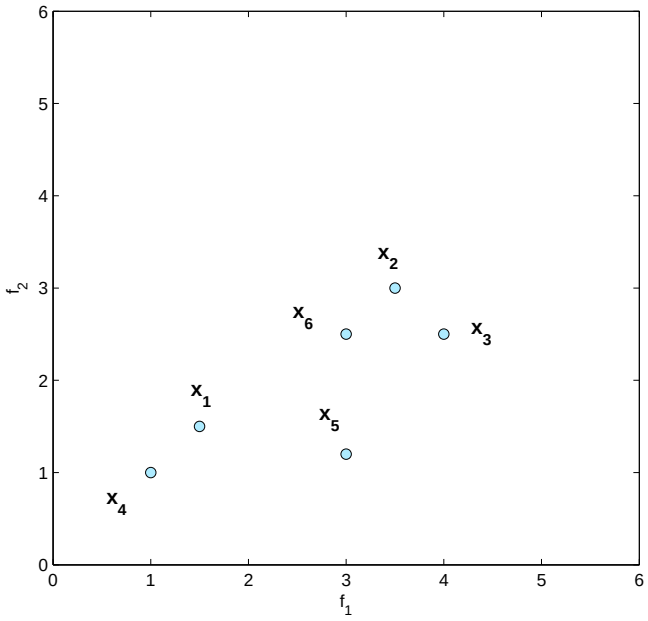


Figure 2.8 : Représentation des données dans l'espace $\mathbb{R}^2$.

Dans un premier cas, supposons que nous ne disposons d'aucune information sur les labels des données représentées dans l'espace $\mathbb{R}^2$ par la figure 2.8.

Dans ce contexte d'apprentissage non supervisé, le score Laplacien (cf. équation (2.17)) de l'attribut f_1 est $SL_1 = 0.6767$, tandis que celui de l'attribut f_2 est $SL_2 = 0.7351$. L'attribut f_1 ayant le score le plus bas est alors le plus pertinent pour la sélection en utilisant le score Laplacien dans un contexte non supervisé.

Dans un contexte d'apprentissage semi-supervisé, nous supposons que l'expert a assigné les

données x_4, x_5 et x_6 à deux classes différentes, tandis que, nous ne disposons d'aucune information sur la classe d'appartenance des données x_1, x_2 et x_3. $\mathscr{X}_l = \{x_4, x_5, x_6\}$ et $\mathscr{X}_u = \{x_1, x_2, x_3\}$.

Les matrices sont alors :

$$X_u = \begin{bmatrix} 1.5 & 1.5 \\ 3.5 & 3 \\ 4 & 2.5 \end{bmatrix}. \tag{2.38}$$

$$X_l = \begin{bmatrix} 3 & 1.2 \\ 1 & 1 \\ 3 & 2.5 \end{bmatrix}. \tag{2.39}$$

$$Y_l = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}. \tag{2.40}$$

Les données x_4 et x_5 sont assignées à la classe 1, la donnée x_6 est assignée la classe 2 et les données x_1, x_2 et x_3 ne sont pas labellisées.

Les données avec les informations sur leur label sont représentées dans l'espace $\mathbb{R}^2$ par la figure 2.9.

Un algorithme de classification opérant soit sur f_1, soit sur f_2 a estimé les vecteurs de labels suivants :

$$\widehat{Y}_{l1} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}. \tag{2.41}$$

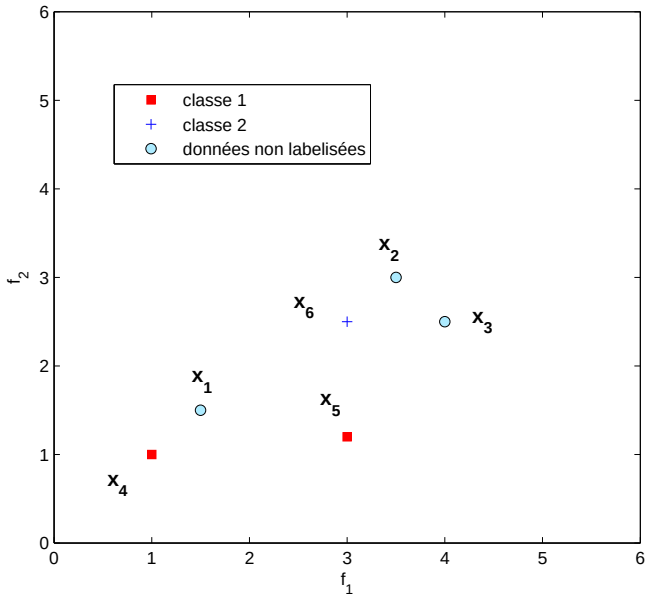


Figure 2.9 : Représentation des données avec une information incomplète sur leur label dans l'espace $\mathbb{R}^2$.

$$\widehat{Y}_{l2} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}. \tag{2.42}$$

En utilisant le score Laplacien couplé avec NMI de l'équation (2.36), l'attribut f_1 a un score $SM_1 = 0.7211$ et l'attribut f_2 a un score $SM_2 = 0.0735$. L'attribut f_2 ayant le score le plus bas est alors le plus pertinent pour la sélection en utilisant le score Laplacien couplé avec NMI dans un contexte semi-supervisé.

En effet, concernant la contribution des données labellisées, l'attribut f_1 a un NMI=0.274, puisque les données labellisées sont mal classifiées en utilisant cet attribut (des données ayant le même label de classe sont assignées à deux classes estimées différentes). Par contre, le valeur NMI de l'attribut f_2 atteint la valeur maximale 1, puisque toutes les données labellisées sont bien classifiées en utilisant cet attribut.

Cet exemple montre alors que le score semi-supervisé *SM* fournit des classements d'attributs différents du score Laplacien non supervisé.

Dans le contexte semi-supervisé, on peut également avoir recours à une connaissance de type comparaison entre deux données. Ces connaissances, nommées avec le terme contraintes ont été récemment intégrées dans les scores d'attributs [ZCZ08] [ZL07a].

2.6 Scores de contraintes

Zhang et al. ont proposé un score d'attributs qui utilise un sous-ensemble de contraintes must-link ($\mathscr{M}$) et cannot-link ($\mathscr{C}$) [ZCZ08].

Ces contraintes étant représentées par des graphes must-link $G^{\mathscr{M}}$ et cannot-link $G^{\mathscr{C}}$, l'idée sous-jacente est de sélectionner les attributs qui permettent de préserver la structure de ces deux graphes.

Ainsi, pour un attribut f_r, les données contraintes par must-link doivent être les plus proches possible, une fois projetées sur ce vecteur d'attribut. Ceci revient donc à minimiser le terme :

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_{ir}-x_{jr}\right)^2 w_{ij}^{\mathscr{M}}. \tag{2.43}$$

où $w_{ij}^{\mathscr{M}}$ est représentée par l'équation (1.30)

Par contre, les données contraintes par cannot-link doivent être les plus éloignées possible, une fois projetées sur ce vecteur d'attribut. Ceci revient donc à maximiser le terme :

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_{ir}-x_{jr}\right)^2 w_{ij}^{\mathscr{C}}. \tag{2.44}$$

où $w_{ij}^{\mathscr{C}}$ est représentée par l'équation (1.31)

En utilisant la même démonstration que celle de l'équation (2.14), les termes des équations (2.43) et (2.44) peuvent être écrits sous forme vectorielle comme suit :

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_{ir}-x_{jr}\right)^2 w_{ij}^{\mathscr{M}} = f_r^T L^{\mathscr{M}} f_r, \tag{2.45}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_{ir}-x_{jr}\right)^2 w_{ij}^{\mathscr{C}} = f_r^T L^{\mathscr{C}} f_r, \tag{2.46}$$

où $L^{\mathscr{M}} = D^{\mathscr{M}} - W^{\mathscr{M}}$ et $L^{\mathscr{C}} = D^{\mathscr{C}} - C^{\mathscr{C}}$, $D^{\mathscr{M}}$ et $D^{\mathscr{C}}$ étant les matrices des degrés définies par :

$$D_{ii}^{\mathscr{M}} = \sum_{j=1}^{n} w_{ij}^{\mathscr{M}} \tag{2.47}$$

$$D_{ii}^{\mathscr{C}} = \sum_{j=1}^{n} w_{ij}^{\mathscr{C}} \tag{2.48}$$

En combinant les termes (2.43) et (2.44) de deux façons différentes (rapport ou soustraction), Zhang et al. proposent les deux scores de contraintes SC_r^1 et SC_r^2 [ZCZ08] :

$$SC_r^1 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_{ir}-x_{jr}\right)^2 w_{ij}^{\mathscr{M}}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_{ir}-x_{jr}\right)^2 w_{ij}^{\mathscr{C}}} = \frac{f_r^T L^{\mathscr{M}} f_r}{f_r^T L^{\mathscr{C}} f_r} \tag{2.49}$$

$$\begin{aligned} SC_r^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_{ir}-x_{jr}\right)^2 w_{ij}^{\mathscr{M}} - \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_{ir}-x_{jr}\right)^2 w_{ij}^{\mathscr{C}} \\ SC_r^2 &= f_r^T L^{\mathscr{M}} f_r - \lambda f_r^T L^{\mathscr{C}} f_r, \end{aligned} \tag{2.50}$$

où λ est un paramètre de régularisation utilisé pour ajuster la contribution relative de chacun des deux termes de l'équation (2.50).

Les attributs ayant les scores SC_r^1 ou SC_r^2 les plus bas sont les plus pertinents pour la sélection. Zhang et al. ont expérimentalement démontré que les attributs sélectionnés par SC_r^1 et SC_r^2 ont des performances similaires lorsque le paramètre λ est bien ajusté [ZCZ08].

L'utilisation des contraintes must-link et cannot-link dans le processus de sélection permet aux attributs sélectionnés de respecter la structure des données tout en garantissant le respect de ces contraintes. Cependant, les scores SC_r^1 et SC_r^2 ne prennent en considération que les contraintes disponibles et négligent l'information apportée par les données pour lesquelles aucune contrainte n'a été précisée par l'expert.

Zhao et al. définissent un score de sélection SC_r^3 qui utilise à la fois les données et les contraintes disponibles. Ce score permettra alors de respecter les propriétés locales des données ainsi que de garantir le respect des contraintes [ZLH08].

Afin de représenter les données non labellisées, les auteurs construisent un nouveau graphe $G^{\mathscr{W}}$ appelé graphe intra-classe au sein duquel les données ayant une grande probabilité d'avoir le même label sont connectées : deux noeuds s_i et s_j sont connectés si (x_i,x_j) ou $(x_j,x_i) \in \mathscr{M}$, ou si les données x_i et x_j ne sont pas contraintes mais sont voisines dans l'espace d'entrée (en utilisant le graphe de k-voisinage).

Les arcs du graphe $G^{\mathscr{W}}$ sont pondérés en utilisant la matrice de similarité $W^{\mathscr{W}}$ $(n \times n)$ exprimée par :

$$w_{ij}^{\mathscr{W}} = \begin{cases} \gamma \text{ si } (x_i, x_j) \in \mathscr{M} \text{ ou } (x_j, x_i) \in \mathscr{M} \\ 1 \text{ si } x_i \text{ ou } x_j \text{ ne sont pas contraintes et } x_i \in KNN(x_j) \text{ ou } x_j \in KNN(x_i) \\ 0 \text{ sinon} \end{cases} \tag{2.51}$$

où $KNN(x_i)$ est l'ensemble des k-plus proches voisins de x_i et γ est une constante empiriquement fixée à 100. ([ZLH08]).

Zhao et al. introduisent un nouveau score Laplacien SC_r^3 (nommé the locality sensitive discriminant analysis score) défini comme suit :

$$SC_r^3 = \frac{\sum_{i=1}^n \sum_{j=1}^n (x_{ir} - x_{jr})^2 w_{ij}^{\mathscr{W}}}{\sum_{i=1}^n \sum_{j=1}^n (x_{ir} - x_{jr})^2 w_{ij}^{\mathscr{C}}} = \frac{f_r^T L^{\mathscr{W}} f_r}{f_r^T L^{\mathscr{C}} f_r}, \tag{2.52}$$

où $L^{\mathscr{W}} = D^{\mathscr{W}} - W^{\mathscr{W}}$, $D^{\mathscr{W}}$ étant la matrice des degrés définies par $D_{ii}^{\mathscr{W}} = \sum_{j=1}^n w_{ij}^{\mathscr{W}}$.

Le score SC_r^3 prend implicitement en considération les données non contraintes. En effet, ce score favorise les paires de données must-link en leur assignant une similarité élevée dans la matrice $W^{\mathscr{W}}$ ($\gamma = 100$) pendant que les paires des données non contraintes sont pondérées par une similarité binaire selon leur relation de voisinage.

En considérant essentiellement les contraintes must-link par la forte valeur de γ, le score SC^3 est très proche du score SC^2. Ces deux scores négligent donc les informations pouvant être apportées par les données non contraintes.

Notons qu'il existe aussi d'autres scores de sélection d'attributs basés sur l'ensemble des contraintes must-link et cannot-link.

Mercado et al. proposent d'appliquer une variante du score Laplacien introduite par Zhao et al. [ZL07c] pour l'exploration de collections de musique [MPLH10]. L'idée sous-jacente à la classification spectrale est que les k premiers vecteurs propres de la matrice Laplacienne permettent de discriminer les différentes classes. Le score est défini par :

$$\phi(f_r) = \sum_{j=2}^{k} \left(\rho(2) - \rho(\beta_j)\right) \theta_j^2 \tag{2.53}$$

où ρ est une fonction rationnelle, k est le nombre de classes, β_j sont les valeurs propres ($0 \leq \beta_1 \leq ... \leq \beta_j$) de la matrice Laplacienne L et θ_j est le cosinus de l'angle entre le vecteur propre ε_j correspondant à la valeur propre β_j et le vecteur $\widehat{f_r}$ ($\widehat{f_r} = \frac{D^{1/2} f_r}{\left\| D^{1/2} f_r \right\|}$)

Les auteurs intègrent les contraintes disponibles indirectement dans la matrice de similarité. Ils définissent alors la matrice de similarité W' comme suit :

$$W' = W + T \tag{2.54}$$

où W est la matrice de similarité basée sur la fonction gaussienne définie par l'équation (1.14) et T est la matrice de contraintes définie par :

$$t_{ij}^{\mathscr{W}} = \begin{cases} m \text{ si } \left(x_i, x_j\right) \in \mathscr{M} \text{ ou } \left(x_j, x_i\right) \in \mathscr{M} \\ -m \text{ si } \left(x_i, x_j\right) \in \mathscr{C} \text{ ou } \left(x_j, x_i\right) \in \mathscr{C} \\ 0 \text{ sinon} \end{cases} \tag{2.55}$$

où m est une constante arbitraire.

Par ailleurs, Yang et al. proposent un score de sélection basé sur l'hypothèse de marges optimales et intégrant seulement des contraintes cannot-link [YMSJ10].

Notons que nous citons juste ces deux scores existants dans la littérature sans les considérer dans notre étude. En effet, Mercado et al. [MPLH10] intègrent indirectement les contraintes must-link et cannot-link dans la construction de la matrice de similarité sans les considérer dans le score de sélection tandis que Yang et al. [YMSJ10] n'intègrent que des contraintes cannot-link dans leur score.

2.7 Sélection semi-supervisée avec contraintes

Prendre en considération les données non contraintes dans le processus de sélection permet de représenter la structure des données et rend ce processus moins sensible aux sous-ensembles de contraintes disponibles. Ceci nous a amenée alors à proposer un nouveau score de sélection semi-supervisée avec contraintes moins sensible au jeu de contraintes fourni par l'expert.

Etant donné les matrices W, $W^{\mathscr{M}}$ et $W^{\mathscr{C}}$, notre score de contrainte semi-supervisé SC_r^4 est défini comme suit [KBMH11] :

$$SC_r^4 = \frac{\tilde{f}_r^T L \tilde{f}_r}{\tilde{f}_r^T D \tilde{f}_r} . \frac{f_r^T L^{\mathscr{M}} f_r}{f_r^T L^{\mathscr{C}} f_r}. \tag{2.56}$$

Le score SC_r^4 résulte du produit des deux scores : le score Laplacien SL_r (cf. équation (2.17)) et le score de contraintes SC_r^1 défini par Zhang (cf. équation (2.49)) :

$$SC_r^4 = SL_r . SC_r^1. \tag{2.57}$$

Le score Laplacien permet de prendre en considération la structure des données pendant que le

score SC_r^1 permet de garantir le respect des contraintes must-link et cannot-link disponibles. Comme dans le cas des scores précédents de contraintes, les attributs sont ordonnés par ordre croissant de leur score SC^4 afin de sélectionner les attributs les plus pertinents.

Exemple :

Nous allons reprendre l'exemple des données représentées par la matrice (2.37), Supposons qu'en plus de ces données, nous disposons de deux sous-ensembles de contraintes must-link $\mathscr{M}$ et cannot-link $\mathscr{C}$:

$\mathscr{M} = \{(x_4, x_5)\}$,

$\mathscr{C} = \{(x_4, x_6); (x_5, x_6)\}$.

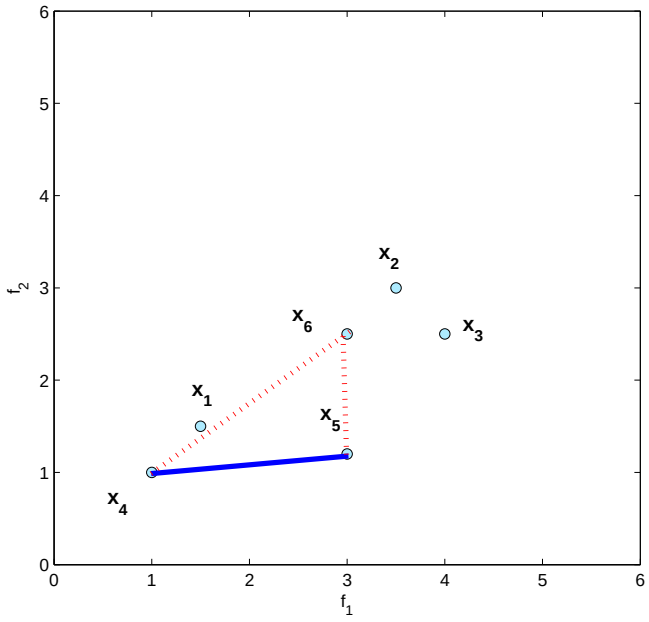


Figure 2.10 : Représentation des données avec définition de contrainte dans l'espace $\mathbb{R}^2$. La contrainte must-link est désignée par un arc gras bleu, tandis que la contrainte cannot-link est désignée par un arc en traits pointillés rouge.

La figure (2.10) illustre la représentation des données projetées dans l'espace $\mathbb{R}^2$. En utilisant les différentes scores SC_1, SC_2, SC_3 et SC_4, l'attribut f_2 est toujours sélectionné comme l'attribut le plus pertinent. Le tableau 2.2 résume les scores des deux attributs f_1 et f_2 obtenus par SC_1, SC_2, SC_3 et SC_4.

La différence entre les différents scores SC_1, SC_2, SC_3 et SC_4 sera mise en évidence dans le

	f_1	f_2
SC^1	1	0.0102
SC^2	3.6	0.354
SC^3	102.875	3.7335
SC^4	0.4391	0.0059

Tableau 2.2 : Les scores des attributs f_1 et f_2 obtenus par SC_1, SC_2, SC_3 et SC_4.

chapitre suivant.

2.8 Conclusion

Dans ce chapitre, nous nous sommes concentrée sur la sélection d'attributs par leur classement selon un score de pertinence. Cette approche très simple de type "filter" utilise un score basé sur les caractéristiques de l'ensemble des données, indépendamment de tout algorithme de classification. Le point clé réside dans la définition du score qui dépend du contexte d'apprentissage au sein duquel s'effectue la sélection d'attributs.

Dans le contexte non supervisé, le score de la Variance évalue la pertinence d'un attribut par une mesure de la dispersion des données projetées sur ce vecteur d'attribut. Un attribut est d'autant plus approprié que les projections sur ce vecteur d'attributs sont dispersées. Le score Laplacien intègre quant à lui la similarité Gaussienne entre données estimée dans l'espace d'attributs d'entrée. En s'appuyant sur ce graphe de similarité entre données, ce score privilégie les attributs pour lesquels les projections des données similaires sont proches. Un attribut est d'autant plus pertinent que la structure locale des données est conservée après la projection sur ce vecteur d'attribut.

Dans le contexte supervisé où les classes sont complètement définies, le score de Fisher met en évidence les attributs pour lesquels les classes sont les plus compactes tout en étant les plus éloignées les unes des autres. Le score Laplacien supervisé est une variante de ce score de Fisher qui intègre un graphe de similarité entre données construit à partir des classes en présence.

Dans un contexte semi-supervisé où seulement quelques données sont labellisées, Zhao et al. proposent de coupler le score Laplacien évalué avec l'ensemble des données, à l'information

mutuelle entre les labels des données et les labels estimés par un algorithme de classification opérant avec l'attribut examiné [ZL07b]. Un attribut est d'autant plus pertinent que les projections des données similaires sont proches et que les labels estimés coïncident avec les labels fournis par l'expert. Ce score intéressant dépend toutefois fortement de l'algorithme de classification utilisé pour estimer les labels.

Comme nous l'avons souligné lors du chapitre précédent, formaliser la connaissance a priori sous la forme d'un nombre restreint de contraintes must-link et cannot-link portant sur des paires de données, est une solution qui peut faciliter le travail de l'expert. Ceci est d'autant plus vrai pour des données caractérisées par un nombre important d'attributs, qui rend les classes à retrouver difficilement définissables. Les attributs doivent donc être sélectionnés en s'appuyant sur les contraintes définies par l'expert.

Pour ce faire, les contraintes sont représentées à l'aide de graphes de similarité entre données, dont les matrices laplaciennes caractéristiques sont utilisées par les scores de contraintes récemment développés. Le respect de ces contraintes est le principal critère retenu par ces scores pour évaluer les attributs. Par conséquent, ils négligent les données sur lesquelles aucune contrainte n'a été définie par l'expert.

Ceci nous a amenée à présenter un nouveau score qui tient compte à la fois du respect des contraintes et de la structure locale des données. Ce score couple un score laplacien s'appuyant uniquement sur les contraintes avec le score laplacien intégrant la similarité Gaussienne entre toutes les données. Il est donc basé sur deux types de graphe de similarité, les graphes représentant les contraintes must-link et cannot-link ainsi que le graphe de similarité entre données.

Les exemples pédagogiques développés au sein ce chapitre montrent clairement que les attributs sélectionnés dépendent fortement du critère de sélection utilisé ainsi que du contexte d'apprentissage. Dans le chapitre suivant, nous nous restreignons au contexte semi-supervisé et nous proposons d'étudier l'influence des sous-ensembles de contraintes définis par l'expert sur l'évaluation des attributs.

Chapitre 3

Influence des contraintes sur le score des attributs

3.1 Introduction

Les scores de contraintes utilisent les sous-ensembles de contraintes must-link et cannot-link fournis par l'expert afin de sélectionner les attributs les plus pertinents. Par conséquent, ces scores sont fortement dépendants du cardinal et de la composition de ces sous-ensembles de contraintes.

En effet, une modification du contenu de ces sous-ensembles de contraintes peut conduire à un changement dans l'ordre de classement de ces attributs selon leur score et donc à un changement des attributs sélectionnés.

L'objectif de ce chapitre est d'étudier ce problème important de la sélection d'attributs par les divers scores de contraintes, à savoir, la dépendance des attributs sélectionnés vis à vis des contraintes. Nous étudions ainsi l'influence du sous-ensemble de contraintes sur les scores SC^1 (cf. équation (2.49)) , SC^2 (cf. équation (2.50)), SC^3 (cf. équation (2.52)) et SC^4 (cf. équation (2.56)). Ces scores sont retenus puisqu'il utilisent à la fois des contraintes must-link et des contraintes cannot-link dans la fonction d'évaluation des attributs.

Ainsi, dans la première partie (cf. section 3.2.1), nous introduisons un exemple illustratif pour illustrer l'influence du choix des contraintes sur le processus de sélection. Ensuite nous propo-

sons d'utiliser la matrice des rangs (cf. sections 3.2.2 et 3.2.3), ainsi que le coefficient de Kendall (cf. section 3.2.4) afin de mesurer cette dépendance. Des résultats expérimentaux réalisés sur des bases de données réelles décrites en section 3.3 montrent que notre score semi-supervisé avec contraintes SC^4 est moins dépendant des sous ensembles de contraintes disponibles que les scores SC^1, SC^2 et SC^3.

3.2 Influence du sous-ensemble de contraintes sur le classement des attributs

Les scores SC^1, SC^2, SC^3 et SC^4, développés dans le chapitre précédent, utilisent les sous-ensembles de contraintes must-link et cannot-link fournis par l'expert pour évaluer chacun des attributs. Ces attributs sont ensuite ordonnés par ordre croissant de leur score de contraintes afin de sélectionner les plus pertinents.

Cependant, l'intégration de ces contraintes dans la fonction d'évaluation rend ces scores très dépendants des sous-ensembles de contraintes fournis par l'expert. En effet, une modification du contenu de ces sous-ensembles de contraintes peut conduire à un changement dans l'ordre de classement de ces attributs selon leur score et donc à un changement des attributs sélectionnés. La procédure de sélection dépend donc fortement des contraintes must-link et cannot-link fournis par l'expert. En effet, à partir d'un ensemble de n données, il peut former C_n^2 contraintes must-link ou cannot-link portant sur des couples de données.

Ce phénomène a été également remarqué par Suna et al. [SZ10]. En effet, les auteurs ont démontré que les performances de diverses méthodes de sélection d'attributs basées sur les contraintes sont influencées par le choix de l'ensemble des contraintes ainsi que par leur cardinal.

Ils appliquent ainsi leurs scores de sélection SC^1 et SC^2 sur 4 bases de données de UCI repository, à savoir 'Credit approval', 'Horse', 'Vehicle' et 'Wine'. Les données de chaque base sont réparties selon la méthode Holdout , la moitié des données sont utilisées comme base d'apprentissage et l'autre moitié est utilisée comme base de test. La procédure de sélection est réalisée

sur la base d'apprentissage. A chaque fois, 60 contraintes sont générées aléatoirement à partir des labels des données de cette base. La base de test est utilisée pour évaluer les taux de bonne classification obtenus en utilisant le classifieur du k-plus proche voisin.

Les auteurs répétent ainsi la procédure de sélection et d'évaluation 100 fois en utilisant à chaque fois un ensemble différent de 60 contraintes. Ils remarquent alors un grand écart entre les performances sur les 100 exécutions. La différence entre les performances minimales et maximales des 100 exécutions sur les 4 bases est respectivement de 21.5%, 32.1%, 20.9% et 34.1% en utilisant le score SC^1 et 23.6%, 27.8%, 19.4% et 39.8% en utilisant le score SC^2.

Suna et al. [SZ10] ont donc montré que les performances de chacun des scores de sélection utilisant un ensemble de contraintes must-link et cannot link, est largement dépendante de ces contraintes. En effet, cette différence de performance entre deux exécutions de l'algorithme de sélection utilisant à chaque fois des contraintes différentes, résulte de la différence des rangs des attributs sélectionnés à chaque exécution. L'ensemble des contraintes utilisé à chaque fois par la procédure de sélection modifie les scores des attributs et ainsi leur ordre de sélection.

Afin d'illustrer le problème de dépendance des attributs sélectionnés par rapport aux sous-ensembles de contraintes must-link $\mathscr{M}$ et cannot-link $\mathscr{C}$, examinons l'exemple pédagogique suivant.

3.2.1 Exemple et discussion

3.2.1.1 Exemple illustratif

Soit l'ensemble de 4 données $\mathscr{X} = \{A, B, C, D\}$, caractérisées par 3 attributs f_1, f_2 et f_3 ($n = 4$, $d = 3$). Ces données sont définies par la matrice X comme suit :

$$X = \begin{bmatrix} -3 & -1 & 1 \\ -2 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -3 & -1 \end{bmatrix} \tag{3.1}$$

A cette matrice de données est associé le vecteur Y des labels des classes suivant :

$$Y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} \tag{3.2}$$

Les données A, B et C ont le label de la classe 1 et la donnée D a le label de la classe 2. La figure 3.1 illustre la représentation de ces données dans l'espace $\mathbb{R}^3$.

Nous pouvons remarquer que l'attribut f_3 est le seul attribut qui permet de discriminer les deux classes. Cet attribut a la valeur 1 pour les données de la classe 1 (A, B et C) et la valeur -1 pour les données de la classe 2 (D). Un algorithme de sélection efficace doit alors identifier f_3 comme l'attribut le plus pertinent.

Considérons le cas où, l'utilisateur construit à partir de Y un sous-ensemble de contraintes $\mathscr{S}$, de cardinal $|\mathscr{S}|$, formé d'une seule contrainte must-link et d'une seule contrainte cannot-link ($|\mathscr{S}| = 2$, $|\mathscr{M}| = 1$ et $|\mathscr{C}| = 1$).

Les paires $\{(A,B)\}$, $\{(A,C)\}$ et $\{(B,C)\}$ sont les sous-ensembles de contraintes must-link pos-

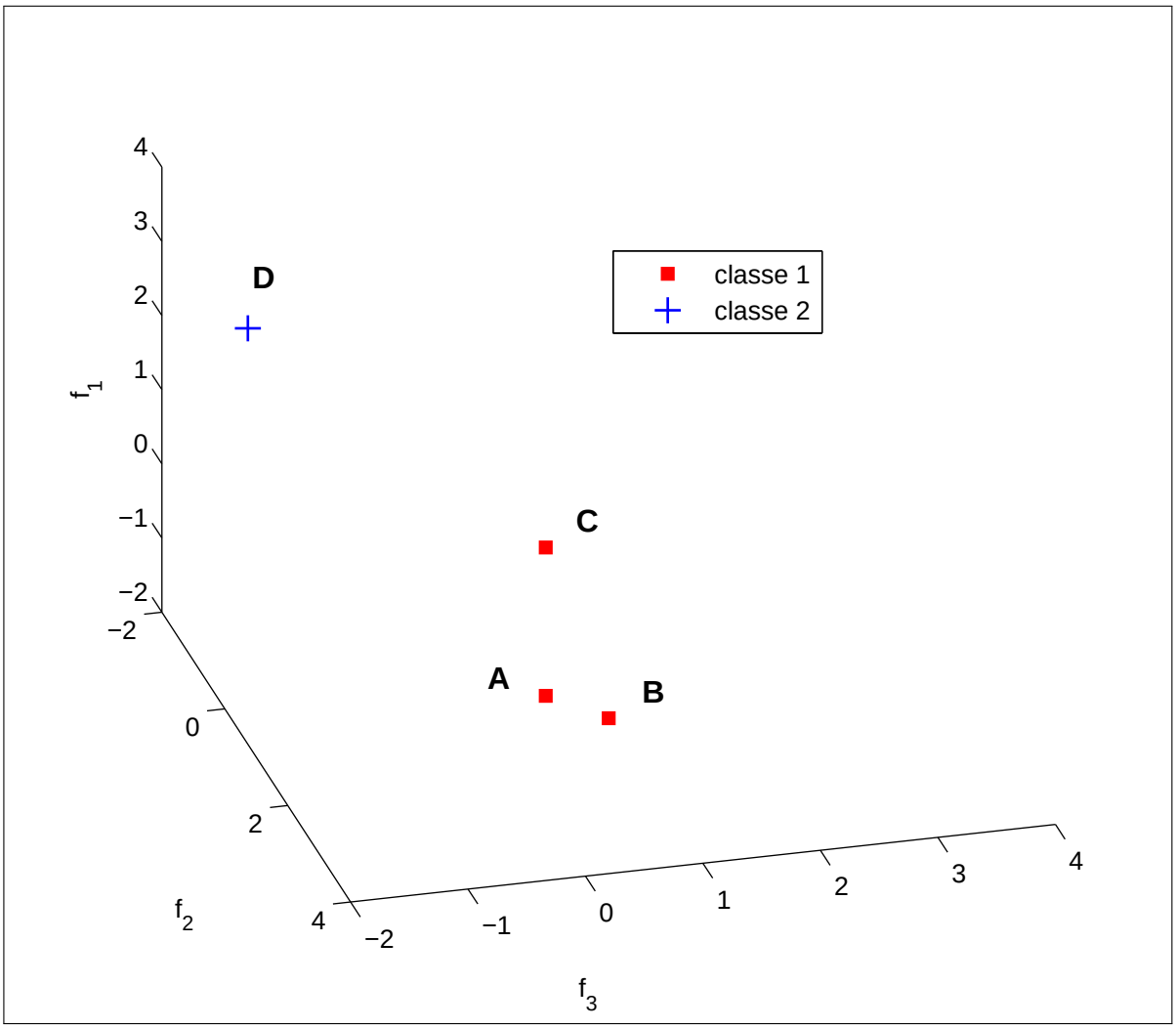


Figure 3.1 : Représentation des données dans l'espace $\mathbb{R}^3$.

sibles $\mathscr{M}$ et les paires $\{(A,D)\}$, $\{(B,D)\}$ et $\{(C,D)\}$ sont les sous-ensembles de contraintes cannot-link possibles $\mathscr{C}$ qui peuvent être construits à partir de ces données. On aura 9 combinaisons possibles de sous-ensembles $\mathscr{S}$ formés d'une contrainte must-link et d'une contrainte cannot-link $\mathscr{S} = \{(A,B),(A,D)\}$, $\{(A,B),(B,D)\}$, $\{(A,B),(C,D)\}$, $\{(A,C),(A,D)\}$, $\{(A,C),(B,D)\}$, $\{(A,C),(C,D)\}$, $\{(B,C),(A,D)\}$, $\{(B,C),(B,D)\}$, $\{(B,C),(C,D)\}$.

Les scores SC^1, SC^2, SC^3 et SC^4 sont utilisés afin d'ordonner les attributs f_1, f_2 et f_3 en utilisant une des neufs combinaisons. Les résultats de classement de ces attributs sont illustrés dans le tableau 3.1 composé de 9 cases.

Les première, deuxième, troisième et quatrième lignes de chaque case correspondent aux clas-

$\mathscr{C}\setminus\mathscr{M}$		$\{(A,B)\}$	$\{(A,C)\}$	$\{(B,C)\}$
$\{(A,D)\}$	SC^1	$\mathbf{f_3 = f_1, f_2}$	$\mathbf{f_3 = f_2, f_1}$	$\mathbf{f_3, f_1, f_2}$
	SC^2	$\mathbf{f_1, f_3, f_2}$	$f_3 = f_2, f_1$	f_3, f_1, f_2
	SC^3	f_1, f_3, f_2	$\mathbf{f_3, f_2, f_1}$	f_3, f_1, f_2
	SC^4	$f_3 = f_1, f_2$	$f_3 = f_2, f_1$	f_3, f_1, f_2
$\{(B,D)\}$	SC^1	$f_3 = f_1, f_2$	$f_3 = f_2, f_1$	$\mathbf{f_3, f_1 = f_2}$
	SC^2	f_1, f_3, f_2	$\mathbf{f_2, f_3, f_1}$	$f_3, f_1 = f_2$
	SC^3	f_1, f_3, f_2	f_2, f_3, f_1	f_3, f_2, f_1
	SC^4	$f_3 = f_1, f_2$	$f_3 = f_2, f_1$	f_3, f_1, f_2
$\{(C,D)\}$	SC^1	$f_3 = f_1, f_2$	$f_3 = f_2, f_1$	$f_3, f_1 = f_2$
	SC^2	$f_3 = f_1, f_2$	$f_3 = f_2, f_1$	$f_3, f_1 = f_2$
	SC^3	f_3, f_1, f_2	f_3, f_2, f_1	f_3, f_2, f_1
	SC^4	$f_3 = f_1, f_2$	$f_3 = f_2, f_1$	f_3, f_1, f_2

Tableau 3.1 : Le rang des différents attributs par les scores de sélection en utilisant le sous-ensemble de contraintes considéré.

sements des 3 attributs (f_1, f_2 et f_3) en utilisant respectivement les scores SC^1, SC^2, SC^3 et SC^4. Le signe '=' entre deux attributs signifie que ces deux attributs ont le même score et donc le même rang.

Examinons la case qui correspond à la contrainte must-link $\{(A,C)\}$ et à la contrainte cannot-link $\{(B,D)\}$ ($\mathscr{S} = \{(A,C), (B,D)\}$). Selon les scores SC^1 et SC^2, les attributs f_2 et f_3 sont les plus pertinents et l'attribut f_1 est classé au troisième rang. Selon les scores SC^2 et SC^3 l'attribut f_2 est classé au premier rang, l'attribut f_3 au second rang et l'attribut f_1 au troisième rang.

En examinant plus particulièrement le tableau 3.1, nous remarquons que l'attribut f_3 est toujours ordonné au premier rang par les scores SC^1 et SC^4, quelque soit le sous-ensemble de contraintes disponibles tandis cet attribut n'est pas classé trois fois au premier rang par les scores SC^2 et SC^3. Les scores SC^1 et SC^4 sont alors plus performants que les scores SC^2 et SC^3 sur cet exemple.

Nous pouvons remarquer que, pour chacun des scores de sélection, le classement des 3 attributs peut varier selon le sous-ensemble de contraintes retenu. En effet, 7 classements différents (indiqués en gras dans le tableau 3.1) des 3 attributs sont obtenus en utilisant les 9 combinaisons de contraintes possibles. Cet exemple simple montre alors clairement que le rang des attributs obtenus à partir des différents scores de contraintes dépend fortement du sous-ensemble de

contraintes disponibles.

3.2.1.2 Discussion

Les performances des différents scores de sélection sont généralement comparées en mesurant le taux de bonne classification des données projetées sur les vecteurs d'attributs sélectionnés par chacun de ces scores. Pour mesurer la dépendance des différents scores par rapport aux sous-ensembles de contraintes disponibles, nous aurions pu estimer la disparité des taux de classification obtenus par chacun des scores avec différents sous-ensembles de contraintes, comme l'ont fait Suna et al. [SZ10]. Cependant, cette évaluation nécessite la définition d'une règle de décision (comme le classifieur des k-plus proches voisins) qui va à chaque fois opérer dans l'espace d'attributs sélectionné. Cette étape de décision influence directement la qualité de classification. Par conséquent, la comparaison des performances des différents scores pourrait dépendre du classifieur utilisé.

Nous préférons alors examiner les attributs sélectionnés en utilisant les différents scores afin que la comparaison de leurs performances ne soit pas perturbée par l'étape de décision. Plus précisément, nous étudions la concordance entre le classement des différents attributs fourni par chacun des scores lorsque le sous-ensemble de contraintes est modifié.

Nous avons choisi le coefficient de Kendall afin de mesurer la variation de rangs des différents attributs fournis par les différents scores en fonction des contraintes. Avant de définir le coefficient de Kendal, il est nécessaire de présenter la matrice des rangs des différents attributs fournis par les différents scores.

3.2.2 Matrice des rangs

Soit le sous-ensemble de n données $\mathscr{X} = \{x_1,...,x_n\}$ caractérisées par d attributs et le vecteur $Y = (y_1,...,y_n)^T$ de labels de classes associé à ces données. Le vecteur Y des labels n'est utilisé que pour construire le $q^{ème}$ sous-ensemble de contraintes $\mathscr{S}_q$ utilisé lors de la $q^{ème}$ évaluation de la pertinence des attributs. Cela s'effectue en tirant au hasard des paires de données et en

mettant une contrainte must-link entre ces données si elles ont le même label et une contrainte cannot-link si elles ont des labels différents. Le cardinal de $\mathscr{M}_q$ et $\mathscr{S}_q$ est constant pour tout q. Les d attributs sont alors ordonnés par les différents scores SC^* ($* = 1, 2, 3, 4$) en utilisant le sous-ensemble de contraintes $\mathscr{S}_q$.

Soit R^*_{qr} le rang de l'attribut f_r par le score SC^*_r en utilisant le sous-ensemble de contraintes $\mathscr{S}_q$. Afin d'évaluer l'influence du sous-ensemble de contraintes, la procédure de classement des attributs est alors répétée p fois en utilisant à chaque fois un sous-ensemble de contraintes différent $\mathscr{S}_q$, $q = 1, ..., p$.

Ainsi, les rangs des d attributs obtenus par le score SC^* en utilisant p différents sous-ensembles de contraintes $\mathscr{S}_q$ sont regroupés dans la matrice R^* $(p \times d)$:

$$R^* = \begin{bmatrix} R^*_{11} & \dots & R^*_{1r} & \dots & R^*_{1d} \\ \dots & \dots & \dots & \dots & \dots \\ R^*_{q1} & \dots & R^*_{qr} & \dots & R^*_{qd} \\ \dots & \dots & \dots & \dots & \dots \\ R^*_{p1} & \dots & R^*_{pr} & \dots & R^*_{pd} \end{bmatrix} \tag{3.3}$$

Chaque ligne q de la matrice R^* représente le rang des d attributs fourni par le score SC^* en utilisant le sous-ensemble de contraintes $\mathscr{S}_q$ $(q = 1,...,p)$. Chaque colonne r représente les rangs de l'attribut f_r en utilisant les p différents sous-ensembles de contraintes. Par conséquent, chaque ligne de R^* est une permutation de rangs des d attributs selon le sous-ensemble de contraintes $\mathscr{S}_q$.

Ci-dessous, nous allons représenter un exemple qui illustre la construction de la matrice des rangs R.

3.2.3 Matrice des rangs évaluée sur un exemple

Reprenons la matrice des données (3.1), A partir du tableau 3.1, nous pouvons construire les différentes matrices R^1, R^2, R^3 et R^4, qui correspondent aux rangs des différents attributs par

les scores SC^1, SC^2, SC^3 et SC^4 en fonction des sous-ensembles de contraintes.

Afin d'illustrer la matrice $R^3(9 \times 3)$ relative au score SC^3, nous construisons le tableau 3.2 qui correspond aux rangs des différents attributs en utilisant le sous-ensemble de contraintes $\mathscr{S}$. En effet, dans ce tableau, nous avons juste recopié la $3^{ème}$ ligne de chaque case du tableau 3.1. A

$\mathscr{S}$	Classement
$\{(A,B),(A,D)\}$	$f_1,\ f_3,\ f_2$
$\{(A,C),(A,D)\}$	$f_3,\ f_2,\ f_1$
$\{(B,C),(A,D)\}$	$f_3,\ f_1,\ f_2$
$\{(A,B),(B,D)\}$	$f_1,\ f_3,\ f_2$
$\{(A,C),(B,D)\}$	$f_2,\ f_3,\ f_1$
$\{(B,C),(B,D)\}$	$f_3,\ f_2,\ f_1$
$\{(A,B),(C,D)\}$	$f_3,\ f_1,\ f_2$
$\{(A,C),(C,D)\}$	$f_3,\ f_2,\ f_1$
$\{(B,C),(C,D)\}$	$f_3,\ f_2,\ f_1$

Tableau 3.2 : Le rang des différents attributs par le score SC^3 en utilisant le sous-ensemble de contraintes considéré.

partir de ce tableau, nous pouvons ainsi construire la matrice R^3 :

$$R^3 = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 3 & 2 \\ 3 & 1 & 2 \\ 3 & 2 & 1 \\ 2 & 3 & 1 \\ 3 & 2 & 1 \\ 3 & 2 & 1 \end{bmatrix} \tag{3.4}$$

Chaque ligne de R^3 correspond aux rangs des attributs f_1, f_2 et f_3 en utilisant un sous-ensemble de contraintes. Chaque colonne représente les différents rangs de l'attribut considéré en fonction des contraintes.

Ainsi, en faisant le lien entre la matrice R^3 et le tableau 3.2, nous pouvons en déduire qu'en

utilisant le sous-ensemble de contraintes $\mathscr{S}$ formé de la contrainte must-link $\{(A,B)\}$ et la contrainte cannot-link $\{(A,D)\}$, l'attribut f_1 a le rang 1, l'attribut f_2 a le rang 3 et l'attribut f_3 a le rang 2.

Il est important de signaler que dans le cas où les attributs ont le même score, ils auront le même rang, à savoir la moyenne des leurs. A titre d'illustration nous représentons par exemple la matrice R^1 qui représente des attributs avec des scores égaux :

$$R^1 = \begin{bmatrix} 1.5 & 3 & 1.5 \\ 3 & 1.5 & 1.5 \\ 2 & 3 & 1 \\ 1.5 & 3 & 1.5 \\ 3 & 1.5 & 1.5 \\ 2.5 & 2.5 & 1 \\ 1.5 & 3 & 1.5 \\ 3 & 1.5 & 1.5 \\ 2.5 & 2.5 & 1 \\ \end{bmatrix} \tag{3.5}$$

3.2.4 Coefficient de Kendal

Nous utilisons le coefficient de Kendall afin de mesurer la concordance ou l'accord entre les rangs des attributs en utilisant p différents sous-ensembles de contraintes. [Grz06].

Le coefficient de Kendall K^* prend en considération les différentes lignes de la matrice R^*. Il est défini par [SC88] :

$$K^* = \frac{12\Delta^*}{p^2\,(d^3-d) - p.\tau^*}, \tag{3.6}$$

avec $\Delta^* = \sum_{r=1}^{d} (R_r^* - \bar{R^*})^2$, $R_r^* = \sum_{q=1}^{p} R_{qr}^*$, $\bar{R^*} = \frac{1}{d}\sum_{r=1}^{d} R_r^*$ et $\tau^* = \sum_{v=1}^{m} \left(\tau_v^{*3} - \tau_v^*\right)$

où Δ^* représente la dispersion des rangs des attributs à travers les p expériences. Le terme τ_v^* est

le nombre d'attributs ayant des scores égaux et par suite des rangs égaux parmi les m groupes de rangs égaux dans R^*. La somme τ^* est calculée sur tous les groupes de rangs égaux trouvés dans les p lignes de la matrice R^*. Par exemple, dans la matrice (3.5), il y a 8 lignes de R^1 où les attributs sont égaux, m est alors égal à 8. τ_v^* représente le nombre d'attributs égaux dans chacune de ces lignes. Par exemple, dans la matrice (3.5), τ_v^* est égal à 2 pour chaque $v^{ème}$ ligne, v=1, ..., m.

Les valeurs du coefficient de Kendall K^* varient entre 0 (rangs complètement indépendants) et 1 (accord parfait entre des rangs identiques). Une valeur proche de 1 de K^* sera interprétée comme une robustesse du classement des attributs vis-à-vis des contraintes retenues.

3.2.5 Application du coefficient de Kendall sur notre exemple

Nous reprenons l'exemple des données de la section 3.2.1, représentées par la matrice (3.1). A partir du tableau 3.1, nous calculons le coefficient de Kendall K^* de chacun des scores SC^* afin de mesurer la concordance des rangs des attributs obtenus par chaque score. Les coefficients de Kendall K^1, K^2, K^3 et K^4 des scores de contraintes SC^1, SC^2, SC^3 et SC^4 sont respectivement 0.4325, 0.2258, 0.3333 et 0.4333.

Ces faibles valeurs des coefficients de Kendall reflètent la forte dépendance des rangs des attributs par rapport aux sous-ensembles de contraintes. Dans cet exemple, le coefficient de Kendall K^4 du score SC^4 est légèrement plus élevé que les autres coefficients. Cependant, il s'agit d'un simple exemple de faible effectif (4 données), dont l'objectif n'est pas de comparer les coefficients de Kendall des différents scores. Nous cherchons à travers cet exemple à mettre en évidence la dépendance du classement des attributs fourni par chacun des scores par rapport au sous-ensemble de contraintes. Cette conclusion rejoint l'observation de Sun et al. [SZ10].

3.2.6 Influence du sous-ensemble de contraintes sur le score SC^4

Le score de sélection SC^4 prend en considération l'ensemble des données en plus des contraintes disponibles dans le processus de sélection. Le fait de prendre en considération les données non-contraintes rend alors ce score moins sensible aux sous-ensembles de contraintes. Plus particulièrement, le couplage du score Laplacien SL et du score de contraintes SC^1, utilisé par le score semi-supervisé avec contraintes, le rend plus stable et donc moins dépendant du jeu de contraintes :

$$SC_r^4 = SL_r.SC_r^1 \tag{3.7}$$

En effet, le premier terme du produit de l'équation (3.7) est constant quelque soit le sous-ensemble de contraintes et c'est le deuxième terme de cette équation qui varie en fonction des contraintes .

Dans la partie suivante, nous allons présenter de nombreux résultats obtenus avec des bases de données réelles afin de montrer que notre score de sélection semi-supervisé avec contraintes SC^4 est moins dépendant des sous-ensembles de contraintes que les scores existants.

3.3 Résultats expérimentaux

Afin de comparer le coefficient de Kendall des différents scores de contraintes SC^1, SC^2, SC^3 ainsi que celui de notre score semi-supervisé avec contraintes SC^4, nous avons choisi six bases de données largement utilisées dans la littérature. Plus précisément, il s'agit des bases 'Wine', 'Image segmentation' et 'Vehicle' de UCI repository ([BKM98]), la base de visages 'ORL' ([SH94]) et les deux bases d'expression de gènes 'Colon Cancer'([ABN+99]) et 'Leukemia'([GST+99]).

Ces bases de données réelles ont été retenues parce qu'elles sont caractérisées par des attributs

numériques. De même, le vecteur Y de labels de classes de données est bien défini pour chacune de ces bases. Ainsi, la génération des contraintes à partir de ces labels est une opération assez facile.

Dans nos expériences, nous avons normalisé les différents attributs entre 0 et 1 afin que l'échelle des valeurs soit identique. Nous avons choisi la méthode de partition hold-out qui consiste à diviser la base de données en deux parties : la première partie est la base d'apprentissage, la deuxième partie étant la base test.

Les différents scores de sélection ainsi que les résultats du coefficient de Kendall sont appliqués sur la base d'apprentissage. La base test est quant à elle utilisée dans l'évaluation des résultats de classification des divers scores détaillée au chapitre 4.

Il est important de signaler qu'aucun critère d'arrêt n'est utilisé dans la procédure de sélection. Ainsi, les n attributs disponibles dans chaque base sont classés en utilisant les scores SC^1, SC^2, SC^3 et SC^4.

3.3.1 Résultats sur les bases UCI et la base ORL

Nous avons en premier lieu réalisé des expérimentations sur 3 bases de données UCI et sur la base de visages ORL, que nous allons décrire.

3.3.1.1 La base 'Wine'

La base de donnée 'Wine' contient les résultats d'une analyse chimique des vins produits dans la même région en Italie, mais provenant de trois cultivateurs différents . Elle est constituée de 178 données caractérisées par 13 attributs (n=178, d=13). Ces données sont divisées en 3 classes ayant pour effectif 59, 71 et 48 données. Nous sélectionnons 30, 36 et 24 données de chaque classe afin de constituer la base d'apprentissage. Les données restantes constituent alors la base test.

3.3.1.2 La base 'Image segmentation'

La base 'Image segmentation' est créée à partir de 7 images d'extérieur segmentées à la main. Cette base de données contient 210 données qui correspondent aux régions dans les images. Ces données sont caractérisées par 19 attributs (n=210, d=19) regroupées en 7 classes. Ces 7 classes sont équiprobables et ont chacun un effectif de 30 données. Nous sélectionnons 15 données de chaque classe afin de constituer la base d'apprentissage et les 15 données restantes de chaque classe constituent alors la base de test.

3.3.1.3 La base 'Vehicle'

La base 'Vehicle' contient 846 données correspondant à des modèles de véhicules caractérisées par 18 attributs (n=846, d=18). Ces données sont divisées en 4 classes d'effectif respectif 212, 217, 218 et 199. Nous sélectionnons 106, 109, 109 et 100 données de chaque classe pour former la base d'apprentissage.

Ces 3 bases de données ('Wine', 'Image segmentation' et 'Vehicle') souffrent du problème de la malédiction de la dimension. En effet, les classes présentes peuvent être aussi bien discriminées dans un espace réduit d'attributs bien sélectionnés que dans l'espace d'attributs original. Certains attributs présents n'apportent aucune information pour la discrimination des différentes classes.

3.3.1.4 La 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 ($n = 400$). 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) (voir Figure 3.2).

Dans nos expériences, les images originales acquises initialement ont été normalisées en échelle

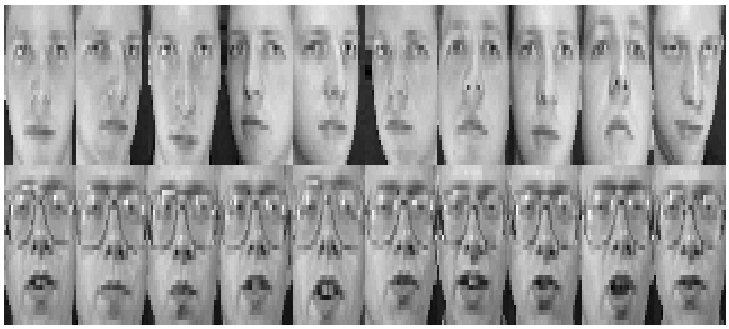


Figure 3.2 : Exemples d'images de la base ORL (2 sujets).

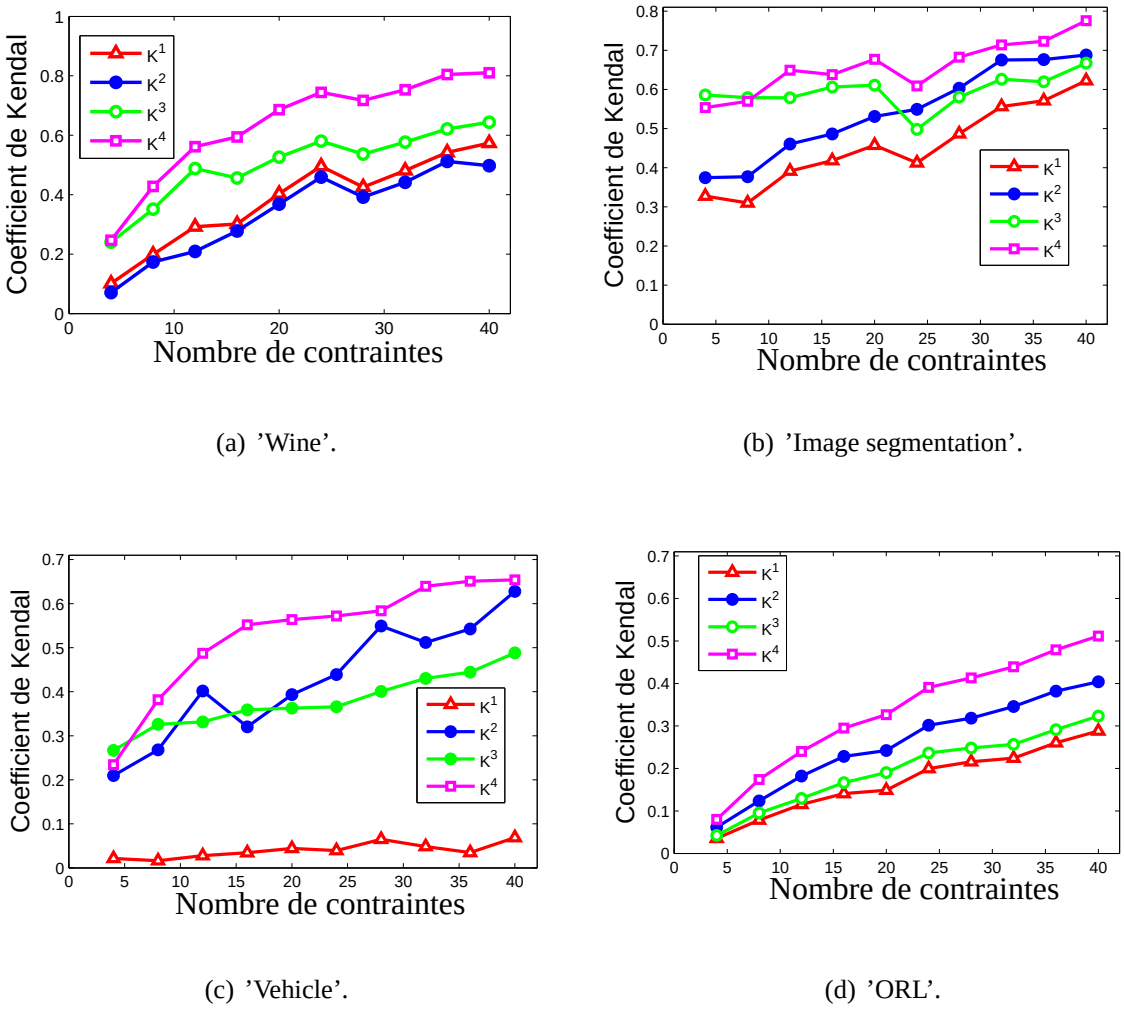
et en orientation pour que les yeux soient alignés dans la même position horizontale. Ensuite, la zone du visage a été découpée afin de former des images de taille 32×32 pixels. Ainsi, chaque image peut être représentée par 1024 attributs ($d = 1024$). Le niveau de gris de chaque pixel est quantifié à l'aide de 256 niveaux.

Nous sélectionnons 5 images de chaque classe (personne) pour construire la base d'apprentissage (soit 200 images). Les 200 images restantes constituent la base de test.

On peut remarquer que la taille de la population des données d'apprentissage de ces 4 bases est relativement faible. Par conséquent, on peut considérer que le nombre de contraintes retenues reflète avec une qualité similaire de représentation les données d'apprentissage.

3.3.1.5 Résultats du coefficient de Kendal

Dans nos expériences, la procédure de sélection est réalisée sur les données d'apprentissage. Les attributs sont ainsi ordonnés selon les différents scores de sélection. A chaque exécution de la procédure de classement des attributs q, q=1,..., p, nous simulons la génération des contraintes comme suit : nous sélectionnons aléatoirement une paire de données de la base d'apprentissage pour créer une contrainte must-link ou cannot link selon que ces deux données appartiennent à la même classe ou à des classes différentes. Cette procédure est itérée jusqu'à ce qu'on obtienne $\frac{|\mathscr{S}_q|}{2}$ contraintes must-link et $\frac{|\mathscr{S}_q|}{2}$ contraintes cannot-link .



(a) 'Wine'. (b) 'Image segmentation'.

(c) 'Vehicle'. (d) 'ORL'.

Figure 3.3 : Coefficients de Kendall en fonction du nombre $\left|\mathscr{S}_q\right|$ de contraintes sur les 3 bases de données UCI ainsi que sur la base ORL.

La procédure de sélection est ainsi répétée 100 fois ($p = 100$) afin de calculer les différents coefficients de Kendall K^1, K^2, K^3 et K^4 sur chacune des bases de données.

Nous réalisons nos expériences pour différents cardinaux de $\mathscr{S}_q$ allant de 4 contraintes (2 must-link et 2 cannot-link), jusqu'à 40 contraintes comme le font Zhang et al. [ZCZ08].

La figure 3.3 montre les résultats des coefficients de Kendall K^1, K^2, K^3 et K^4 calculés sur 100 exécutions ($p = 100$) pour différents nombres de contraintes sur les 4 bases de données considérées (Wine, Image segmentation, Vehicle et ORL). A chaque exécution, le même sous-ensemble de contraintes est bien sûr considéré par les 4 critères de sélection testés.

A partir de cette figure, nous pouvons remarquer que les différents coefficients K^1, K^2, K^3 et

K^4 ont des valeurs faibles (<0.5) lorsque $|\mathscr{S}_q|$ est faible. Ces faibles valeurs de K^1, K^2, K^3 et K^4 dans cette figure montrent que les attributs sélectionnés par les scores SC^1, SC^2, SC^3 et SC^4 sont fortement dépendants des sous-ensembles de contraintes disponibles, lorsque nous ne considérons que quelques contraintes dans la procédure de sélection.

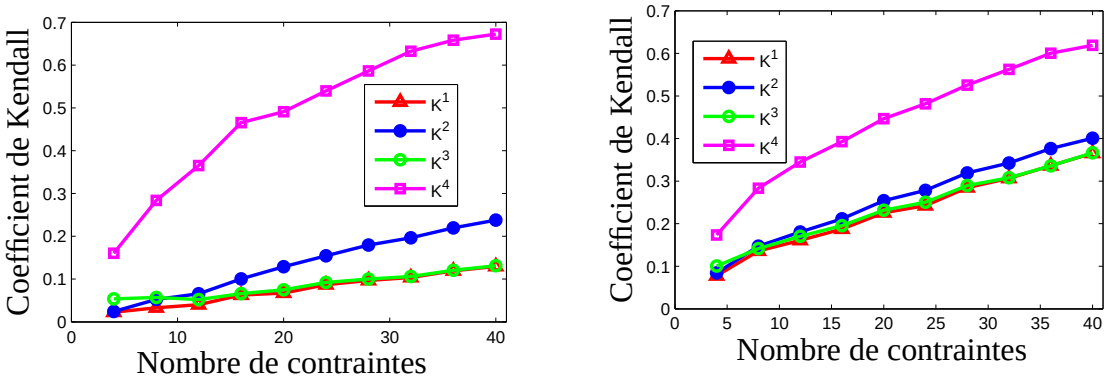
En examinant particulièrement les différentes courbes de la figure 3.3, nous pouvons remarquer que les courbes de K^1, K^2, K^3 et K^4 sont croissantes. Ainsi le coefficient de Kendall augmente avec le nombre $|\mathscr{S}_q|$ de contraintes considérées. En effet, plus le nombre de contraintes $|\mathscr{S}_q|$ est élevé, plus l'information supervisée est complète. Ainsi, lorsque $|\mathscr{S}_q|$ est élevé, le contexte semi-supervisé tend à devenir un contexte supervisé. Cela explique alors le fait que la concordance entre les rangs des attributs augmente lorsque $|\mathscr{S}_q|$ augmente, dans les différentes courbes de la figure 3.3.

Nous pouvons aussi noter que les valeurs de K^4 sont les plus élevées pour les différents nombres $|\mathscr{S}_q|$ de contraintes et pour toutes les bases de données considérées. Il atteint même des valeurs proches de 0.8 pour les bases 'Wine' et 'Image segmentation' lorsque le nombre de contraintes est important. Cela confirme alors que notre score de sélection semi-supervisée avec contraintes SC^4 , qui prend en considération toutes les données, est moins dépendant des sous-ensembles de contraintes must-link et cannot-link que les autres scores existants (SC^1, SC^2 et SC^3).

3.3.2 Résultats sur les bases d'expression de gènes

Dans cette partie, nous présentons les résultats expérimentaux réalisés sur deux bases d'expression : 'Colon Cancer' ([ABN+99]) et 'Leukemia' ([GST+99]). Ces deux bases contiennent un faible nombre de données caractérisées par un nombre élevé d'attributs. Ces deux cas sont alors confrontés au problème de 'small-sample problem'. La dimension de l'espace original de représentation des données est nettement supérieure à la taille de la population des données.

Avant de représenter les résultats du coefficient de Kendall sur ces deux bases, nous donnons une description de chacune d'elles.



(a) 'Colon Cancer'. (b) 'Leukemia'.

Figure 3.4 : Coefficients de Kendall en fonction du nombre $\left|\mathscr{S}_q\right|$ de contraintes sur les 2 bases d'expression de gènes.

3.3.2.1 La base 'Colon Cancer'

Cette base est constituée de 62 tissus répartis en 2 classes, 40 tumoraux et 22 normaux ($n = 62$). Ces tissus sont caractérisés par l'expression de 2000 gènes ($d = 2000$). Nous sélectionnons 20 tissus tumoraux et 11 normaux pour former la base d'apprentissage. Les données restantes constituent la base test.

3.3.2.2 La base 'Leukemia'

Cette base de données contient des informations d'expression de gènes chez 72 personnes atteints de deux types de leucémies AML (acute myeloid leukemias) et ALL (acute lymphoblastic leukemias)($n = 72$). La base de données originales est formée de 6817 gènes, cependant nous éliminons les gènes dont nous ignorons les mesures sur au moins un échantillon. Au total, 5147 gènes sont utilisés dans notre expérimentation ($d = 5147$).

Comme les données de la base 'Leukemia' sont déjà partitionnées en un sous-ensemble d'apprentissage (27 ALL et 11 AML) et un sous-ensemble test (20 ALL et 14 AML) ([GST+99]), toutes les expériences sur cette base ont été réalisées en utilisant cette partition.

3.3.2.3 Résultats du coefficient de Kendal

La figure 3.4 montre les coefficients K^1, K^2, K^3 et K^4 calculés pour 100 exécutions de la procédure de sélection $p = 100$ et en considérant differents nombres de contraintes sur les bases 'Colon Cancer' et 'Leukemia'.

Nous en déduisons les mêmes remarques que pour les résultats des coefficients de Kendall sur les bases UCI et la base 'ORL'. D'après la figure 3.4 , les coefficients K^1, K^2, K^3 et K^4 ont des valeurs faibles (<0.5) lorsque $\left|\mathscr{S}_q\right|$ est faible. De même, les différentes courbes sont croissantes : le coefficient de Kendall augmente avec le nombre $\left|\mathscr{S}_q\right|$ de contraintes considéré. En plus, comme pour les bases de données précédentes, les valeurs de K^4 sont les plus élevées pour les différents nombres de contraintes. Cela confirme que notre score SC^4 est moins dépendant des contraintes must-link et cannot-link que les scores classiques.

En examinant les figures 3.4(a) et 3.4(b), nous constatons que l'écart entre le coefficient K^4 d'une part et les autres coefficients K^1, K^2 et K^3 d'autre part est globalement plus élevé sur ces bases de données (Colon Cancer et Leukemia) que sur les autres bases. Cela est probablement dû au fait que le nombre d'attributs dans ces bases est plus élevé.

Zhao et Zhang ont expérimentalement démontré qu'un petit nombre de contraintes $\left|\mathscr{S}_q\right|$ est suffisant pour que leurs scores de contraintes SC^1 et SC^2 sélectionnent les attributs permettant de discriminer les différentes classes en présence [ZCZ08]. Cependant, les figures 3.3 et 3.4 montrent que, pour des sous-ensembles de contraintes différents, les rangs des attributs changent. Les attributs sélectionnés changent, surtout lorsque le nombre $\left|\mathscr{S}_q\right|$ de contraintes est petit. Par conséquent, la qualité de la discrimination peut fortement varier en fonction du sous-ensemble de contraintes en présence.

3.4 Conclusion

Les scores de contraintes nécessitent de définir les sous-ensembles de contraintes must-link et cannot-link dans la fonction d'évaluation. Les attributs sélectionnés par ces scores sont alors fortement dépendants de l'ensemble des contraintes.

Dans ce chapitre, nous avons étudié l'influence des contraintes définies par l'expert sur le classement des attributs à partir des différents scores. Nous avons ainsi utilisé le coefficient de Kendall pour mesurer la dépendance des rangs des attributs fournis par les différents scores par rapport au changement de contraintes disponibles. A notre connaissance, c'est le premier travail de ce genre, qui compare la robustesse des divers scores en mesurant leur dépendance vis-à-vis des contraintes.

Le coefficient de Kendall appliqué sur un exemple illustratif a clairement montré que les rangs des attributs classés par les différents scores de contraintes dépendent fortement des contraintes disponibles. Ceci s'explique par le fait que ces scores de contraintes ne prennent pas en considération l'information fournie par l'ensemble des données non contraintes. Ceci nous a amenée à proposer le score SC^4 de sélection semi-supervisé avec contraintes qui prend en considération à la fois l'ensemble des données et le sous-ensemble de contraintes disponibles.

Les résultats expérimentaux réalisés sur trois bases UCI, une base de visages et deux bases d'expressions de gène ont permis de comparer notre score SC^4 avec les scores SC^1, SC^2 et SC^3. Les résultats du coefficient de Kendall montrent que le score SC^4 est moins dépendant du jeu de contraintes que les autres scores puisque le coefficient de Kendall K^4 est le plus élevé. Les résultats sur les bases d'expression de gènes ont aussi montré que l'amélioration du coefficient de Kendall par notre score tend à augmenter avec le nombre d'attributs considérés.

Nous proposons de comparer les performances en termes de qualité de classification, atteintes par ces différents scores de contraintes dans le prochain chapitre.

Chapitre 4

Évaluation des méthodes de sélection d'attributs par la qualité de la classification

4.1 Introduction

La comparaison des différents scores de sélection d'attributs qui utilisent un ensemble de contraintes must-link et cannot link a montré que notre score semi-supervisé avec contraintes SC^4 est moins dépendant des sous-ensembles de contraintes que les scores classiques SC^1, SC^2 et SC^3. Cette comparaison qui utilise le coefficient de Kendall examine uniquement les rangs des attributs fournis par chacun des scores sans considérer les performances en classification.

Pour comparer les différents scores de sélection d'attributs, les données sont divisées en une base d'apprentissage et une base test. Après avoir réalisé la procédure de sélection d'attributs sur la base d'apprentissage, la performance de l'algorithme de sélection est mesurée par les taux de bonne classification des données test obtenus par un algorithme de classification qui opère dans l'espace de représentation défini par les attributs sélectionnés.

Le classifieur le plus utilisé dans ce cadre est celui du plus proche voisin car il permet de discriminer des données non linéairement séparables. Comme ce classifieur du plus proche voisin nécessite la définition de prototypes des classes, il utilise les données d'apprentissage avec leurs

labels comme prototypes.

Cependant, ces labels n'ont pas été exploités par les scores de sélection. En effet, ces scores utilisent uniquement un nombre restreint de contraintes entre données et/ou l'ensemble de données non labellisées. Par conséquent, les données test sont classifiées dans un contexte supervisé car les données d'apprentissage avec leur labels constituent les prototypes des classes, alors que les attributs ont été sélectionnés dans un contexte semi-supervisé car les scores utilisent seulement un nombre restreint de contraintes must-link et cannot-link.

Par ailleurs, dans le cas d'applications réelles, la base d'apprentissage est soit non labellisée, soit partiellement labellisée. L'emploi de l'algorithme supervisé du plus proche voisin est par conséquent non réaliste étant donné la non disponibilité des labels de classes de la base d'apprentissage.

C'est pour cette raison que nous proposons une procédure d'évaluation semi-supervisée des différents scores. La seule information disponible étant les contraintes must-link et cannot-link portant sur quelques paires de données, c'est donc cette information qui sera exploitée à la fois dans la procédure de sélection et de classification.

Ainsi, notre chapitre sera organisé comme suit. Dans la première partie (cf. section 4.2), nous détaillons la procédure supervisée généralement utilisée dans la littérature pour évaluer les différents scores. Nous évoquons alors l'algorithme des k-plus proche voisins (cf. section 4.2.1) et nous appliquons cette procédure sur les différentes bases de données utilisées dans le chapitre 3 afin de comparer les performances des quatres scores SC^1, SC^2, SC^3 et SC^4 (cf. section 4.2.2). Dans la deuxième partie, nous proposons une procédure de comparaison cohérente avec la procédure de sélection développée (cf. section 4.3). Ainsi, pour le score Laplacien nous proposons une évaluation non supervisée basée sur l'algorithme des k-means (cf. section 4.3.1) et pour les scores de contraintes, nous proposons une évaluation semi-supervisée basée sur l'algorithme des k-means sous-contraintes (cf. section 4.3.2). Nous comparons les résultats de classification des divers scores obtenus dans un contexte semi-supervisé avec les résultats obtenus par l'éva-

luation supervisée classique (cf. section 4.3.2.2). Nous comparons aussi les différents scores de contraintes entre eux en utilisant l'évaluation semi-supervisée (section 4.3.2.3).

4.2 Evaluation supervisée des performances des scores

Afin de comparer les performances de plusieurs méthodes de sélection d'attributs, les travaux dans la littérature jugent de la qualité de classification obtenue par un classifieur opérant dans l'espace des attributs sélectionnés [ZCZ08] [ZLH08]. Une méthode de sélection est considérée supérieure à une autre si elle permet d'obtenir des meilleurs taux de bonne classification. La sélection est réalisée à partir de contraintes disponibles pouvant être déduites des données d'apprentissage. La comparaison de la performance de la sélection d'attributs associée aux différents scores est ensuite réalisée sur les données test. Ainsi, l'algorithme des k-plus proches voisins opère dans l'espace des attributs sélectionnés par les différents scores, en utilisant les données d'apprentissage comme prototypes des classes. Les résultats de classification des données test sont ensuite comparés.

Comme cette comparaison est basée sur l'algorithme des k-plus proches voisins, nous détaillons cet algorithme avant de présenter les résultats obtenus sur différentes bases de données.

4.2.1 L'algorithme des k-plus proches voisins

La méthode des k-plus proches voisins notée k-ppv est une méthode de classification simple permettant de traiter des données non linéairement séparables. Elle est couramment utilisée pour valider la sélection et comparer ainsi diverses méthodes de sélection d'attributs. Elle permet, à partir des données d'apprentissage dont on connait le label, de classifier les données test. Cet algorithme est basé sur l'idée de mesurer dans l'espace d'attributs sélectionné la proximité entre la donnée test à classer et les données d'apprentissage représentant les différents prototypes des classes. Cette proximité est généralement mesurée à l'aide de la distance Euclidienne dans l'espace d'attributs considéré. Le classifieur k-ppv détermine la classe d'une donnée test en

lui attribuant la classe majoritaire des k données qui lui sont les plus proches dans la base d'apprentissage.

Afin d'être le plus simple et indépendant des règles de décision sur le vote majoritaire en cas d'égalité de votes, nous avons choisi de fixer k à 1, c'est donc le classifieur du plus proche voisin qui est utilisé. A chaque donnée test est alors attribuée la même classe que celle de la donnée d'apprentissage qui lui est la plus proche dans l'espace d'attributs considéré et cela en utilisant la distance Euclidienne.

4.2.2 Résultats expérimentaux

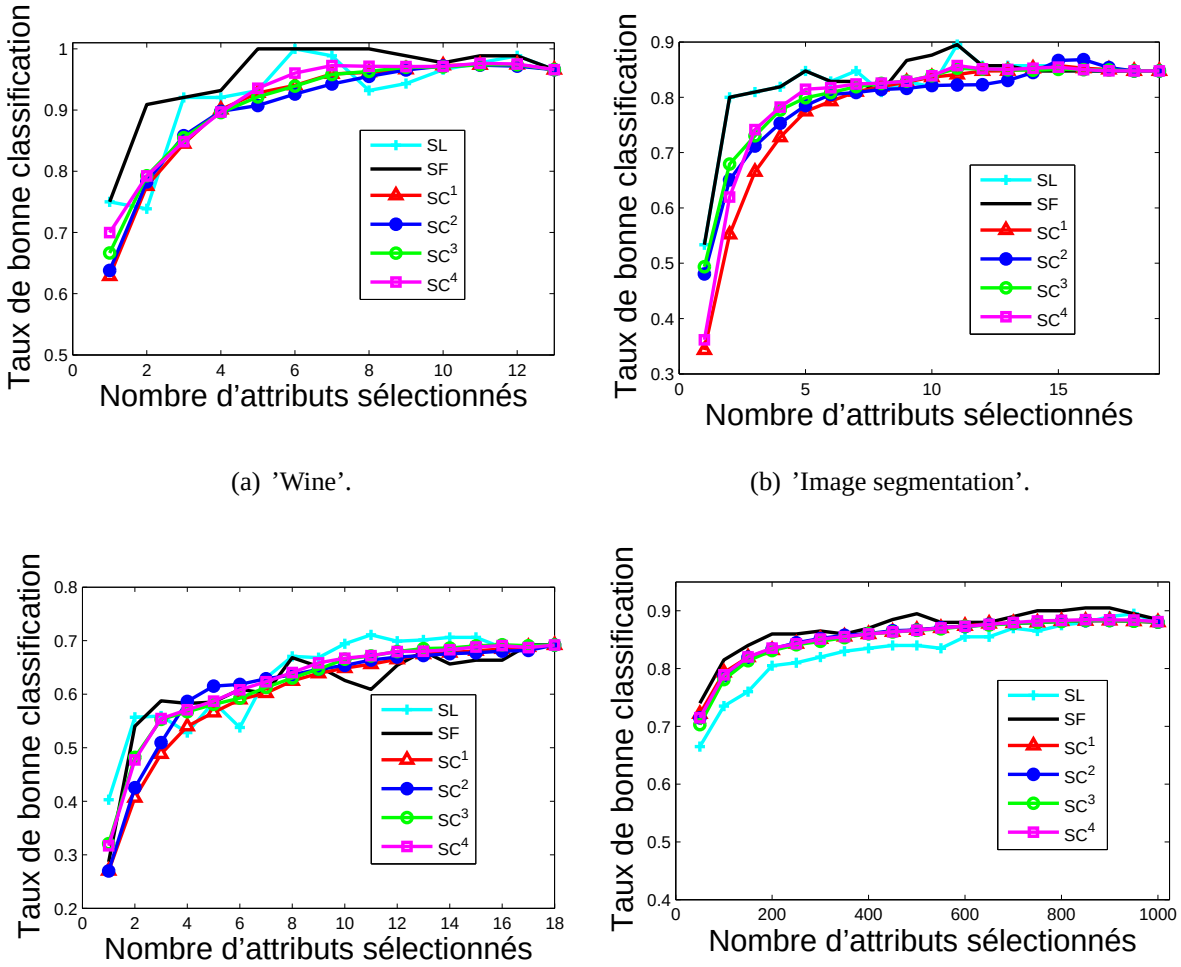
Dans cette partie, nous comparons les performances des différents scores de sélection opérant dans 3 contextes d'apprentissage différents : le score Laplacien non supervisé *SL* (voir équation (2.16)) , le score de Fisher supervisé *SF* (voir équation (2.24)), les scores de contraintes SC^1 (voir équation (2.49)) , SC^2 (voir équation (2.50)), SC^3 (voir équation (2.52)), ainsi que notre score de sélection semi-supervisé avec contraintes SC^4 (voir équation (2.56)). Pour ce faire, nous utilisons la procédure de comparaison basée sur une classification supervisée des données test.

Les expérimentations sont réalisées sur les mêmes bases de données décrites dans le chapitre 3 : les bases 'Wine', 'Image segmentation' et 'Vehicle' de UCI repository ([BKM98]), la base de visage 'ORL' ([SH94]) et les deux bases d'expression de gènes 'Colon Cancer'([ABN$^+$99]) et 'Leukemia'([GST$^+$99]). La partition retenue entre données d'apprentissage et données test est celle décrite également dans le chapitre 3.

4.2.2.1 Résultats sur les bases UCI

4.2.2.1.a Comparaison des taux moyens de classification

Nous comparons la performance (taux de bonne classification ou accuracy) obtenue en utilisant le classifieur du plus proche voisin qui opère dans l'espace sélectionné par les différents scores considérés. Cet espace est constitué des attributs ayant obtenu individuellement les



(a) 'Wine'. (b) 'Image segmentation'.

(c) 'Vehicle'. (d) 'ORL'.

Figure 4.1 : Taux de bonne classification en fonction du nombre d'attributs sélectionnés sur les 4 bases de données. 10 contraintes formées de 5 must-link et 5 cannot-link sont utilisées. L'évaluation est effectuée dans un contexte supervisé : l'algorithme du 1-ppv utilise les labels réels des données d'apprentissage comme prototypes des classes.

meilleurs scores. Le taux de bonne classification est la proportion de données test bien classifiés par rapport au nombre total de ces données.

Comme pour la mesure de coefficients de Kendal, les taux de bonne classification sont moyennés sur 100 exécutions ($p = 100$) avec différentes générations de contraintes pour les quatre scores de contraintes examinés.

La figure 4.1 illustre les taux de bonne classification en fonction du nombre d'attributs sélectionnés respectivement sur les bases 'Wine' 4.1(a), 'Image segmentation' 4.1(b), 'Vehicle' 4.1(c) et 'ORL' 4.1(d) et cela en utilisant 10 contraintes ($\left|\mathscr{S}_q\right| = 10$) incluant 5 must-link et 5 cannot-

link.

A partir de cette figure, nous pouvons remarquer que le taux maximal de bonne classification obtenu par chacun des scores, que cela soit dans un contexte non-supervisé, supervisé ou semi-supervisé n'est pas toujours atteint dans l'espace d'entré formé par la totalité des attributs. Cela illustre alors le problème de la malédiction de la dimension et la nécessité de sélectionner un sous-ensemble d'attributs à partir de l'ensemble initial d'attributs.

Nous nous attendons à ce que les taux de bonne classification obtenus par les scores SC^1, SC^2, SC^3 et SC^4 se situent entre les taux obtenus par le score Laplacien non supervisé et ceux obtenus par le score supervisé de Fisher. Comme le nombre $|\mathscr{S}_q|$ de contraintes est fixé à 10, les scores semi-supervisés utilisent un peu plus de connaissance a priori que le score Laplacien non supervisé mais beaucoup moins que le score supervisé de Fisher. Cette attente est vérifiée pour la base 'ORL', puisque d'après la figure 4.1(d), quelque soit le nombre d'attributs sélectionnés les taux de bonne classification des scores de contraintes sont supérieurs à ceux du score Laplacien. Cependant pour les bases 'Wine', 'Image segmentation' et 'Vehicle', les performances des scores de contraintes dépassent ceux du score Laplacien, pour respectivement, 7 dimensions sur 13 (voir figure 4.1(a)), 8 dimensions sur 19 (voir figure 4.1(b)) et 10 dimensions sur 18 (voir figure 4.1(c)). En effet, les paramètres du score Laplacien ont été bien ajustés pour chacune des bases afin de représenter au mieux la distribution des données dans les différentes classes. Ainsi le paramètre de dispersion de la similarité Gaussienne σ est respectivement fixé à 0.7, 1, 0.5 et 5 pour les bases 'Wine', 'Image segmentation', 'Vehicle' et 'ORL'.

Il est vrai qu'un algorithme de classification basé sur un apprentissage supervisé doit atteindre des performances plus élevées que le même basé sur un apprentissage semi-supervisé. De même, le fait de prendre en considération quelques contraintes doit permettre d'améliorer les résultats de classification comparés à ceux obtenus dans un contexte non-supervisé. Ces résultats confirmés sur la base 'ORL', le sont moins sur les bases 'Wine', 'Image segmentation' et 'Vehicle'. Cela permet alors de soulever la question sur le choix des différentes contraintes, ce

choix étant totalement aléatoire dans nos expériences.

4.2.2.1.b Comparaison des taux de bonne classification pour chaque sous-ensemble de contraintes

En examinant particulièrement les courbes de SC^1, SC^2, SC^3 et SC^4 de la figure 4.1, nous constatons qu'elles sont très proches. Il est donc difficile de comparer les performances de ces scores, surtout que ces résultats sont moyennés sur 100 exécutions avec différentes générations de contraintes.

Cela nous amène alors à comparer ces différents scores en examinant leurs résultats de classification sur chacune des 100 exécutions. Pour un nombre fixé d'attributs à sélectionner et pour chacune des 100 exécutions, nous proposons d'ordonner les 4 scores dans un ordre décroissant de leur taux de bonne classification.

Soit $rang_q^*$ le rang du critère SC^* à l'exécution q, ce rang pouvant prendre les valeurs 1, 2, 3 ou 4. Pour chaque exécution q, le rang 1 sera attribué à la méthode dont le taux de bonne classification est le plus grand et le rang 4 sera attribué à la méthode ayant le taux de bonne classification le plus petit. Nous attribuons le même rang aux méthodes ayant le même taux de bonne classification.

Nous calculons ainsi T^* qui totalise l'ensemble des rangs pour chacun des scores SC^* comme suit :

$$T^* = \sum_{q=1}^{100} rang_q^*, \tag{4.1}$$

où * est 1, 2, 3 ou 4 correspondant respectivement aux scores SC^1, SC^2, SC^3 ou SC^4. Le score ayant le total des rangs le plus petit est considérée comme celui qui fournit les meilleurs résultats.

Pour les bases 'Wine', 'Image segmentation', 'Vehicle' et 'ORL', les taux de bonne classification des données obtenus avec chacun des 4 critères de sélection SC^1, SC^2, SC^3 et SC^4 semblent

être stables lorsque le nombre d'attributs sélectionnés est respectivement supérieur à 6, 5, 8 et 300 (voir figure 4.1). Nous proposons alors de calculer le total des rangs de chacun de ces scores en considérant les 6 premiers attributs pour la base 'Wine', les 5 premiers attributs pour la base 'Image segmentation', les 8 premiers attributs pour la base 'Vehicle' et les 300 premiers pour la base 'ORL'.

$\lvert\mathscr{S}_q\rvert \backslash T$	T^1	T^2	T^3	T^4
4 contraintes	198	238	219	**180**
10 contraintes	195	210	185	**162**
40 contraintes	180	299	168	**136**

Tableau 4.1 : Le total des rangs des différents scores pour différents nombres $\lvert\mathscr{S}_q\rvert$ de contraintes pour la base de données 'Wine'.

$\lvert\mathscr{S}_q\rvert \backslash T$	T^1	T^2	T^3	T^4
4 contraintes	208	228	**168**	179
10 contraintes	228	210	**154**	183
40 contraintes	184	177	153	**146**

Tableau 4.2 : Le total des rangs des différents scores pour différents nombres $\lvert\mathscr{S}_q\rvert$ de contraintes pour la base de données 'Image segmentation'.

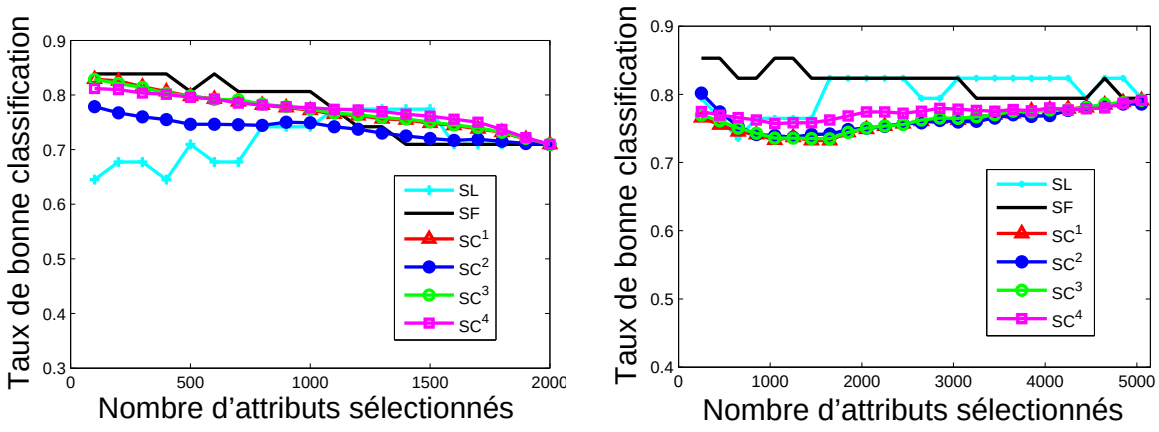
$\lvert\mathscr{S}_q\rvert \backslash T$	T^1	T^2	T^3	T^4
4 contraintes	237	239	232	**219**
10 contraintes	271	246	**189**	240
40 contraintes	290	240	208	**207**

Tableau 4.3 : Le total des rangs des différents scores pour différents nombres $\lvert\mathscr{S}_q\rvert$ de contraintes pour la base de données 'Vehicle'.

$\lvert\mathscr{S}_q\rvert \backslash T$	T^1	T^2	T^3	T^4
4 contraintes	**194**	212	311	200
10 contraintes	**185**	189	250	196
40 contraintes	**190**	202	300	203

Tableau 4.4 : Le total des rangs des différents scores pour différents nombres $\lvert\mathscr{S}_q\rvert$ de contraintes pour la base de données 'ORL'.

Les tableaux 4.1, 4.2, 4.3 et 4.4 montrent le total des rangs T^* pour différents nombres $\lvert\mathscr{S}_q\rvert$ de contraintes (4, 10 et 40) respectivement sur les bases 'Wine', 'Image segmentation', 'Vehicle' et 'ORL'. Ainsi, nous examinons le cas où peu de contraintes (4) sont retenues, où un nombre



(a) 'Colon Cancer'. (b) 'Leukemia'.

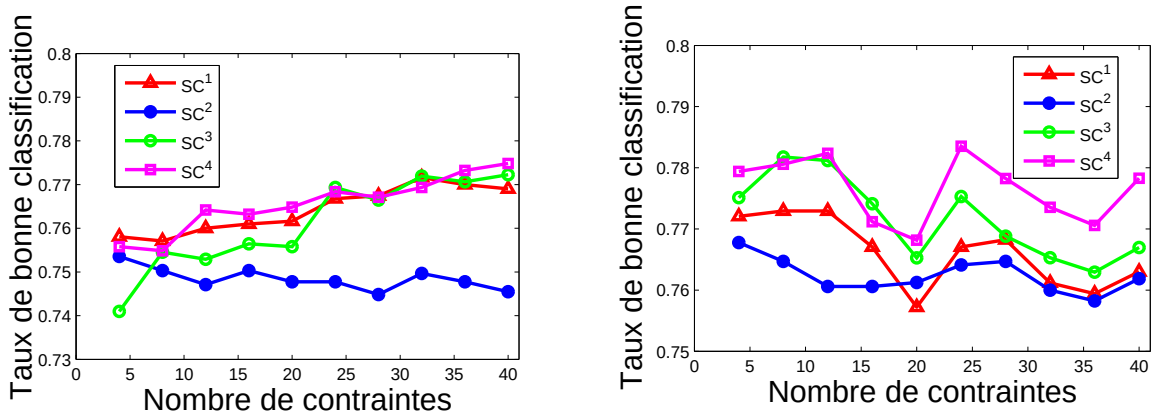
Figure 4.2 : Taux de bonne classification en fonction du nombre d'attributs sélectionnés sur les bases d'expression des gènes. 60 contraintes formées de 30 must-link et 30 cannot-link sont utilisées. L'évaluation est effectuée dans un contexte supervisé : l'algorithme du 1-ppv utilise les labels réels des données d'apprentissage comme prototypes des classes.

un peu plus important est considéré (10) et le cas où le nombre de contraintes au regard de la population de données est important (40). Chaque case du tableau indique le total des rangs du score testé en utilisant le nombre de contraintes considéré et nous notons en gras le total des rangs le plus faible pour chaque cas.

A partir de ces tableaux, nous pouvons remarquer que les valeurs de T^1, T^2, T^3 et T^4 sont proches. Les attributs sélectionnés par notre score SC^4 fournit des taux de bonne classification comparables à ceux obtenus par les attributs sélectionnés par SC^1, SC^2 et SC^3. En effet, notre score a le total de rangs T le plus petit 6 fois (indiquées en gras) parmi les 12 lignes des tableaux 4.1, 4.2, 4.3 et 4.4.

4.2.2.2 Résultats sur les bases d'expression de gènes

La figure 4.2 montre les taux de bonne classification en fonction de différents nombres d'attributs sélectionnés sur les bases 'Colon Cancer' et 'Leukemia'. Le nombre de contraintes est fixé à 60 ($\left|\mathscr{S}_q\right| = 60$) comme le font Sun et al. ([SZ10]). 30 contraintes must-link et 30 cannot-link sont alors fournies par l'expert. Notons que le terme σ utilisé pour le score Laplacien est respectivement fixé à 10 et à 15 pour ces deux bases.



(a) 'Colon Cancer'. (b) 'Leukemia'.

Figure 4.3 : Taux de bonne classification en fonction de différents nombres de contraintes (pour SC^1, SC^2, SC^3 et SC^4) sur les bases d'expression des gènes. Le nombre d'attributs sélectionnés est la moitié du nombre initial d'attributs.

A partir de la figure 4.2(a), nous pouvons remarquer que les résultats de classification obtenus par les scores SC^1, SC^2, SC^3 et SC^4 se situent entre les résultats de classification obtenus par le score Laplacien non-supervisé et ceux obtenus par le score supervisé de Fisher sur la base 'Colon Cancer'. Les résultats de classification obtenus grâce à ces scores dépassent même les résultats du score supervisé de Fisher pour un nombre d'attributs sélectionnés supérieur à 1050. Cependant, pour la base 'Leukemia' (voir figure 4.2(b)), les résultats de classification obtenus par le score Laplacien dépassent ceux obtenus par les scores de contraintes lorsque le nombre d'attributs est supérieur à 1050. Ces résultats confirment que les contraintes n'apportent pas toujours d'informations sur la discrimination des classes surtout si ces contraintes sont choisies aléatoirement.

La figure 4.3 affiche le taux de bonne classification en fonction du nombre de contraintes pour un nombre d'attributs fixe (fixé à la moitié du nombre d'attributs initial comme dans [SZ10]) sur les deux bases d'expression de gènes. A partir de cette figure, nous pouvons remarquer que notre score SC^4 atteint des performances supérieures à SC^1, SC^2 et SC^3. Nous pouvons noter aussi que le taux de bonne classification n'augmente pas toujours en fonction du nombre de contraintes puisque les différentes courbes de la figure 4.3 ne sont pas forcément croissantes. En effet, nous

pouvons remarquer dans la figure 4.3(b) que pour 20 contraintes, les performances des 4 scores subissent une chute brutale. Cela est probablement dû à un mauvais choix des contraintes. Ces résultats confirment alors notre hypothèse que le processus de sélection dépend fortement du sous-ensemble de contraintes fourni par l'expert (voir section 3.3.2.3). Cependant, nous notons que cette chute n'est pas importante (à peu près 2%) pour un total de 34 données de la base test. Comme le taux de bonne classification est moyenné sur 100 exécutions, il suffit alors qu'une donnée soit mal classifiée dans un certain nombre d'itérations pour que cette chute apparaisse.

$\lvert\mathscr{S}_q\rvert \backslash T$	T^1	T^2	T^3	T^4
4 contraintes	**148**	195	235	158
10 contraintes	156	242	203	**153**
40 contraintes	169	286	153	**138**
60 contraintes	157	271	149	**144**

Tableau 4.5 : Le total des rangs des différents scores pour différents nombres $\lvert\mathscr{S}_q\rvert$ de contraintes pour la base de données 'Colon Cancer'.

$\lvert\mathscr{S}_q\rvert \backslash T$	T^{21}	T^2	T^3	T^4
4 contraintes	184	240	194	**182**
10 contraintes	176	276	150	**134**
40 contraintes	170	196	148	**118**
60 contraintes	186	198	168	**126**

Tableau 4.6 : Le total des rangs des différents scores pour différents nombres $\lvert\mathscr{S}_q\rvert$ de contraintes pour la base de données 'Leukemia'.

En outre, les tableaux 4.5 et 4.6 montrent la somme des rangs T^* pour différents nombres $\lvert\mathscr{S}_q\rvert$ de contraintes (4, 10, 40 et 60) respectivement sur les bases 'Colon Cancer' et 'Leukemia'. Le total des rangs de chacun des scores est calculé en considérant la moitié du nombre initial d'attributs de chacun des bases d'expression des gènes.

A partir de ces deux tableaux, nous pouvons remarquer que, pour la base de données 'Colon Cancer', le total des rangs T^4 relatif à notre score de contraintes SC^4 a la valeur la plus faible 3 fois (indiquées en gras) parmi les 4 lignes du tableau 4.5 (lorsque le nombre de contraintes est supérieur à 4). Pour la base 'Leukemia', T^4 a la valeur la plus faible (indiquée en gras) pour tous les nombres de contraintes (4, 10, 40 et 60). Ces résultats montrent alors que les attributs

sélectionnés par notre score SC^4 fournissent des résultats de classification supérieurs à ceux obtenus par les attributs sélectionnés par SC^1, SC^2 et SC^3.

4.2.3 Discussion

Nous avons comparé les résultats de classification des données test en utilisant les attributs sélectionnés par les différents scores. Cette classification est réalisée à l'aide du plus proche voisin qui utilise les données d'apprentissage comme prototypes des classes.

Cependant, les labels des données n'ont été exploités que par le score supervisé de Fisher pour évaluer la pertinence des attributs. Comme le score Laplacien effectue la sélection dans un contexte non supervisé, il suppose l'absence totale d'information sur les labels des données. Les scores SC^1, SC^2, SC^3 et SC^4 utilisent juste quelques contraintes entre données et donc aucune information sur les labels des données d'apprentissage.

Nous rappelons que les données test ont été classifiées dans un contexte supervisé en utilisant les données d'apprentissage comme prototype de classes, alors que les attributs sont sélectionnés dans un contexte non-supervisé ou semi-supervisé en utilisant juste quelques contraintes must-link et cannot-link.

Pour comparer les performances des différents scores de sélection d'attributs opérant dans un contexte non-supervisé ou semi-supervisé, les données test doivent alors être classifiées dans le même contexte non-supervisé ou semi-supervisé. Nous proposons donc deux procédures d'évaluation, une non supervisée pour le score Laplacien et une semi-supervisée pour les scores de contraintes, de telle sorte que la procédure de sélection et la procédure de classification soient menées dans le même contexte. Ces résultats permettront de juger de la qualité des attributs sélectionnés dans un cadre proche d'une application réelle.

4.3 Evaluation selon le contexte

Dans cette partie, l'évaluation de la qualité des scores sera menée selon les contextes non-supervisé et semi-supervisé.

4.3.1 Evaluation non-supervisée

Le classifieur du plus proche voisin nécessite la définition des prototypes des classes. Ceux-ci seront construits à partir des données d'apprentissage. Or, dans un contexte d'apprentissage non-supervisé, l'utilisateur ne dispose pas d'information sur les labels des données d'apprentissage. Pour respecter ce contexte, il est nécessaire d'estimer les labels de ces données d'apprentissage afin de définir les prototypes des classes.

Dans le contexte non-supervisé, un algorithme de classification simple, à savoir celui des k-means, peut être utilisé pour classifier les données d'apprentissage [Llo82].

Les labels des données d'apprentissage étant alors estimés, nous pourrons alors les utiliser comme prototypes de classes. Pour l'étape de validation, le classifieur du plus proche voisin est alors appliqué sur la base test en utilisant ces prototypes estimés. Cependant, comme les classes réelles des données d'apprentissage peuvent être différentes de celles estimées par l'algorithme des k-means, nous ne pourrons pas utiliser directement ces labels. Par conséquent, avant de les utiliser par l'algorithme du plus proche voisin, nous devons mettre en correspondance les labels des données d'apprentissage et leurs labels estimés. Pour ce faire, nous avons utilisé une version de l'algorithme de Carpaneto et al. [CT80].

Avant de présenter les résultats sur différentes bases de données, nous allons décrire l'algorithme des k-means.

4.3.1.1 L'algorithme des k-means

L'algorithme des k-means [Mac67] est un algorithme de classification itératif basé sur le calcul de distances entre les points et les centres des classes et dont le but est de minimiser la

somme des inerties intra-classes et maximiser la somme des inerties inter-classes.

Les principales étapes de l'algorithme des k-means sont les suivantes :

1. L'algorithme choisit aléatoirement *k* points comme étant les centres des classes dans l'espace des attributs sélectionnés.
2. Les points restants sont assignés à la classe dont le centre est le plus proche.
3. L'algorithme met ensuite à jour le centre de chaque classe ainsi obtenue.
4. Les instructions 2 et 3 sont réitérées jusqu'à ce que les centres des classes ne varient plus au sens d'un critère d'arrêt.

4.3.1.2 Résultats de comparaison de l'évaluation supervisée et de l'évaluation non-supervisée du score Laplacien

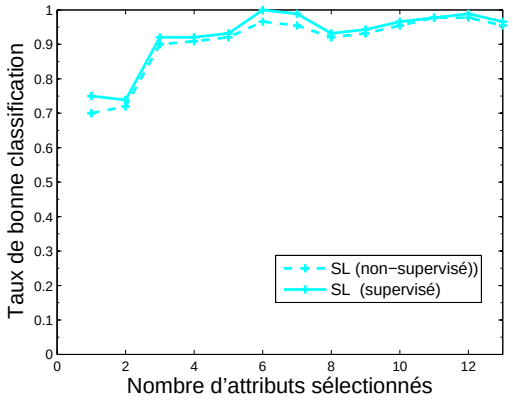
Dans cette partie, nous comparons les deux méthodes d'évaluation du score Laplacien sur les bases de données utilisées précédemment.

Nous comparons la méthode d'évaluation supervisée classique qui utilise l'algorithme du 1-ppv avec les données d'apprentissage comme prototypes des classes et la méthode d'évaluation non-supervisée, proposée dans la section 4.3.1, qui utilise l'algorithme des k-means afin d'estimer les labels des données d'apprentissage. Ces labels estimés seront ensuite utilisés par l'algorithme du 1-ppv afin de calculer les taux de bonne classification des données test.

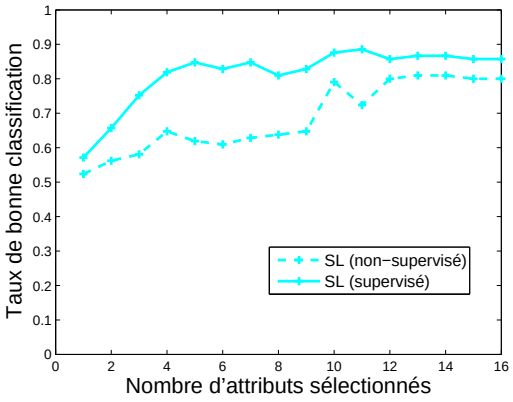
Pour évaluer le score Laplacien non-supervisé, l'algorithme des k-means opère dans l'espace des attributs sélectionnés afin d'estimer les labels des données d'apprentissage.

La figures 4.5 montre les résultats de comparaison des taux de bonne classification en utilisant les attributs sélectionnés par le score Laplacien sur respectivement 'Wine','Image segmentation', 'Vehicle', 'ORL', 'Colon cancer' et 'Leukemia'.

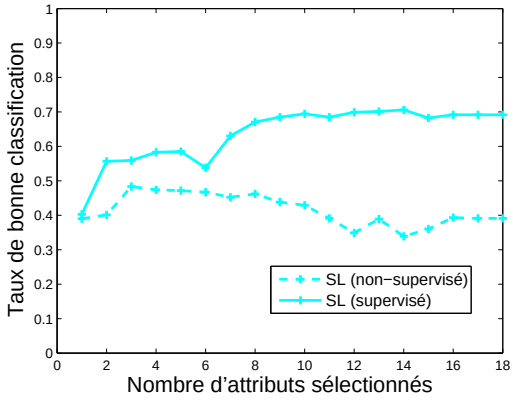
D'après cette figure, nous pouvons remarquer qu'il existe un écart entre les taux de bonne classification des deux méthodes d'évaluation. Cet écart est fortement variable entre les différents bases de données. Les écarts moyens entre les deux méthodes d'évaluation sur ces bases sont



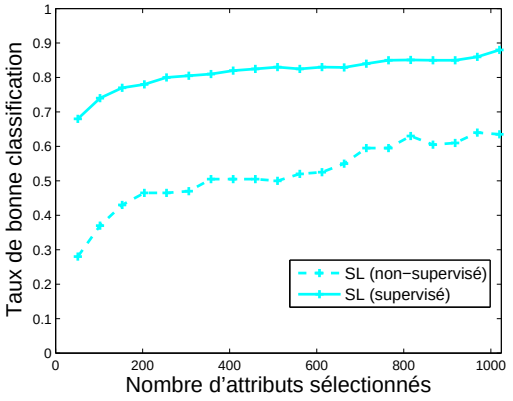
(a) 'Wine'



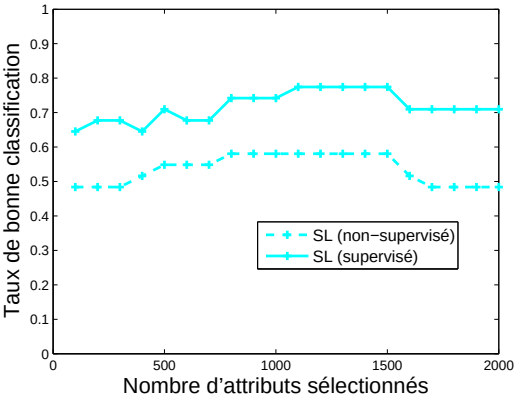
(b) 'Image segmentation'



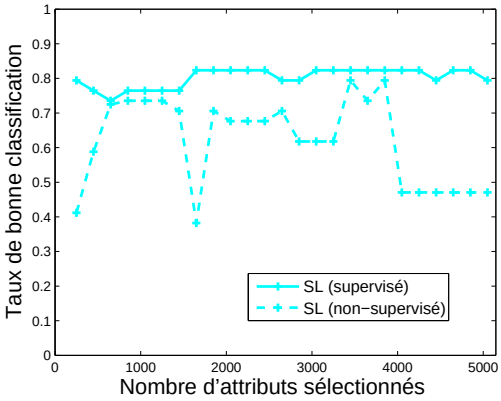
(c) 'Vehicle'



(d) 'ORL'



(e) 'Colon Cancer'



(f) 'Leukemia'

Figure 4.4 : Comparaison entre les taux de bonne classification en utilisant la méthode d'évaluation supervisée et ceux obtenus en utilisant la méthode non-supervisée en fonction du nombre d'attributs sélectionnés par le score Laplacien *SL* pour les base 'Wine', 'Image segmentation', 'Vehicle', 'ORL', 'Colon Cancer' et 'Leukemia'

respectivement 1.82%, 12.74%, 21.23%, 27.9%, 19.2% et 16.12%. En effet, il est tout à fait normal que les taux de bonne classification des données test en utilisant une évaluation supervisée soient supérieurs à ceux obtenus en utilisant une évaluation non-supervisée. Il est toutefois plus réaliste de réaliser la sélection et la classification dans le même contexte.

4.3.2 Evaluation semi-supervisé

Dans un contexte semi-supervisé, l'expert fournit une information partielle sous forme de contraintes must-link et cannot-link. En présence de ces contraintes, nous proposons d'utiliser l'algorithme des k-means sous contraintes développé par Wagstaff et al., afin d'estimer les labels des données d'apprentissage [WCR+01]. Cet algorithme va opérer dans l'espace d'attributs considéré. Ensuite, ces labels estimés seront utilisés comme prototype de classes par le classifieur du plus proche voisin afin de classifier les données test. Comme pour l'évaluation non-supervisée, nous réalisons une étape de mise en correspondance entre les labels réels et labels estimés des données d'apprentissage en utilisant l'algorithme de Carpaneto et al. [CT80]. Avant de présenter les résultats sur différentes bases de données, nous allons décrire l'algorithme des k-means sous contraintes de Wagstaff et al. [WCR+01].

4.3.2.1 L'algorithme des k-means sous-contraintes

L'algorithme des k-means sous-contraintes de Wagstaff et al. [WCR+01] est une modification de l'algorithme classique des k-means afin de respecter les contraintes must-link $\mathscr{M}$ et cannot-link $\mathscr{C}$ fournis par l'expert.

L'algorithme se décompose selon les étapes successives suivantes :

1. L'algorithme choisit aléatoirement *k* points comme étant les centres des classes dans l'espace des attributs sélectionnés.
2. Les points restants sont assignés à la classe dont le centre est le plus proche **tout en vérifiant que l'ensemble des contraintes must-link $\mathscr{M}$ et cannot-link $\mathscr{C}$ ne sont pas**

violées.

3. L'algorithme met ensuite à jour le centre de chaque classe ainsi obtenue.
4. Les opérations 2 et 3 sont réitérées jusqu'à ce que les centres des classes ne varient plus au sens d'un critère d'arrêt.

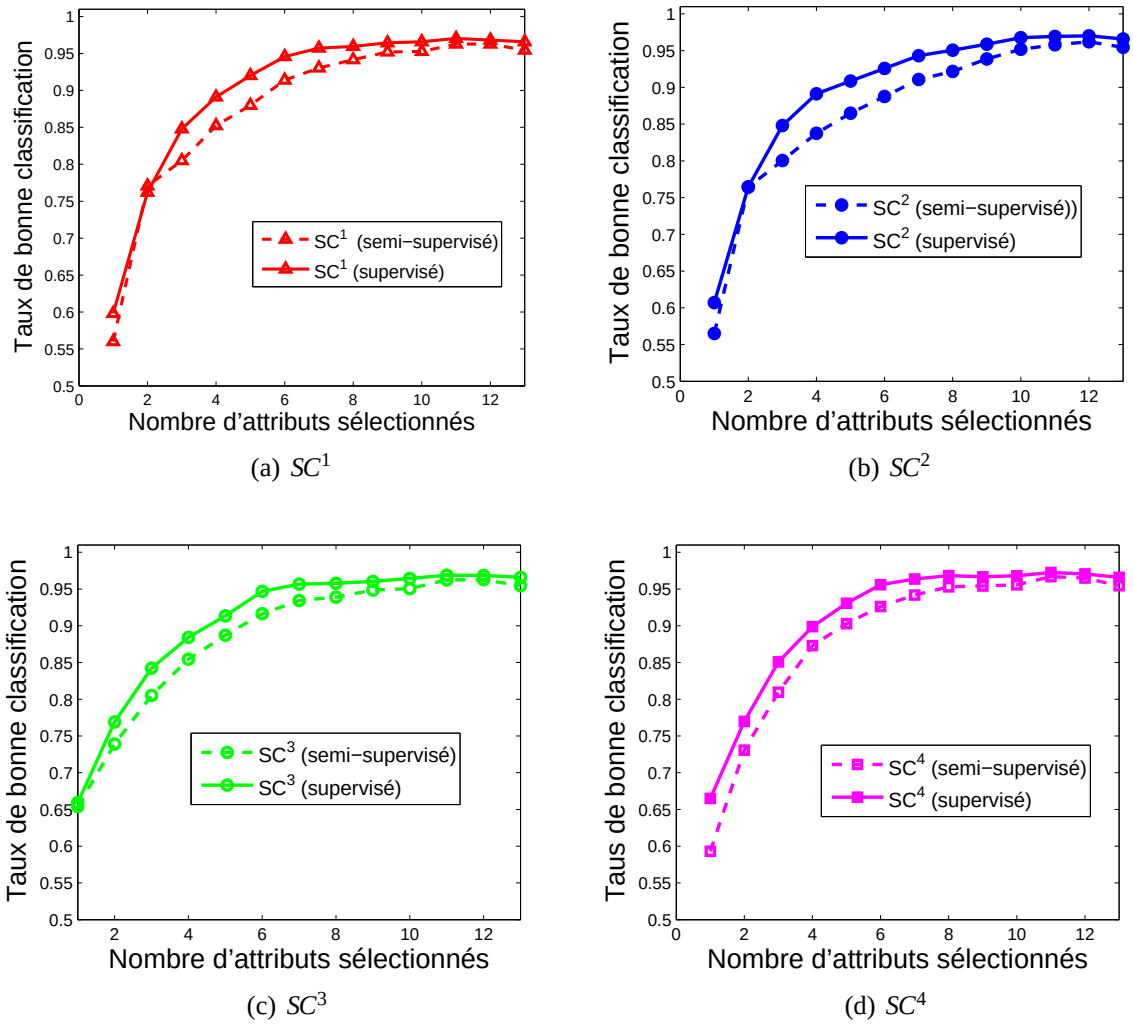
La modification majeure indiquée en gras consiste à s'assurer, au moment de l'assignation des données aux différentes classes, que l'ensemble des contraintes sont effectivement respectées. Dans le cas contraire, Wagstaff et al. n'assigne la donnée correspondante à aucune classe. Cependant, nous choisissons de forcer le respect des différentes contraintes afin d'estimer les labels de toutes les données d'apprentissage.

4.3.2.2 Résultats de comparaison de l'évaluation supervisée et de l'évaluation semi-supervisée des scores de contraintes

Dans cette partie, nous appliquons les deux méthodes d'évaluation des divers scores de contraintes SC^1, SC^2, SC^3 et SC^4 aux bases de données.

Nous comparons la méthode d'évaluation supervisée classique qui utilise l'algorithme du 1-ppv avec les données d'apprentissage comme prototypes des classes et la méthode d'évaluation semi-supervisée, proposée dans la section 4.3.2, qui utilise l'algorithme des k-means sous contraintes afin d'estimer les labels des données d'apprentissage. Ces labels estimés seront ensuite utilisés par l'algorithme du plus proche voisin afin de calculer les taux de bonne classification des données test.

Pour les scores de contraintes SC^1, SC^2, SC^3 et SC^4, le taux de bonne classification obtenu par chacun des scores est moyenné sur 100 exécutions avec différentes générations des contraintes. Dans notre procédure d'évaluation, nous utilisons l'algorithme des k-means sous contraintes qui opère dans l'espace considéré afin de labelliser les données d'apprentissage tout en garantissant le respect des différentes contraintes. Ces labels estimés seront ensuite utilisés comme prototypes de classes par l'algorithme du plus proche voisin afin de calculer le taux de bonne



(a) SC^1 (b) SC^2

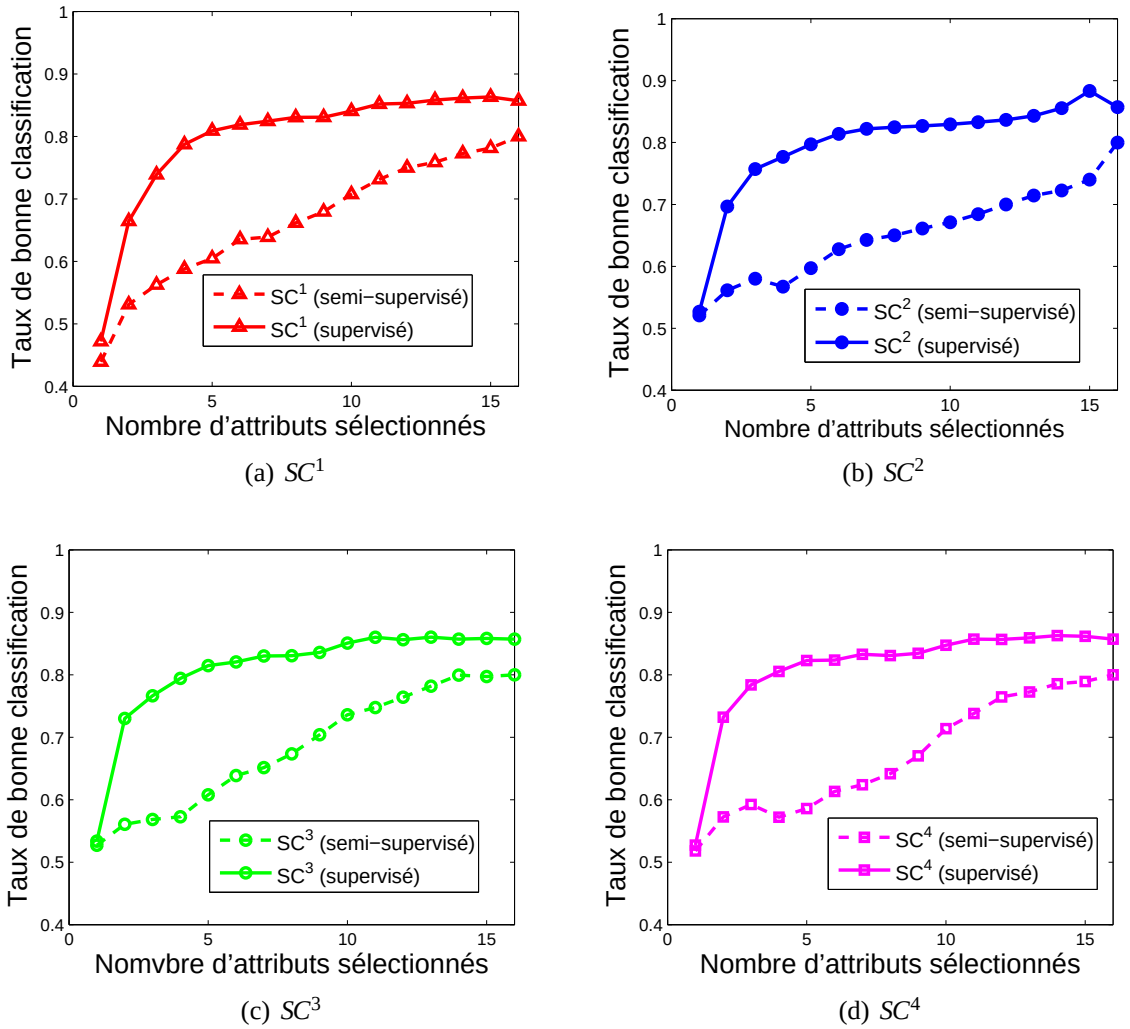
(c) SC^3 (d) SC^4

Figure 4.5 : Comparaison entre les taux de bonne classification en utilisant la méthode d'évaluation supervisée et ceux obtenus en utilisant la méthode semi-supervisée pour SC^1, SC^2, SC^3 et SC^4 en fonction du nombre d'attributs sélectionnés pour la base 'Wine'. 10 contraintes formées de 5 must-link et 5 cannot-link sont utlisées.

classification des données test.

Les figures 4.5, 4.6, 4.7, 4.8, 4.9 et 4.10 montrent les résultats de comparaison en utilisant les attributs sélectionnés par les différents scores SC^1, SC^2, SC^3 et SC^4 sur respectivement les bases 'Wine','Image segmentation', 'Vehicle', 'ORL', 'Colon cancer' et 'Leukemia'.

D'après ces différentes figures, nous pouvons remarquer l'écart entre les taux de bonne classification des deux méthodes d'évaluation des scores de contraintes SC^1, SC^2, SC^3 et SC^4. Les écarts moyens entre les deux méthodes d'évaluation des différents scores sur les différentes bases sont résumés dans le tableau 4.7. D'après ce tableau nous pouvons remarquer que l'écart



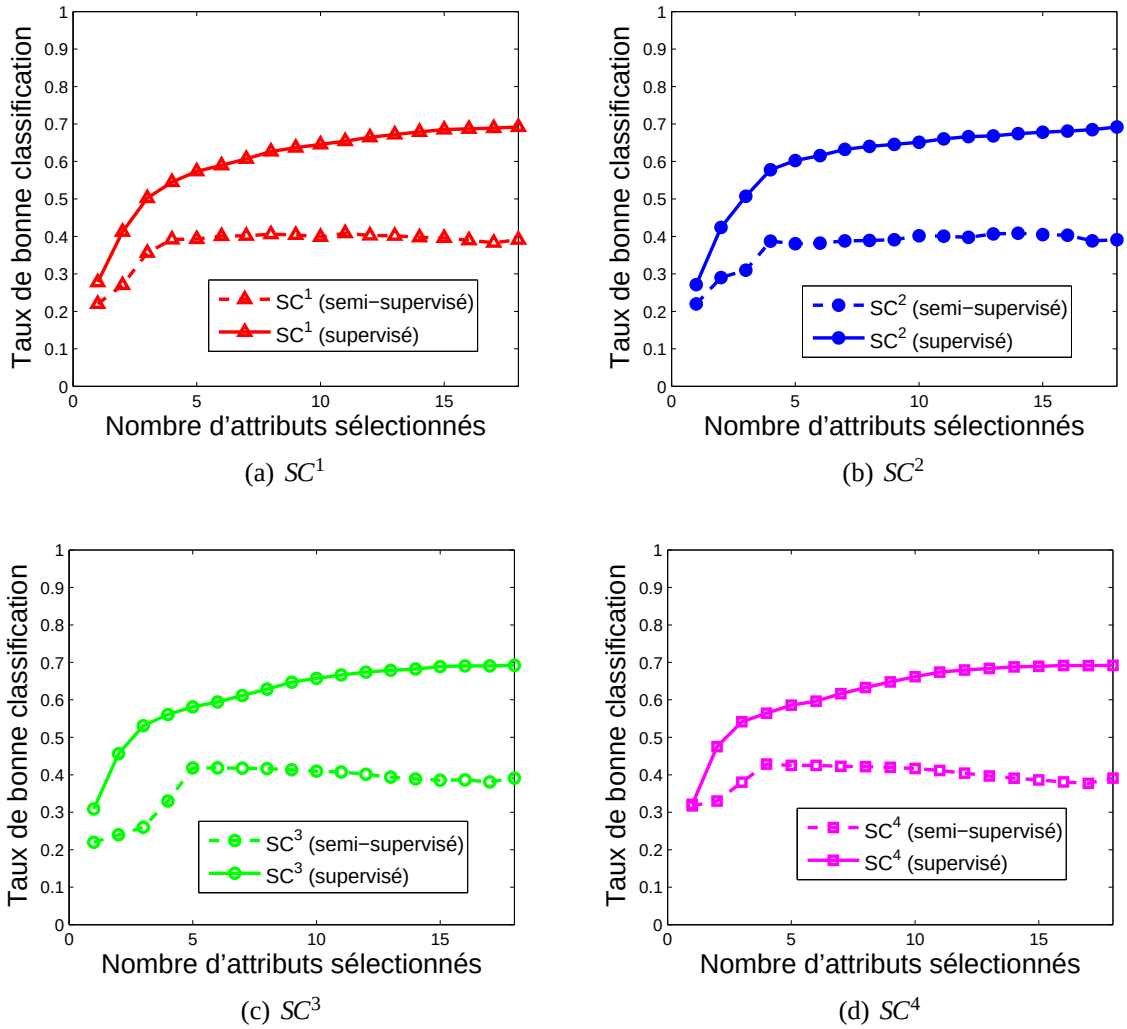
(a) SC^1 (b) SC^2

(c) SC^3 (d) SC^4

Figure 4.6 : Comparaison entre les taux de bonne classification en utilisant la méthode d'évaluation supervisée et ceux obtenus en utilisant la méthode semi-supervisée pour SC^1, SC^2, SC^3 et SC^4 en fonction du nombre d'attributs sélectionnés pour la base 'Image segmentation'. 10 contraintes formées de 5 must-link et 5 cannot-link sont utlisées.

est vraiment variable entre les différentes bases de données. Cependant pour une même base de données, cet écart est très proche entre les différents scores.

On peut noter que l'algorithme des k-means sous contraintes n'est pas l'algorithme qui fournit des performances optimales en présence de contraintes. Cependant, nous avons utilisé cet algorithme simple afin de montrer la différence entre les taux de bonne classification selon que les labels des données d'apprentissage sont estimés ou non.



(a) SC^1 (b) SC^2

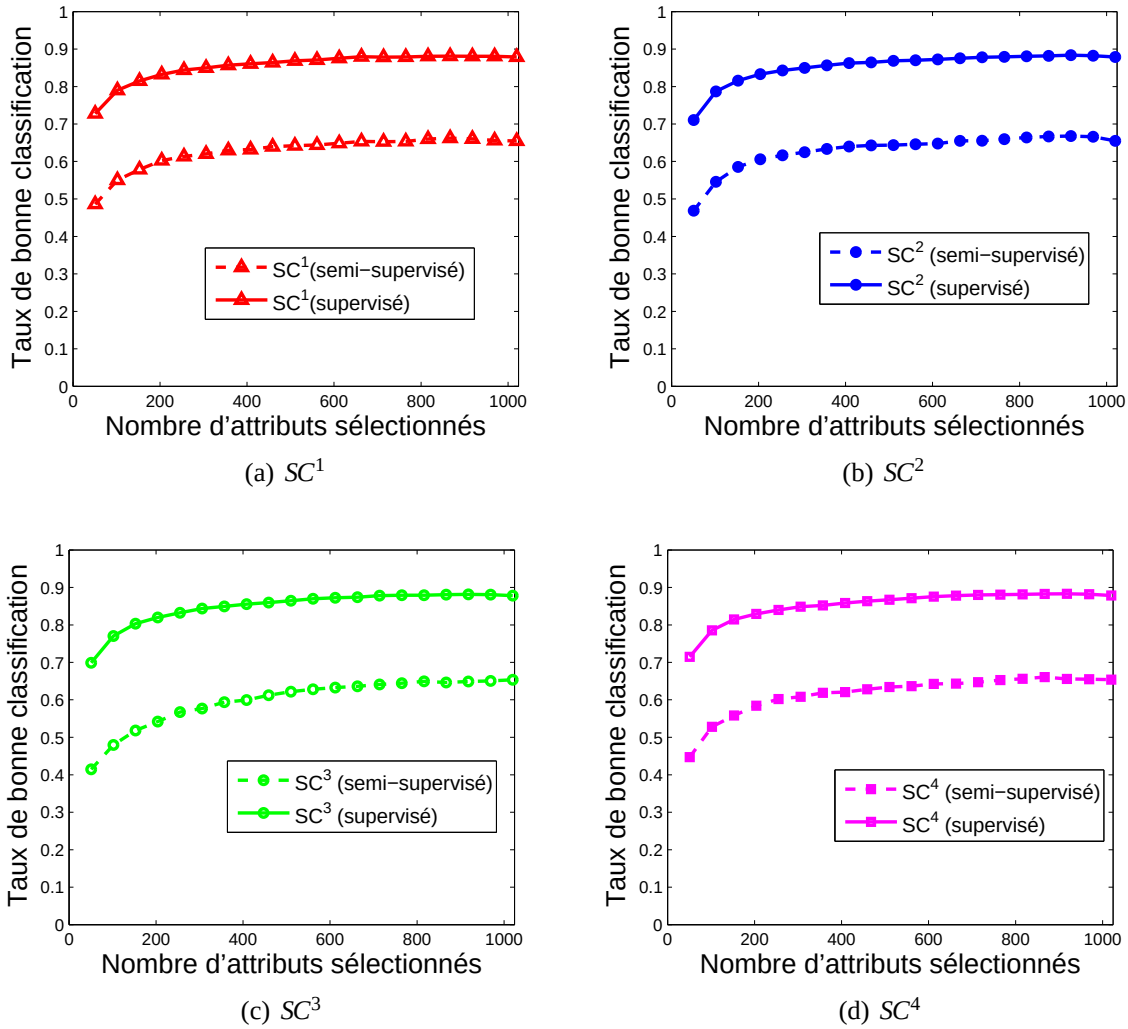
(c) SC^3 (d) SC^4

Figure 4.7 : Comparaison entre les taux de bonne classification en utilisant la méthode d'évaluation supervisée et ceux obtenus en utilisant la méthode semi-supervisée pour SC^1, SC^2, SC^3 et SC^4 en fonction du nombre d'attributs sélectionnés pour la base 'Vehicle'. 10 contraintes formées de 5 must-link et 5 cannot-link sont utlisées.

4.3.2.3 Comparaison des résultats de classification des scores de contraintes en utilisant une évaluation semi-supervisée

Afin de comparer les résultats de classification des scores SC^1, SC^2, SC^3 et SC^4 en utilisant une évaluation semi-supervisée, la figure 4.11 montre les taux de bonne classification des scores SC^1, SC^2, SC^3 et SC^4 en fonction de différents nombres d'attributs sélectionnés sur les bases 'Wine','Image segmentation', 'Vehicle', 'ORL', 'Colon cancer' et 'Leukemia'.

A partir de cette figure, nous pouvons remarquer que les résultats de classification obtenus par les scores SC^1, SC^2, SC^3 et SC^4 sont très proches, les différentes courbes sont voisines. Comme



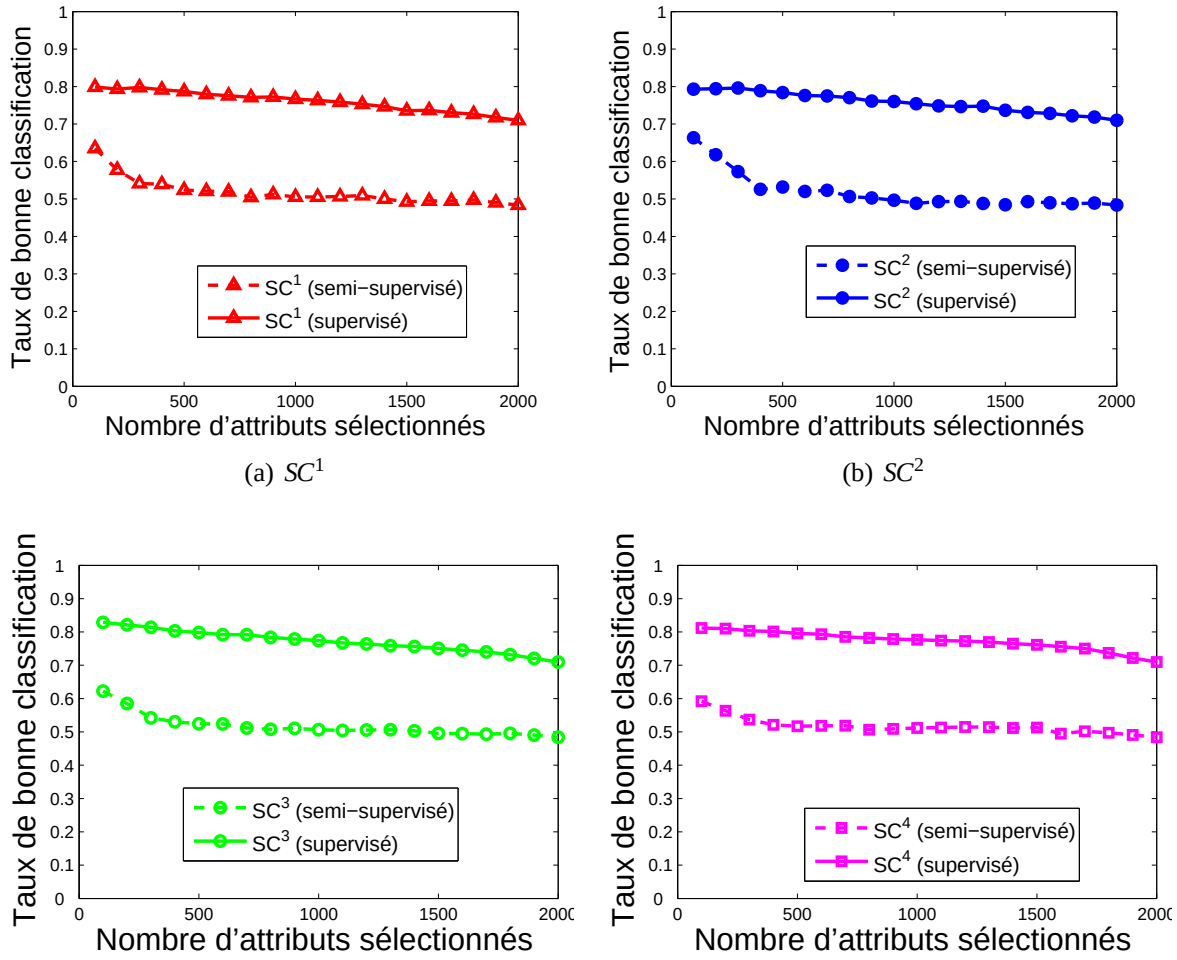
(a) SC^1 (b) SC^2

(c) SC^3 (d) SC^4

Figure 4.8 : Comparaison entre les taux de bonne classification en utilisant la méthode d'évaluation supervisée et ceux obtenus en utilisant la méthode semi-supervisée pour SC^1, SC^2, SC^3 et SC^4 en fonction du nombre d'attributs sélectionnés pour la base 'ORL'. 60 contraintes formées de 30 must-link et 30 cannot-link sont utlisées.

ces résultats sont moyennés sur 100 exécutions, il est difficile de les comparer.

En outre, le tableau 4.8 montre la somme des rangs T^* pour les différentes bases de données 'Wine', 'Image segmentation', 'Vehicle', 'ORL', 'Colon Cancer' et 'Leukemia'. Le total des rangs de chacun des scores est calculé en considérant le même nombre d'attributs que nous avons considéré pour l'évaluation dans le contexte supervisé, soit respectivement 6, 5, 8, 300, 1000 et 2576 attributs. Rappelons que pour les bases 'Wine', 'Image segmentation', 'Vehicle' et 'ORL', 10 contraintes (5 must-link et 5 cannot-link) ont été prises en compte et pour les bases 'Colon Cancer' et 'Leukemia', il s'agit de 60 contraintes (30 must-link et 30 cannot-link).

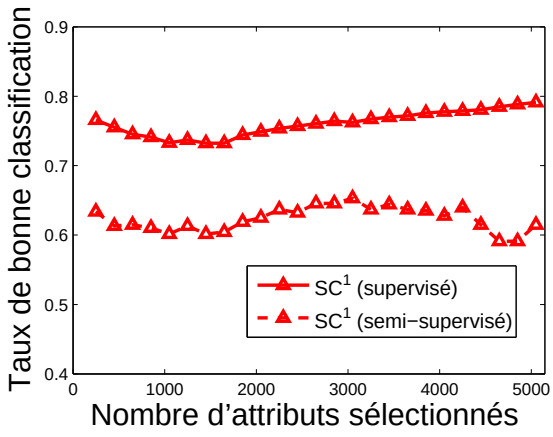


(a) SC^1 (b) SC^2

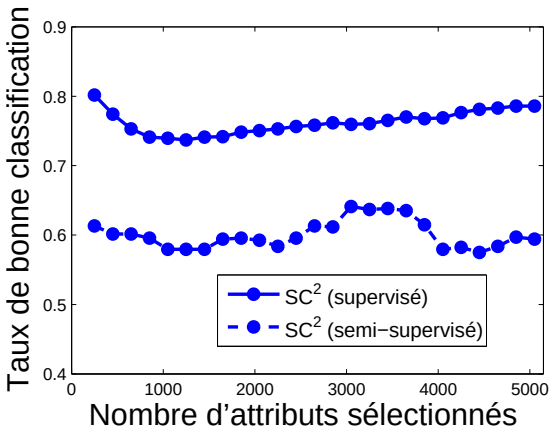
(c) SC^3 (d) SC^4

Figure 4.9 : Comparaison entre les taux de bonne classification en utilisant la méthode d'évaluation supervisée et ceux obtenus en utilisant la méthode semi-supervisée pour SC^1, SC^2, SC^3 et SC^4 en fonction du nombre d'attributs sélectionnés pour la base 'Colon Cancer'. 60 contraintes formées de 30 must-link et 30 cannot-link sont utlisées.

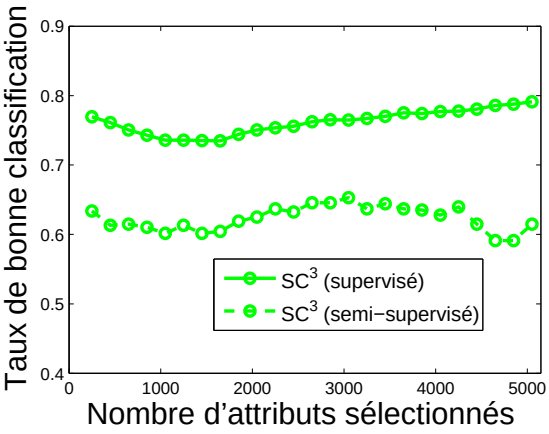
Notre score SC^4 a la valeur la plus faible 3 fois (indiqué en gras) parmi les 6 lignes du tableau 4.8. Les trois autres scores se partagent chacun une ligne. Cela confirme alors que les attributs sélectionnés par le score SC^4 fournissent des résultats de classification supérieurs à ceux des attributs sélectionnés par les scores existants en utilisant aussi bien une évaluation supervisée q'une évaluation semi-supervisée.



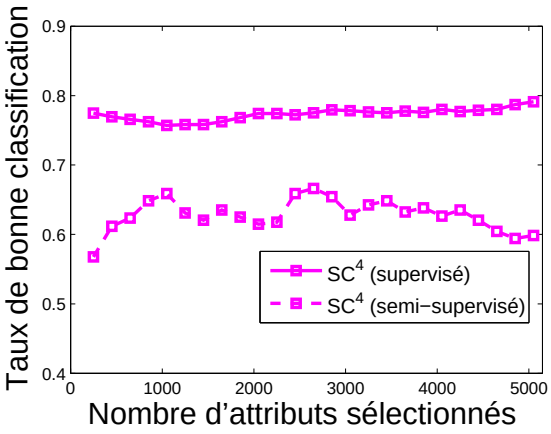
(a) SC^1



(b) SC^2



(c) SC^3



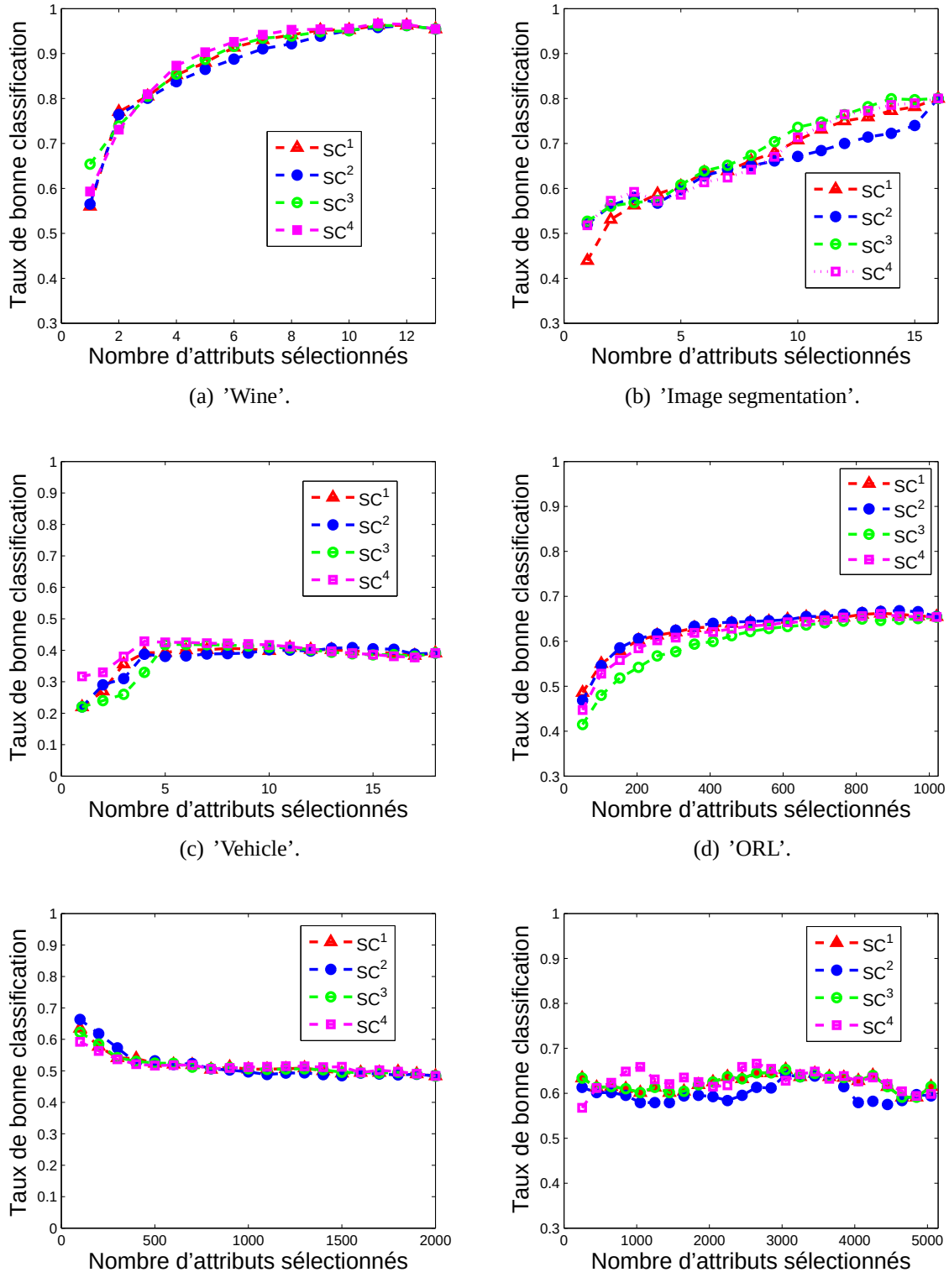
(d) SC^4

Figure 4.10 : Comparaison entre les taux de bonne classification en utilisant la méthode d'évaluation supervisée et ceux obtenus en utilisant la méthode semi-supervisée pour SC^1, SC^2, SC^3 et SC^4 en fonction du nombre d'attributs sélectionnés pour la base 'Leukemia'. 60 contraintes formées de 30 must-link et 30 cannot-link sont utlisées.

4.4 Conclusion

Les différents scores de sélection sont généralement comparés en évaluant le taux de bonne classification des données test. Pour ce faire, le classifieur du plus proche voisin opère dans l'espace composé des attributs sélectionnés en utilisant les labels des données d'apprentissage comme prototypes des classes.

Dans ce chapitre, nous appliquons cette procédure d'évaluation afin de comparer les scores de contraintes SC^1, SC^2, SC^3 et SC^4 sur différentes bases de données de référence. Les résultats ont montré que le taux de bonne classification des données test obtenu dans l'espace original des



(a) 'Wine'. (b) 'Image segmentation'.

(c) 'Vehicle'. (d) 'ORL'.

(e) 'Colon Cancer'. (f) 'Leukemia'.

Figure 4.11 : Taux de bonne classification en fonction du nombre d'attributs sélectionnés sur les 6 bases de données. 10 contraintes formées de 5 must-link et 5 cannot-link sont utilisées pour les bases 'Wine', 'Image segmentation', 'Vehicle' et 'ORL' et 60 contraintes formées de 30 must-link et 30 cannot-link sont utilisées pour les bases 'Colon Cancer' et 'Leukemia'. L'évaluation est effectuée dans un contexte semi- supervisé : l'algorithme du 1-ppv utilise les labels estimés des données d'apprentissage comme prototypes des classes.

Base de données \ SC	SC^1	SC^2	SC^3	SC^4
'Wine'	2.14%	2.72%	1.928%	2.46%
'Image segmentation'	13.25%	14.62%	12.66%	14.01%
'Vehicle'	20.76%	21.79%	20.78%	20.94%
'ORL'	22.77%	22.41%	25.06%	23.66%
'Colon Cancer'	21.92%	20.53%	20.01%	20.33%
'Leukemia'	17.86%	19.28%	17.85%	15.71%

Tableau 4.7 : Les écarts moyens entre l'évaluation supervisée et l'évaluation semi-supervisée des différents scores de contraintes sur les différentes bases de données.

Base de données \ T	T^1	T^2	T^3	T^4
'Wine'	191	277	206	**157**
'Image segmentation'	165	219	**161**	196
'Vehicle'	261	239	238	**209**
'ORL'	230	**200**	264	228
'Colon Cancer'	150	216	162	**150**
'Leukemia'	**140**	240	140	176

Tableau 4.8 : Le total des rangs des différents scores pour différentes bases de données.

attributs est généralement plus faible que celui obtenu dans un sous-espace composé d'attributs sélectionnés. Ceci prouve que ces bases de données souffrent de la malédiction de la dimension, à savoir que les attributs sont soit peu pertinents, soit corrélés entre eux. Par conséquent, il est opportun de ne sélectionner que les plus pertinents.

Parmi les six bases utilisées, les bases 'Colon Cancer' et 'Leukemia' sont particulièrement intéressantes car le nombre de données d'apprentissage est nettement inférieur au nombre d'attributs. La réduction de la dimension de l'espace de représentation par une sélection des attributs est une étape nécessaire à l'analyse des données. Grâce à la prise en compte conjointe des contraintes et de la structure locale des données, notre score SC^4 permet d'obtenir des résultats de classification qui surpassent ceux des autres scores de contraintes pour ces deux bases.

Lors de ce chapitre, nous avons mis en évidence que généralement la classification des données test opère dans le contexte d'apprentissage supervisé, puisque les labels des données d'apprentissage sont utilisés comme prototypes des classes par le classifieur du plus proche voisin. Cependant, la sélection des attributs à partir des contraintes fournies par l'expert a été menée dans le contexte semi-supervisé où nous ne disposons pas d'information a priori sur les

labels des données d'apprentissage.

Nous proposons donc que le contexte dans lequel opère l'évaluation des performances soit le même que celui retenu pour sélectionner les attributs. Les connaissances a priori seront donc les mêmes pour la sélection des attributs à partir de l'analyse des données d'apprentissage et pour la classification des données test. En effet, un utilisateur suit une démarche similaire pour comparer l'efficacité de différents scores de contraintes dans le cadre de son application réelle.

Dans le cas où la connaissance a priori est composée uniquement de contraintes must-link et cannot-link, nous avons proposé que les labels des données d'apprentissage soient estimés par l'algorithme des k-means sous contraintes qui tente de fournir une partition respectant au maximum les contraintes. Cet algorithme n'est pas le plus performant puisqu'il ne garantit pas le respect de toutes les contraintes disponibles, surtout si deux contraintes portent sur une même donnée. Il serait alors intéressant d'utiliser un algorithme de classification qui permet de garantir le respect de toutes les contraintes fournies par l'expert. Les labels estimés des données d'apprentissage sont ensuite utilisés comme prototypes des classes par le classifieur du plus proche voisin afin de classer les donnés test. Nous avons appelé cette démarche, évaluation semi-supervisée pour la différencier de l'évaluation supervisée qui est classiquement suivie.

La comparaison entre les évaluations supervisée et semi-supervisée des scores de contraintes montre que le taux de bonne classification des données test obtenu de manière supervisée est toujours supérieur à celui obtenu de manière semi-supervisée, avec un écart moyen variant selon les bases de données entre environ 2% et 28%. Les performances atteintes par les scores de contraintes sont donc fortement impactées par le mode d'évaluation retenue.

Le classement relatif entre les scores de contraintes à partir des performances peut également varier selon le mode d'évaluation. Par exemple, notre score SC^4 qui est le meilleur pour la base 'Leukemia' quelque soit le nombre de contraintes (cf. tableau 4.6) selon l'évaluation supervisée se retrouve en deuxième position selon l'évaluation semi-supervisée (cf.tableau 4.8). Toutefois, notre score SC^4 atteint des performances globalement supérieures à celles des autres scores de

contraintes, quelque soit le mode d'évaluation retenu.

Dans le chapitre suivant, nous proposons d'appliquer la sélection d'attributs grâce aux scores de contraintes à un problème d'analyse d'images, à savoir la classification de textures couleur.

Chapitre 5

Sélection d'attributs pour la classification de textures couleur

5.1 Introduction

Les algorithmes de classification d'images représentant des textures analysent généralement les attributs caractérisant les textures en présence.

Dans le cadre de la classification d'images de texture couleur, l'exploitation de la seule information de luminance est souvent insuffisante pour discriminer les textures. En effet, plusieurs auteurs ont montré que l'exploitation de la couleur permet d'améliorer les résultats en termes de classification de textures, comparativement à une simple analyse en niveaux de gris.

En effet, différents auteurs ont montré l'apport de la couleur à la classification d'images de texture. Van den Broek, ainsi que Hernandez montrent que l'exploitation de la couleur permet d'améliorer les résultats de classification de textures [BR04], [HCG$^+$05].

Palm et Porebski et al. ont montré que les attributs d'Haralick extraits des matrices de co-occurrences sont des attributs pertinents qui permettent de discriminer différentes classes de texture couleur [Pal04] [Por09] [BR04]. C'est pour cette raison que nous avons choisi ces attributs afin de caractériser les différentes images de textures.

La couleur des pixels peut être codée dans différents espaces couleur qui respectent des propriétés colorimétriques spécifiques. Porebski et al. ont montré que l'analyse des attributs d'Haralick

extraits des matrices de cooccurrences calculées sur des images codées dans plusieurs espaces permet d'améliorer les performances de classification [PVM08].

Cependant, le fait d'extraire des attributs d'images codées dans plusieurs espaces couleur augmente le nombre d'attributs. Comme ce nombre élevé peut dégrader la qualité de discrimination vis-à-vis des classes en présence, il est nécessaire de sélectionner les attributs les plus discriminants.

Le but de ce chapitre sera alors d'appliquer les différents scores de sélection semi-supervisée à la sélection d'attributs de textures couleur. Ainsi, nous commençons par une définition des attributs de textures utilisés. Il s'agit des attributs d'Haralick extraits des matrices de co-occurrences. Ensuite, nous exposons les différentes bases de textures de référence avec lesquelles nous avons mené nos expériences. Nous comparons alors les performances obtenues avec les scores de contraintes sur ces bases. Cette comparaison est basée sur le coefficient de Kendall et les taux de bonne classification des images test. Nous menons cette évaluation dans un contexte supervisé ainsi que dans un contexte semi-supervisé cohérent avec celui de la sélection.

5.2 Exploitation de la couleur pour l'analyse de textures

Une image couleur numérique est une matrice de pixels. La couleur de chaque pixel peut être définie par trois composantes rouge, verte et bleue. Elle peut être également représentée dans différents espaces couleurs selon différentes propriétés physiques, physiologiques et psychovisuelles.

D'une manière générale, la couleur d'un pixel est représentée par trois composantes notées C_1, C_2 et C_3. Ces trois composantes forment un espace vectoriel d'origine O appelé espace couleur et noté (C_1,C_2,C_3). Dans cet espace, la couleur d'un pixel donne naissance à un point C dont les coordonnées sont les niveaux des composantes C_1, C_2 et C_3. La figure 5.1 illustre ce propos.

Dans la littérature, il existe un grand nombre d'espaces couleur [TFMB04]. Chacun a ses pro-

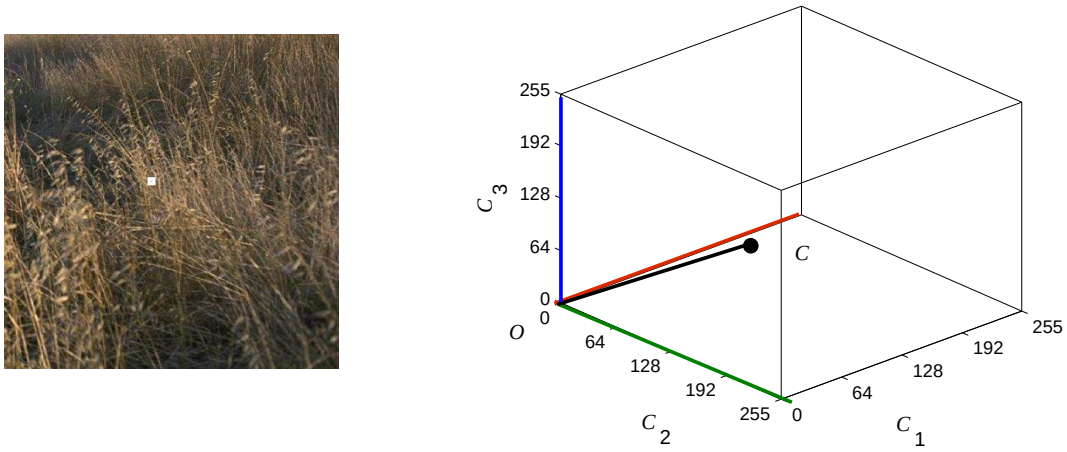


Figure 5.1 : Le pixel encadré en blanc dont les composantes couleur sont C_1, C_2 et C_3 donne naissance à un point C dans l'espace (C_1,C_2,C_3).

priétés spécifiques, ses avantages et ses inconvénients. Porebski a montré qu'il n'existe pas un unique espace couleur adapté à la classification de toutes les textures [Por09].

Au lieu de choisir un espace couleur spécifique, nous proposons alors de combiner les espaces (R,G,B), (H,S,V) et (L^*,a^*,b^*) afin d'améliorer les performances de classification des images couleur.

- L'espace (R,G,B) est l'espace couleur le plus utilisé, il est basé sur les 3 primaires : le rouge, le vert et le bleu. La reproduction de n'importe quelle couleur peut être obtenue par synthèse additive de ces 3 primaires.
- L'espace **(H,S,V)** est un espace perceptuel dont les acronymes sont : H : Hue, S : Saturation et V : Value. Dans cet espace, la couleur est décrite par rapport à la **teinte**, la **saturation** et la **luminosité**. La luminosité caractérise le niveau lumineux d'un stimulus de couleur. La teinte correspond aux dénominations des couleurs telles que rouge, vert, bleu, jaune, ... La saturation, elle, est une grandeur permettant d'estimer le niveau de coloration d'une teinte indépendamment de sa luminosité.
- L'espace (L^*,a^*,b^*) est un espace perceptuellement uniforme défini par le Commission Internationale de l'Eclairage (CIE). Les points correspondant à des stimuli de couleur sont contenus dans une sphère, comme l'illustre la figure 5.2. Dans le cas des espaces

perceptuellement uniformes, la composante de luminance est appelée **clarté**.

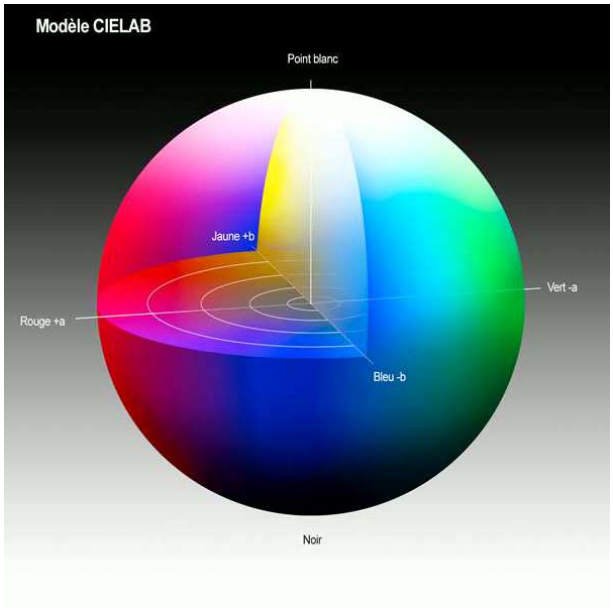


Figure 5.2 : Espace couleur (L^*,a^*,b^*).

Les transformations de l'espace RGB vers ces espaces sont présentées dans la thèse de Vandenbroucke [Van00] [TFMB04].

Ainsi, nous associons les informations provenant de ces différents espaces couleurs en codant les images de texture dans ces espaces et en calculant des attributs de textures extraits à partir des images ainsi codées. Ces espaces couleurs ont été retenus puisque la transformation RGB est non linéaire.

5.3 Attributs de textures couleur

Il existe un grand nombre d'attributs pouvant caractériser des textures couleur dans la littérature. Ces attributs sont regroupés en plusieurs catégories :

- les attributs géométriques [ZC06],
- les attributs basés sur la modélisation spatiale des textures [HK04] [BK98] [KH06],
- les attributs spatio-fréquentiels [TJ98] [Zen86]

  Nous citons juste ces catégories sans les détailler et le lecteur peut trouver plus de détails dans [Por09].

Porebski et al. [PVM09] ont montré que les attributs d'Haralick extraits des matrices de co-occurrences sont des attributs qui permettent de discriminer efficacement les différentes classes de textures couleur. C'est pour cette raison que nous avons choisi ces attributs afin de caractériser les différentes images de textures.

5.3.1 Matrices de co-occurrence

Les matrices de co-occurrences, introduites par Haralick en 1973, ont tout d'abord été implémentées en niveaux de gris [HSD73]. L'exploitation de la couleur permettant d'améliorer les résultats en termes de classification de textures, Palm propose d'étendre le concept de matrices de co-occurrences aux images couleur en définissant les matrices de co-occurrences chromatiques [Pal04]. Cet outil statistique est intéressant de part le fait qu'il mesure la distribution des composantes couleur dans l'image, tout en prenant en compte les interactions spatiales entre les pixels.

Expliquons le calcul de ces matrices pour une image **I** dont la couleur est codée dans un espace couleur (C_1,C_2,C_3). Considérons :

- $C_k, C_{k'} \in (C_1,C_2,C_3)$, deux des trois composantes couleur,
- et $M^{C_k,C_{k'}}[\mathbf{I}]$, la matrice de co-occurrences chromatique qui mesure l'interaction spatiale entre les composantes C_k et $C_{k'}$ des pixels voisins dans un voisinage 3×3 de l'image **I**

Le contenu de la cellule $M^{C_k,C_{k'}}[\mathbf{I}](i,j)$ de cette matrice indique le nombre de fois qu'un pixel P de l'image **I**, dont le niveau de composante couleur $C_k(P)$ est égal à i possède, dans son voisinage 3×3, un pixel P' dont le niveau de composante $C_{k'}(P')$ est égal à j.

Ces matrices de co-occurrences sont insensibles aux translations des objets dans les images. Comme elles mesurent les interactions locales entre les pixels, elles dépendent de la taille de l'image et sont sensibles à la résolution spatiale. Pour les rendre indépendantes de la taille de l'image, il est nécessaire de normaliser ces matrices par le nombre total de

co-occurrences dans la matrice considérée $\sum_{i=0}^{N-1}\ \sum_{j=0}^{N-1}\ M^{C_k,C_{k'}}[\mathbf{I}](i,j)$:

$$m^{C_k,C_{k'}}[\mathbf{I}]((a,b) = \frac{M^{C_k,C_{k'}}[\mathbf{I}](a,b)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} M^{C_k,C_{k'}}[\mathbf{I}](i,j)} \quad \forall (a,b) \in [0,\ldots,(N-1)] \tag{5.1}$$

où N est le niveau de quantification des composantes couleur.

Le niveau de quantification N des composants couleur est fixé à 32 afin de diminuer le coût de stockage en mémoire et le temps de calcul de ces matrices ainsi que des attributs extraits.

Une image couleur $\mathbf{I}$ dont la couleur est codée dans un espace couleur (C_1,C_2,C_3), est caractérisée par $N_M = 6$ matrices suivantes :

– trois matrices intra-composantes :

$m^{C_1,C_1}[\mathbf{I}]$, $m^{C_2,C_2}[\mathbf{I}]$ et $m^{C_3,C_3}[\mathbf{I}]$.

– trois matrices inter-composantes :

$m^{C_1,C_2}[\mathbf{I}]$, $m^{C_1,C_3}[\mathbf{I}]$ et $m^{C_2,C_3}[\mathbf{I}]$.

5.3.2 Attributs d'Haralick extraits des matrices de co-occurrences

Comme les matrices de co-occurrences contiennent beaucoup d'informations et sont donc consommatrices en espace mémoire, elles ne sont pas directement exploitées pour caractériser les textures couleur. Les utilisateurs préfèrent donc extraire de ces matrices des attributs afin de réduire la quantité d'informations à manipuler, tout en conservant la pertinence de ces descripteurs. C'est pour cette raison que nous avons choisi d'utiliser les 13 ($N_H = 13$) premiers attributs d'Haralick noté $f_1 \ldots f_{13}$ extraits à partir de ces matrices [HSD73].

Nous notons $f_i^{C_k,C_{k'}}$, l'attribut d'indice i extrait de la matrice de co-occurrences chromatique normalisée $m^{C_k,C_{k'}}[\mathbf{I}]$.

Haralick propose quatorze attributs de textures extraits des matrices de co-occurrences

[HSD73] :

1. **Second moment angulaire (ou énergie) :**

$$f_1^{C_k,C_{k'}} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left\{ m^{C_k,C_{k'}}[\mathbf{I}](i,j) \right\}^2 \tag{5.2}$$

2. **Contraste :**

$$f_2^{C_k,C_{k'}} = \sum_{n=0}^{N-1} n^2 \left\{ \sum_{\substack{i=0 \\ |i-j|=n}}^{N-1} \sum_{j=0}^{N-1} m^{C_k,C_{k'}}[\mathbf{I}],(i,j) \right\} \tag{5.3}$$

3. **Corrélation :**

$$f_3^{C_k,C_{k'}} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i-\mu_x)(j-\mu_y) m^{C_k,C_{k'}}[\mathbf{I}](i,j)}{\sigma_x \sigma_y} \tag{5.4}$$

où μ_x, μ_y, σ_x et σ_y sont respectivement les centres de gravité et les écarts type de $m_x^{C_k,C_{k'}}[\mathbf{I}](i)$ et $m_y^{C_k,C_{k'}}[\mathbf{I}](j)$, avec

$$m_x^{C_k,C_{k'}}[\mathbf{I}](i) = \sum_{j=0}^{N-1} m^{C_k,C_{k'}}[\mathbf{I}](i,j)$$

et

$$m_y^{C_k,C_{k'}}[\mathbf{I}](j) = \sum_{i=0}^{N-1} m^{C_k,C_{k'}}[\mathbf{I}](i,j)$$

4. **Variance (ou inertie) :**

$$f_4^{C_k,C_{k'}} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(i-\mu)^2 m^{C_k,C_{k'}}[\mathbf{I}](i,j) \tag{5.5}$$

où μ est le centre de gravité des coefficients de la matrice $m^{C_k,C_{k'}}[\mathbf{I}]$.

5. **Moment différentiel inverse (ou homogénéité) :**

$$f_5^{C_k,C_{k'}} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\frac{1}{1+(i-j)^2}\, m^{C_k,C_{k'}}[\mathbf{I}](i,j) \tag{5.6}$$

6. **Moyenne des sommes :**

$$f_6^{C_k,C_{k'}} = \sum_{l=0}^{2(N-1)} l.m_{x+y}^{C_k,C_{k'}}[\mathbf{I}](l) \tag{5.7}$$

où

$$m_{x+y}^{C_k,C_{k'}}[\mathbf{I}](l) = \sum_{\substack{i=0\\ i+j=l}}^{N-1}\sum_{j=0}^{N-1} m^{C_k,C_{k'}}[\mathbf{I}](i,j),\ \ l=0,1,\ldots,2(N-1)$$

7. **Variance des sommes :**

$$f_7^{C_k,C_{k'}} = \sum_{l=0}^{2(N-1)}(l - f_6^{C_k,C_{k'},v,\theta})^2\, m_{x+y}^{C_k,C_{k'}}[\mathbf{I}](l) \tag{5.8}$$

8. **Entropie des sommes :**

$$f_8^{C_k,C_{k'}} = -\sum_{l=0}^{2(N-1)} m_{x+y}^{C_k,C_{k'}}[\mathbf{I}](l)\ \log\left\{m_{x+y}^{C_k,C_{k'}}[\mathbf{I}](l)\right\} \tag{5.9}$$

9. **Entropie :**

$$f_9^{C_k,C_{k'}} = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} m^{C_k,C_{k'}}[\mathbf{I}](i,j)\ \log\left\{m^{C_k,C_{k'}}[\mathbf{I}](i,j)\right\} \tag{5.10}$$

10. **Variance des différences :**

$$f_{10}^{C_k,C_{k'}} = \sum_{l=0}^{N-1} (l\ -\ \mu_{x-y})^2\ m_{x-y}^{C_k,C_{k'}}[\mathbf{I}](l) \tag{5.11}$$

où

$$m_{x-y}^{C_k,C_{k'}}[\mathbf{I}](l) = \sum_{\substack{i=0\\|i-j|=l}}^{N-1}\sum_{j=0}^{N-1} m^{C_k,C_{k'}}[\mathbf{I}](i,j),\ \ l = 0,1,\ldots,(N-1)$$

et

$$\mu_{x-y} = \sum_{l=0}^{N-1} l.m_{x-y}^{C_k,C_{k'}}[\mathbf{I}](l)$$

11. **Entropie des différences :**

$$f_{11}^{C_k,C_{k'}} = -\sum_{l=0}^{N-1} m_{x-y}^{C_k,C_{k'}}[\mathbf{I}](l)\ \log\left\{m_{x-y}^{C_k,C_{k'}}[\mathbf{I}](l)\right\} \tag{5.12}$$

12. **Information sur la corrélation :**

$$f_{12}^{C_k,C_{k'}} = \frac{f_9^{C_k,C_{k'}} - HXY1}{\max\{HX,HY\}} \tag{5.13}$$

où

$$HXY1 = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} m^{C_k,C_{k'}}[\mathbf{I}](i,j)\log\left\{m_x^{C_k,C_{k'}}[\mathbf{I}](i)\times m_y^{C_k,C_{k'}}[\mathbf{I}](j)\right\}$$

$$HX = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} m_x^{C_k,C_{k'}}[\mathbf{I}](i)\ \log\left\{m_x^{C_k,C_{k'}}[\mathbf{I}](i)\right\}$$

$$HY = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} m_y^{C_k,C_{k'}}[\mathbf{I}](j)\ \log\left\{m_y^{C_k,C_{k'}}[\mathbf{I}](j)\right\}$$

13. **Information sur la corrélation :**

$$f_{13}^{C_k,C_{k'}} = (1 - \exp[-2.0(HXY2 - f_9^{C_k,C_{k'}})])^{\frac{1}{2}} \tag{5.14}$$

où

$$HXY2 = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} m_x^{C_k,C_{k'}}[\mathbf{I}](i)\times m_y^{C_k,C_{k'}}[\mathbf{I}](j)$$
$$\log\left\{m_x^{C_k,C_{k'}}[\mathbf{I}](i)\times m_y^{C_k,C_{k'}}[\mathbf{I}](j)\right\}$$

Ainsi, pour exploiter les propriétés des 3 espaces couleurs retenus ($N_S = 3$), chaque image est codée dans chacun de ces espaces. Ensuite, pour chaque espace couleur, les 6 matrices de co-occurrences sont calculées ($N_M = 6$) et les 13 attributs d'Haralick sont extraits à partir de chaque matrice ($N_H = 13$). Comme nous avons retenu 3 espaces couleur, chaque image texture couleur est caractérisée par $d =$

$N_H \times N_M \times N_S = 13 \times 6 \times 3 = 234$ attributs de textures couleur candidats.

Afin de classifier les différentes images de texture, un classifieur opère généralement dans l'espace d'attributs de départ. Cependant, un nombre élevé d'attributs peut diminuer le taux de bonne classification de ces images et augmenter le temps de calcul [JZ97].

Nous proposons alors de sélectionner, à partir de l'ensemble d'attributs de départ, les attributs les plus pertinents dans un contexte semi-supervisé.

5.4 Résultats expérimentaux

Les bases d'images de texture couleur VisTex, BarkTex et OuTex sont considérées comme des bases de référence car elles sont fréquemment employées dans la littérature afin de comparer expérimentalement les résultats de classification de textures couleur obtenus par différentes méthodes [PGM$^+$], [Lak], [OMP$^+$02]. Avant de présenter les différents résultats , nous détaillons ces différentes bases.

5.4.1 La base 'OuTex'

OuTex contient un grand nombre de textures acquises sous des conditions contrôlées par une camera couleur 3-CCD. Pour construire cette base, 68 images couleurs de texture sont divisées en imagettes de taille 128×128 pixels. Comme les images originales sont de taille 746×538 pixels, cela fait un total de 20 imagettes par classe de texture. La figure 5.3 illustre une image de chaque classe de la base OuTex.

Parmi les 1360 images de la base OuTex, 680 images sont utilisées comme base d'apprentissage et 680 comme base de test [PMV02].

Figure 5.3 : Exemple de textures de la base OuTex : chaque image représente une classe de texture.

5.4.2 La base 'VisTex'

Vistex est une base de données formée d'un ensemble d'images de texture couleur extraites de scènes naturelles [PGM+]. Cette base de référence est constituée de 168 images acquises dans des conditions non contrôlées et réparties en 19 catégories de textures couleur.

Pour construire cette base, 54 images couleur de texture sont divisées en imagettes disjointes de taille 128×128 pixels. Comme la taille des images originales est 512×512 pixels, cela fait un total de 16 imagettes par classe de texture (voir figure 5.4).

Figure 5.4 : Exemple de textures de la base VisTex : chaque image représente une classe de texture.

Parmi les 864 images de la base Vistex, 432 images sont utilisées pour la base d'apprentissage et 432 pour la base de test [PMV02].

5.4.3 La base 'BarkTex'

Les images de la base BarkTex sont divisées également en 6 classes d'écorces de 68 images. La taille de chaque image est 256×384 pixels.

Pour construire la base BarkTex, une région d'intérêt centrée sur l'écorce et de taille 128×128 pixels, est définie en premier. Ensuite, 4 imagette de taille 64×64 pixels sont extraites de chaque région. On obtient alors un ensemble de $68 \times 4 = 272$ imagettes par classe (voir figure 5.5).

Figure 5.5 : Exemple de textures de la base BarkTex : chaque colonne représente une classe de texture.

Parmi les 1632 images de la base BakTex, 816 images sont utilisées comme base d'apprentissage et 816 pour la base de test [Pal04].

5.4.4 Construction des contraintes

Avant de représenter les résultats sur ces différentes bases, notons que les contraintes must-link et cannot-link sont définies entre les différentes images. Ainsi, une contrainte must-link est construite entre deux images de texture de la même classe et une contrainte cannot-link est construite entre deux images de texture de classes dif-

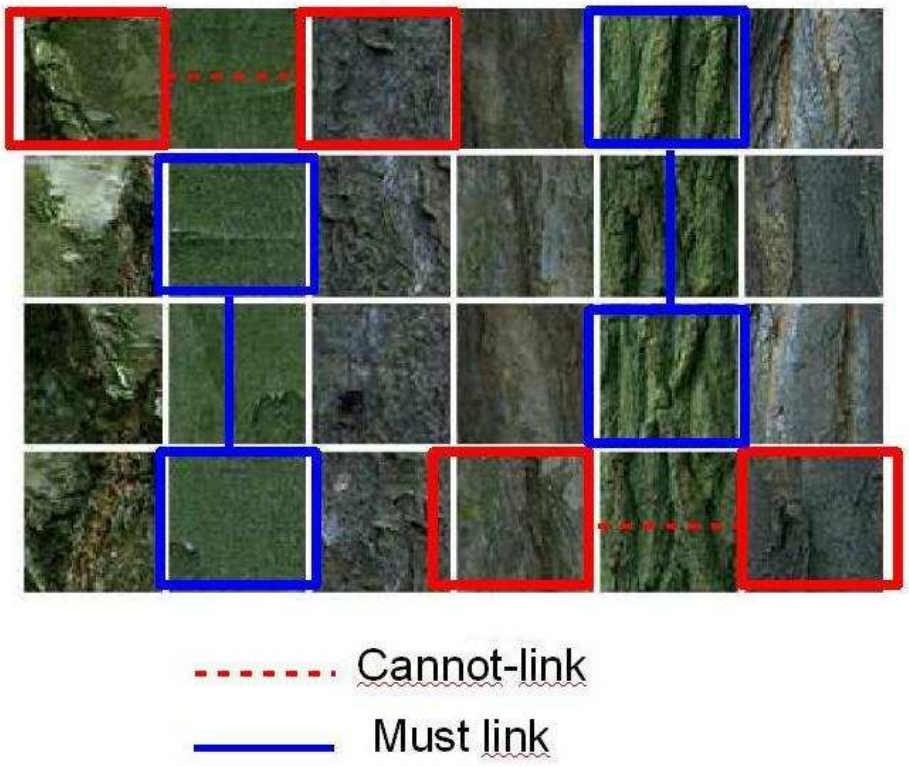


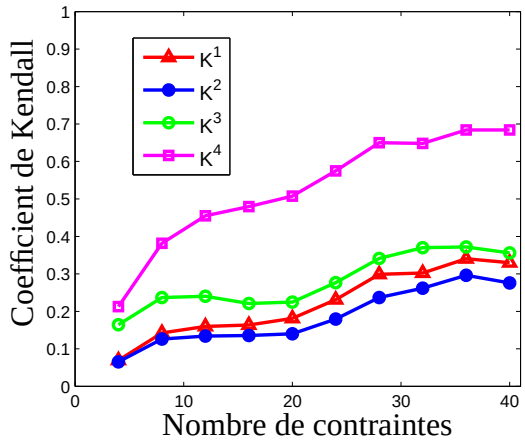
Figure 5.6 : Exemple de contraintes must-link et cannot-link sur la base BarkTex.

férentes. La figure 5.6 illustre ce propos sur la base BarkTex.

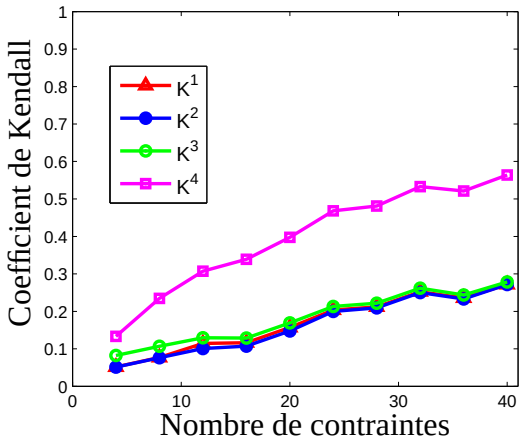
5.4.5 Résultats du coefficient de Kendal

Comme dans le cas des bases de données des chapitres précédents, la procédure de sélection est réalisée sur la base d'apprentissage. Pour les différents scores SC^1, SC^2, SC^3 et SC^4, la procédure de sélection est réalisée sur 100 itérations. A chaque itération, un sous-ensemble différent de contraintes must-link et cannot-link est généré aléatoirement.

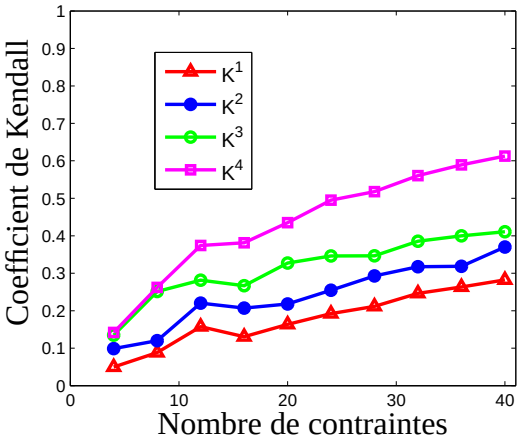
La figure 5.7 illustre les résultats de coefficient de Kendall sur les différentes bases de textures et pour différents cardinaux de S_q allant de 4 contraintes (2 must-link et



(a) 'OuTex'.



(b) 'VisTex'.



(c) 'BarkTex'.

Figure 5.7 : Coefficients de Kendall en fonction du nombre de contraintes $|\mathscr{S}_q|$ sur les 3 bases de texture.

2 cannot-link), jusqu'à 40 contraintes.

Nous pouvons remarquer que les résultats du coefficient de Kendall sur ces bases confirment les résultats obtenus sur les bases précédentes. Les coefficients K^1, K^2, K^3 et K^4 relatives aux scores SC^1, SC^2, SC^3 et SC^4 augmentent en fonction du nombre de contraintes. De même, le coefficient K^4 a toujours la valeur la plus élevée quelque soit le nombre de contraintes sélectionnées. Ceci prouve encore une fois que notre score SC^4 est moins dépendant du sous-ensemble de contraintes que les autres scores de contraintes.

5.4.6 Résultats de classification

Nous comparons aussi les performances des différents scores de sélection d'attributs. Les performances sont comparées en utilisant le classifieur du plus proche voisin qui opère dans l'espace composé des attributs seléctionnés par les scores de contraintes. L'évaluation est menée selon la procédure classique, à savoir que la classification opère dans un contexte supervisé. Pour ce faire, l'algorithme du plus proche voisin utilise alors les labels des données d'apprentissage comme prototypes de classes. Ensuite, nous suivons la démarche expérimentale que nous avons proposé dans le chapitre 4, à savoir que le contexte d'apprentissage est le même pour la sélection et la classification des images test. L'évaluation opère alors dans un contexte semi-supervisé en estimant les labels des données d'apprentissage par la méthode des k-means sous-contraintes, puis ces labels estimés sont utilisés par l'algorithme du plus proche voisin comme prototypes de classes.

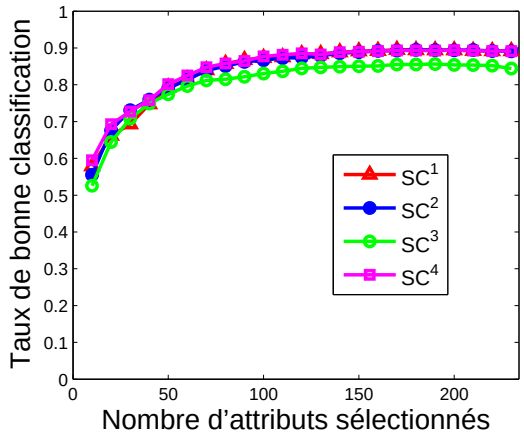
5.4.6.1 Evaluation supervisée

La figure 5.8 montre les résultats de classification des différents scores de sélection. L'évaluation est effectuée dans un contexte supervisé. Les labels des données d'apprentissage sont utilisés afin de classer les données test.

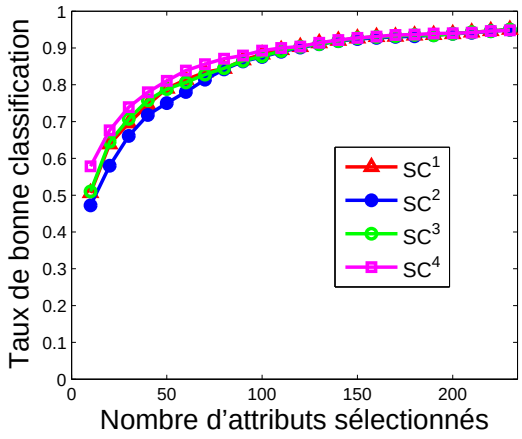
D'après cette figure, nous pouvons voir que les taux de bonne classification des scores SC^1, SC^2, SC^3 et SC^4 sont très proches. Il est doc difficile d'identifier le score qui fournit les meilleurs performances.

Afin de comparer les résultats de classification des différentes scores, nous calculons alors le total de leurs rangs en considérant l'espace composé des 50 premiers attributs sélectionnés.

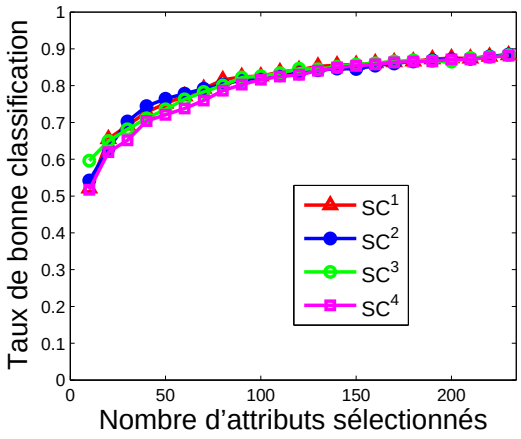
Le tableau 5.1 montre le total des rangs T^* mené sur 100 tirages aléatoires des contraintes pour les différentes bases de texture 'OuTex', 'VisTex' et 'BarkTex'.



(a) 'OuTex'.



(b) 'VisTex'.



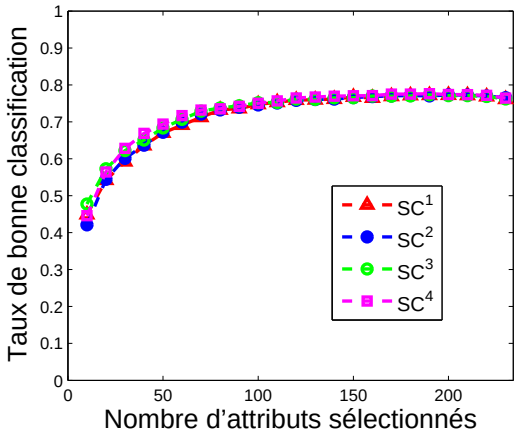
(c) 'BarkTex'.

Figure 5.8 : Taux de bonne classification en fonction du nombre d'attributs sélectionnés sur les 3 bases de textures : 10 contraintes formées de 5 must-link et 5 cannot-link sont utilisées. L'évaluation est effectuée dans un contexte supervisé : l'algorithme du 1-ppv utilise les labels des données d'apprentissage comme prototypes des classes.

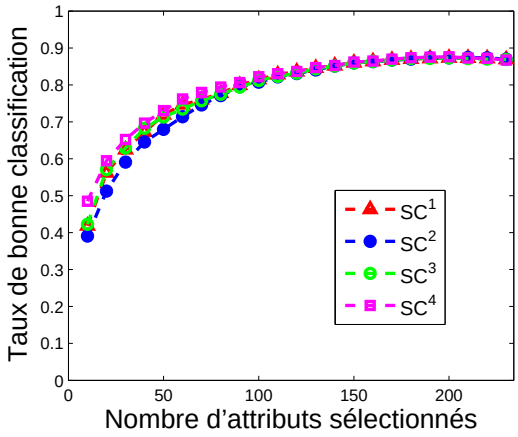
Base de données $\backslash T$	T^1	T^2	T^3	T^4
'OuTex'	228	256	284	**212**
'VisTex'	296	280	244	**200**
'BarkTex'	220	**200**	310	250

Tableau 5.1 : Le total des rangs T^* des différents scores pour différentes bases de données.

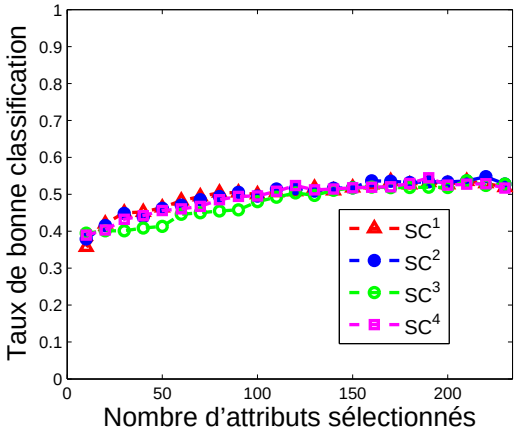
Notre score SC^4 a la valeur la plus faible (signalée en gras) sur les bases 'OuTex' et 'VisTex' et il est classé troisième sur la base 'BarkTex'. Ceci confirme alors que la prise en compte conjointe des contraintes et de la structure locale des données par notre score permet de sélectionner les attributs dont les résultats de classification sont meilleurs que ceux obtenus avec les attributs sélectionnés par les scores de



(a) 'OuTex'.



(b) 'VisTex'.



(c) 'BarkTex'.

Figure 5.9 : Taux de bonne classification en fonction du nombre d'attributs sélectionnés sur les 3 bases de textures : 10 contraintes formées de 5 must-link et 5 cannot-link sont utilisées. L'évaluation est effectuée dans un contexte semi-supervisé : l'algorithme du 1-ppv utilise les labels estimés des données d'apprentissage comme prototypes des classes. Ces labels sont estimés en utilisant k-means sous contraintes

contraintes existants.

5.4.6.2 Contexte d'évaluation cohérent avec celui de la sélection d'attributs

Afin que l'évaluation et la sélection soient cohérentes, nous évaluons les différents scores dans un contexte semi-supervisé. Ainsi, l'algorithmes des k-means sous contraintes opère sur les données d'apprentissage dans l'espace sélectionné par les différents scores afin d'estimer les labels de classes de ces données. Ces labels estimés seront alors utilisés comme prototypes des classes par l'algorithme du plus

proche voisin afin de classifier les données test et les comparer à la vérité du terrain pour évaluer les performances des différents scores.

La figure 5.9 montre les résultats de classification des différents scores de contraintes dans un contexte semi-supervisé. Dans cette figure, nous pouvons voir que les taux de bonne classification des scores SC^1, SC^2, SC^3 et SC^4 sont trés proches et leurs courbes respectives sont confondues.

Base de données $\backslash T$	T^1	T^2	T^3	T^4
'OuTex'	226	219	206	**200**
'VisTex'	226	318	214	**212**
'BarkTex'	**170**	230	330	260

Tableau 5.2 : Le total des rangs T^* des différents scores pour différentes bases de données.

Le tableau 5.2 montre le total des rangs T^* pour les différentes bases de texture 'OuTex', 'VisTex' et 'BarkTex'. L'évaluation dans le contexte semi-supervisé confirment les résultats d'évaluation dans le contexte supervisé. Notre score SC^4 a la valeur la plus faible (indiquée en gras) sur les bases 'OuTex' et 'VisTex' et se classe en troisième position pour la base 'BarkTex'.

Ainsi, les résultats d'évaluation des différents scores dans un contexte supervisé et dans un contexte semi-supervisé prouvent que notre score de sélection permet de sélectionner des attributs de textures dont les résultats de classification sont meilleurs à ceux obtenus en utilisant les attributs sélectionnés par les scores de contraintes existants. En plus, les résultats de coefficient de Kendall ont déjà prouvé que notre score est moins dépendant du jeu de contraintes choisi par l'expert.

5.5 Conclusion

Les attributs d'Haralick extraits des matrices de co-occurrences sont des attributs pertinents qui permettent de discriminer les différentes classes de texture couleur.

Cependant un nombre élevé de ces attributs nécessite une étape de sélection des at-

tributs les plus pertinents. Cela permet alors de réduire le temps de calcul tout en gardant une bonne qualité de classification.

L'utilisateur peut disposer d'une information partielle sous formes de contraintes must-link et cannot-link portant sur quelques images de texture. Les scores de contraintes sont alors utilisés afin de sélectionner les attributs les plus pertinents avant d'effectuer la classification .

Des expériences réalisées sur les bases de texture de référence montrent que notre score de contraintes est moins sensible aux contraintes disponibles que les scores existants. En plus, les attributs sélectionnés par ce score permettent d'obtenir des taux de classification meilleurs à ceux obtenus en utilisant les attributs sélectionnés par les scores existants.

L'utilisation des contraintes must-link et cannot-link pour la sélection d'attributs de textures est une approche originale et prometteuse puisque ces contraintes sont faciles à obtenir. Un expert peut déterminer visuellement si deux images appartiennent à la même classe de textures et ainsi définir une contrainte must-link entre ces deux images ou si deux images n'appartiennent pas à la même classe de textures et ainsi définir une contrainte cannot-link entre ces deux images.

Chapitre 6

Conclusion générale et perspectives

6.1 Conclusion générale

Les travaux présentés dans cette thèse s'inscrivent dans le cadre de la sélection de type "filter" d'attributs à l'aide de scores s'appuyant sur la théorie des graphes. La connaissance a priori apportée par l'utilisateur définit le contexte d'apprentissage dans lequel opère la sélection. Nous nous intéressons au contexte semi-supervisé où la connaissance a priori est formalisée sous forme de contraintes de type must-link et cannot-link entre données. Les différents scores récemment publiés dans la littérature intégrent les contraintes must-link et cannot-link mais négligent l'information fournie par les données non contraintes et pouvant être utilisée dans la sélection. Ceci nous a amené à proposer notre score de sélection semi-supervisée avec contraintes qui intègre à la fois la structure locale des données et le respect des contraintes.

Comme ces scores de contraintes sont généralement sensibles aux contraintes retenues par l'expert, nous avons proposé de mesurer la dépendance des scores de sélection vis-à-vis des contraintes en utilisant le coefficient de Kendall et la matrice des rangs. En effet, le coefficient de Kendall permet de mesurer la dépendance des rangs des attributs par les différents scores par rapport au changement de contraintes

disponibles. Grâce à ce coefficient, nous avons montré à travers des expériences sur différentes bases de données réelles que ce nouveau score est moins sensible à l'ensemble de contraintes que les scores existants. Par ailleurs, il permet de sélectionner les attributs dont les taux de bonne classification sont comparables voire meilleurs à ceux obtenus par les attributs sélectionnés par les scores existants.

La méthode d'évaluation classique des performances des attributs sélectionnés ne prend pas en considération le contexte d'apprentissage dans lequel opère la sélection d'attributs et considère toujours que les données d'apprentissage sont labellisées. Nous avons proposé une méthode d'évaluation non-supervisée et semi-supervisée de façon à ce que la sélection et la classification soient effectuées dans le même contexte d'apprentissage. Il est vrai que les résultats de classification obtenus en utilisant l'évaluation supervisée sont plus élevés que ceux obtenus en utilisant l'évaluation non-supervisée ou semi-supervisée. Cependant, il est plus réaliste de réaliser la sélection et la classification dans le même contexte. Nous avons comparé encore une fois les différents scores en utilisant la méthode d'évaluation proposée. Les résultats obtenus en utilisant la méthode d'évaluation proposée rejoignent ceux atteints par la méthode d'évaluation classique supervisée ; notre score permet d'obtenir des résultats de classification comparables aux résultats fournis par les autres scores.

Dans le cadre de la classification d'images de textures couleur, nous avons appliqué avec succès les divers scores de contraintes afin de sélectionner les attributs de textures les plus pertinents permettant de caractériser ces images. Pour ce faire, des contraintes must-link et cannot-link ont été définies entre les images de textures. Ces contraintes ont été utilisées aussi bien dans la sélection que dans la classification pour évaluer la qualité des résultats obtenus par les différents scores

6.2 Perspectives

Les perspectives à développer à l'issue des travaux de cette thèse sont sur plusieurs niveaux, à savoir la définition de contraintes, la prise en compte des contraintes must-link et cannot-link, mais également sur la pertinence des contraintes définies par l'expert et enfin sur les applications potentielles en analyse d'images.

– Le score proposé dans ce manuscrit est le produit entre le score Laplacien et le score SC^1 de Zhang. Le score Laplacien utilisé dans notre score est le score classique de He et al. [HCN05]. Il pourrait être intéressant d'utiliser les autres formes du score Laplacien introduites par Zhao et al. [ZL07c].

  Notre score traite à égalité la contribution du score Laplacien et du score SC^1. Il serait intéressant de pondérer la contribution des données contraintes et non contraintes afin de favoriser l'une ou l'autre selon la structure des données en présence.

– Il serait intéressant d'étudier la contribution des contraintes must-link et cannot-link dans l'évaluation de la pertinence des attributs. En effet, durant nos travaux nous avons toujours utilisé un nombre égal de contraintes must-link et cannot-link. Zhang et al. considèrent que les contraintes must-link sont plus importantes que les contraintes cannot-link [ZCZ08], tandis que pour Yang et al., ce sont les contraintes cannot-link qui sont les plus importantes [YMSJ10].

– Durant nos expérimentations, les contraintes ont été générées aléatoirement en prélevant au hasard deux données et en mettant entre ces données une contrainte must-link si leurs labels respectifs sont égaux ou une contrainte cannot-link s'ils sont différents. Cette méthode ne permet pas de prendre en considération l'utilité ou l'information apportée par chacune de ces contraintes. Cependant, les différentes contraintes n'ont pas forcément la même importance. En effet, certaines contraintes peuvent être redondantes, incohérentes ou incomplètes par rapport à

l'ensemble des données [DWB06]. C'est pour cette raison qu'il serait important de prendre en considération la cohérence entre les différentes contraintes ainsi que l'information apportée par chacune d'elles dans le processus de sélection d'attributs.

Ce choix est mis en évidence dans la figure 4.3 du chapitre 4. L'augmentation du nombre de contraintes a induit une diminution des performances de classification sur les bases d'expression des gènes. Ce cas révèle la nécessité d'un bon choix des contraintes adaptées à la structure des données à analyser.

– Lors de l'évaluation semi-supervisée proposée dans le chapitre 4, nous avons utilisé l'algorithme des k-means sous contraintes afin de labelliser les données d'apprentissage. Nous avons bien souligné que cet algorithme ne fournit pas toujours des résultats satisfaisants. En effet, il ne garantit pas forcément le respect de toutes les contraintes, surtout s'il existe plusieurs contraintes portant sur la même donnée. Il serait intéressant d'utiliser d'autres algorithmes de classification sous-contraintes plus performants [DWB06].
– Enfin, nous avons appliqué la sélection d'attributs à la classification des images de textures couleur. Pour ce faire, nous avons construit des contraintes must-link et cannot-link entre les images. Une autre problématique intéressante serait la segmentation d'images pour la robotique mobile. Ainsi, l'image sera grossièrement présegmentée en régions par un algorithme tel que la ligne de partage des eaux. Ensuite, des attributs de textures peuvent être extraits pour caractériser chacune de ces régions. L'expert peut construire des contraintes must-link et cannot link entre ces régions. A partir de ces contraintes, nous pouvons sélectionner les attributs les plus pertinents par les divers scores de contraintes et ainsi segmenter les images par classification des régions.

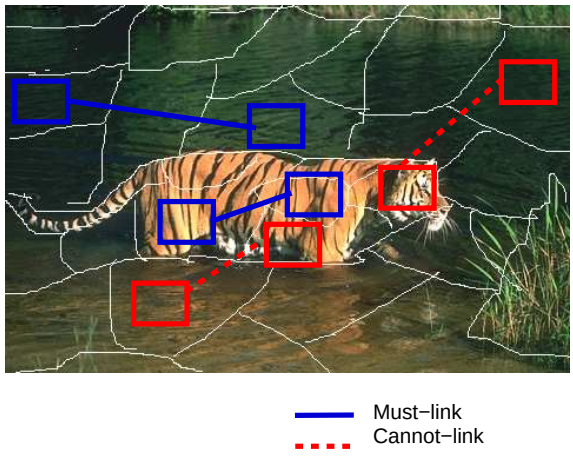


Figure 6.1 : Image présegmentée par l'algorithme de la ligne de partage des eaux. Des contraintes must-link et cannot-link sont construites entre les régions.

A titre d'illustration, la figure 6.1 montre l'image d'un tigre grossièrement présegmentée en régions par la ligne de partage des eaux. Un expert a ensuite construit des contraintes must-link et cannot-link entre ces régions.

Cette approche est importante pour l'application à la segmentation d'images. En effet, la construction des contraintes must-link et cannot-link est une tâche facile que peut réaliser l'expert.

Bibliographie

[ABN+99] U. Alon, N. Barkai, D. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Science of the USA*, 96 :6745–6750, 1999.

[BDK+05] J. Biesiada, W. Duch, A. Kachel, K. Maczka, and S. Palucha. Feature ranking methods based on information entropy with parzen windows. In *Proceedings of the International Conference on Research in Electrotechnology and Applied Informatics 'ICREAI 05'*, pages 109–119, Katowice-Poland, August 2005.

[Big93] N. L. Biggs. *Algebraic graph theory*. Cambridge University Press, 1993.

[Bis96] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, USA, 1996.

[Bis00] G. Bisson. *La similarité : une notion symbolique/ numérique. Apprentissage symbolique-numérique*. Eds Moulet, Brito, Cepadues Edition, 2000.

[BK98] J. Bennett and A. Khotanzad. Multispectral random field models for synthesis and analysis of color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3) :327–332, 1998.

[BKM98] C. Blake, E. Keogh, and C.J. Merz. UCI repository of machine learning databases. http ://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.

[Bla06] A. Blansché. *Classification non supervisée avec pondération d'attributs par des méthodes évolutionnaires*. PhD thesis, Université Louis Pasteur, Strasbourg, 2006.

[BR04] E.L. Van Den Broek and E.M. Van Rikxoort. Evaluation of color representation for texture analysis. In *Proceedings of the Belgium-Dutch Conference on Artificial Intelligence*, pages 35–42, 2004.

[Cas96] P. Casin. L'analyse en composantes principales généralisée. *Revue de statistique appliquée*, 44(3) :63–81, 1996.

[CCM03] D. Cohn, R. Caruana, and A. Mccallum. Semi-supervised clustering with user feedback. Technical report, 2003.

[CT80] G. Carpaneto and P. Toth. Algorithm 548 : solution of the assignment problem. *ACM Transactions on Mathematical Software*, 1980.

[CT91] T. Cover and J. Thomas. *Elements of information*. Wiley-Interscience Edition, 1991.

[Das01] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the 18th International Conference on Machine Learning 'ICML 01'*, pages 74–81, Williamstown, MA, USA, June 2001.

[DB04] J. Dy and C. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5 :845–889, 2004.

[DL97] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1 :131–156, 1997.

[DLM00] M. Dash, H. Liu, and H. Motoda. Consistency based feature selection.

In *Procedings of the 4th International Conference on Knowledge Discovery and Data Mining 'ICKDDM00'*, pages 98–109, 2000.

[DLPT82] E. Diday, J. Lemaire, J. Pouget, and F. Testu. *Elements d'analyse de données*. Dunod, 1982.

[DWB06] I. Davidson, K. L. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases 'PKDD06'*, pages 115–126, Berlin, Germany, September 2006.

[Gol89] D. Goldberg. *Genetic algorithms in search, optimization, and machine learning [*. Addison-Wisley Editions, 1989.

[Grz06] P. Grzegorzewski. The coefficient of concordance for vague data. *Computational Statistics and Data Analysis*, 51 :314–322, 2006.

[GST+99] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and C. D. Bloomfield. Molecular classification of cancer : class discovery and class prediction by gene expression monitoring. *Science*, 286 :531–537, 1999.

[Hal00] M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the 17th International Conference on Machine Learning 'ICML 00'*, pages 359–366, January 2000.

[HCG+05] O.J. Hernandez, J. Cook, M. Griffin, C. De Rama, and M. McGovern. Classification of color textures with random field models and neural networks. *Journal of Computer Science & Technology*, 5(3) :150–157, 2005.

[HCN05] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Proceedings of the Advances in Neural Information Processing Systems 'NIPS 05'*, pages 507–514, Vancouver, Canada, December 2005.

[HK04] O.J. Hernandez and A. Khotanzad. Color image segmentation using multispectral random field texture model and color content features. *Journal of Computer Science & Technology*, 4(3) :141–146, 2004.

[Hol75] J.H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor, the university of Michigan Press, 1975.

[HSD73] R. Haralick, K. Shanmugan, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6) :610–621, 1973.

[Hus10] S. Hussain. *A new co-similarity measure : Application to text mining and bioinformatics*. Phdthesis, Université de Grenoble, September 2010.

[JMGW10] S. Jing, Y. Ming, J. Genlin, and C. Wenbin. A new feature selection algorithm based on mutual information with pairwise constraints. In *Proceedings of the 2nd International Conference on Advanced Computer Control 'ICACC 10'*, volume 3, pages 483–486, 2010.

[JZ97] A. Jain and D. Zongker. Feature selection : Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 :153–158, 1997.

[KBMH11] M. Kalakech, P. Biela, L. Macaire, and D. Hamad. Constraint scores for semi-supervised feature selection : A comparative study. *Pattern Recognition Letters*, 32(5) :656–665, April 2011.

[KH06] A. Khotanzad and O.J. Hernandez. A classification methodology for

color textures using multispectral random field mathematical models. *Mathematical and Computational Applications*, 11(2) :111–120, 2006.

[KKM03] S. Kamvar, D. Klein, and C. Manning. Spectral learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence 'IJCAI 03'*, pages 561–566, Acapulco , Mexico, August 2003.

[KS96] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning ' ICM 96'*, pages 129–134, Bari, Italy, July 1996.

[KS00] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33 :25–41, 2000.

[Lak] R. Lakmann. Base d'images couleur texturées Barktex. Université Koblenz-Landau, ftp ://ftphost.uni-koblenz.de/outgoing/vision/Lakmann/BarkTex.

[Llo82] S. Lloyd. Least squares quantization. *IEEE transactions on information theory*, 28(2) :129–137, 1982.

[LM68] P. Lachenbruch and M. Mickey. Estimation of error rates in discriminant analysis. *Technometrics*, 10(1) :1–11, 1968.

[LM08] H. Liu and H. Motoda. *Computational methods of feature selection*. Chapman and Hall/CRC Data Mining and Knowledge Discovery Series, 2008.

[Lop05] P. Lopez. *Cours de Graphes*, November 2005.

[Los98] R. M. Losee. *Text retrieval and filtering analytic models of performance*. Kluwer Academic Publishers, 1998.

[Lux07] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4) :395–416, 2007.

[Mac67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability PBSMSP67*, pages 281–297, University of California Press, 1967.

[MJ01] M. Meila and J.Shi. A random walks view of spectral segmentation. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics AISTATS 01*, Florida, USA, January 2001.

[MPLH10] Mercado, Pedro, Lukashevich, and Hanna. Feature selection in clustering with constraints : Application to active exploration of music collections. In *Proceedings of the 9th International Conference on Machine Learning and Applications (ICMLA)*, pages 649 –654, Washington, USA, December 2010.

[MRB96] B. Bouchon Meunier, M. Rifqi, and S. Bothorel. Towards general measures of comparison of objects. *Fuzzy Sets and Systems*, 84(2) :143–153, 1996.

[OMP+02] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex new framework for empirical evaluation of texture analysis algorithms. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 1, pages 701–706, 2002.

[Pal04] C. Palm. Color texture classification by integrative co-occurrence matrices. *Pattern Recognition*, 37(5) :965–976, 2004.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6) :559Ű572, 1901.

[PGM+] R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard, and L. Campbell. Base d'images couleur texturées Vistex. Media Laboratory,

Massachusetts Institute of Technology (MIT), Cambridge, http ://vismod.media.mit.edu/pub/VisTex/VisTex.tar.gz.

[PMV02] M. Pietikäinen, T. Mäenpää, and J. Viertola. Color texture classification with color histograms and local binary patterns. In *Proceedings of the 2nd International Workshop on Texture Analysis and Synthesis*, pages 109–112, 2002.

[Por09] A. Porebski. *Sélection d'attributs de texture couleur pour la classification d'images. Application à l'identification de défauts sur les décors verriers imprimés par sérigraphie*. PhD thesis, Université Lille 1, Sciences et Technologies, November 2009.

[PPL97] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4) :380–393, 1997.

[PVM08] A. Porebski, N. Vandenbroucke, and L. Macaire. Haralick feature extraction from LBP images for color texture classification. In *Proceedings of the International Workshop on Image Processing Theory, Tools and Applications (IPTA'08)*, pages 1–8, Sousse, Tunisie, November 2008.

[PVM09] A. Porebski, N. Vandenbroucke, and L. Macaire. Selection of color texture features from reduced size chromatic co-occurrence matrices. In *IEEE International Conference on Signal and Image Processing Applications (ICSIPA'09)*, pages 273 – 278, Kuala Lumpur, Malaysia, November 2009.

[RJ97] R.Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, pages 273–324, 1997.

[Roj96] R. Rojas. *Neural network : a systematic introduction*. Editions Springer-Verlag, 1996.

[RS00] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500) :2323–2326, December 2000.

[Sap06] G. Saporta. *Probabilités, analyses des données et statistiques*. Technip edition, 2006.

[SB88] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5) :513–523, 1988.

[SC88] S. Siegel and N.J. Castellan. *Nonparametric statistics for the behavioral sciences*. McGraw-Hill, New York, USA, 1988.

[Sem04] D. Semani. *Une méthode supervisée de sélection et de discrimination avec rejet. Application au projet Aquathèque*. PhD thesis, Université de La Rochelle, May 2004.

[SH94] F.S. Samaria and A.C. Hartert. Parameterisation of a stochastic model for human face identification. In *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision 'ACV 94'*, pages 138–142, Sarasota, USA, 1994.

[SSM96] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Technical report, Max-Planck-Institut für biologische Kybernetik Arbeitsgruppe Bülthoff, 1996.

[SW49] C. Shannon and W. Weaver. *The mathematical theory of communication"*. University of Illinois Press, 1949.

[SZ10] D. Sun and D. Zhang. Bagging constraint score for feature selection with pairwise constraints. *Pattern Recognition*, 43 :2106–2118, 2010.

[Tal99] L. Talavera. Feature selection as a preprocessing step for hierarchical clustering. In *Proceedings of the 16th International Conference on Machine Learning 'ICML 99'*, pages 433–443, Bled, Slovenia, 1999.

[TFMB04] Alain Trémeau, Christine Fernandez-Maloigne, and Pierre Bonton. *Image numérique couleur - De l'acquisition au traitement*. Dunod, Paris, France, 2004.

[TJ98] M. Tuceryan and A. K. Jain. Texture analysis. In *The Handbook of Pattern Recognition and Computer Vision*, pages 207–248. Editions World Scientific Publishing Co., 1998.

[TV72] G.T. Toussaint, , and T.R. Vilmansen. Comments on feature selection with a linear dependence measure". *IEEE Transactions on Computers*, page 408, 1972.

[Van00] N. Vandenbroucke. *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application à l'analyse d'images de football*. Thèse de doctorat, Université des Sciences et Technologies de Lille, Décembre 2000.

[WC00] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings of the 17th International Conference on Machine Learning 'ICML 00'*, pages 1103–1110, January 2000.

[WCR+01] K. Wagstaff, C. Cardie, S. Rogers, , and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning 'ICML 01'*, pages 577–584, Williamstown, MA, USA, June 2001.

[YL03] L. Yu and H. Liu. Feature selection for high-dimensional data : A fast correlation-based filter solution. In *Proceedings of the 20th Interna-*

tional Conference on Machine Learning 'ICML 03', pages 856–863, Washington, USA, August 2003.

[YL04] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5 :1205–1224, 2004.

[YMSJ10] Yang, Ming, Song, and Jing. A novel hypothesis-margin based approach for feature selection with side pairwise constraints. *Neurocomput.*, 73 :2859–2872, October 2010.

[ZC06] K. Zhang and L.-W. Chan. ICA by PCA approach : relating higher-order statistics to second-order moments. In *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation 'ICA 06'*, volume 3889, pages 311–318, 2006.

[ZCZ08] D. Zhang, S. Chen, and Z.H. Zhou. Constraint score : A new filter method for feature selection with pairwise constraints. *Pattern Recognition*, (41) :1440–1451, October 2008.

[Zen86] S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing (CVGIP)*, 33 :116–125, 1986.

[ZL07a] Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. In *Proceedings of the SIAM International Conference on Data Mining 'ICDM 07'*, pages 641–646, Minneapolis, USA, April 2007.

[ZL07b] Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. Technical report, Computer Science and Engineering (CSE) Department, Arizona State University (ASU), 2007.

[ZL07c] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th International Conference*

on Machine Learning 'ICML 07', pages 1151–1157, Corvalis, USA, August 2007.

[ZLH08] J. Zhao, K. Lu, and X. He. Locality sensitive semi-supervised feature selection. *Neurocomputing*, 71 :1842–1849, 2008.

[ZOD07] Z. Zhu, Y. S. Ong, and M. Dash. Wrapper-filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man and Cybernetics*, 37(1) :70–76, 2007.

Table des figures

Table des tableaux

Résumé :

Dans le cadre de cette thèse, nous nous intéressons à la sélection des attributs en s'appuyant sur la théorie des graphes dans les différents contextes d'apprentissage non supervisé, semi-supervisé et supervisé. En particulier, nous nous intéressons aux scores de classement d'attributs basés sur des contraintes must-link et cannot-link. En effet, ces contraintes sont faciles à obtenir dans le cadre des applications réelles. Elles nécessitent juste de formuler pour deux données si elles se ressemblent et donc doivent être regroupées ensemble ou non, sans requérir d'informations détaillées sur les classes à retrouver.

Les scores de contraintes ont montré de bonnes performances pour la sélection semi-supervisée des attributs. Cependant, ils sont fortement dépendants du sous-ensemble de contraintes disponibles. Nous proposons alors un score qui utilise à la fois l'ensemble des contraintes disponibles et les propriétés locales des données non contraintes.

Des expériences réalisées sur des bases de données artificielles et réelles montrent que ce nouveau score est moins dépendant de l'ensemble de contraintes disponibles que les scores existants tout en atteignant des performances de classification similaires.

La sélection semi-supervisée d'attributs a également été appliquée avec succès à la classification de textures couleur. En effet, parmi les nombreux attributs de texture pouvant être extraits des images couleur, il est nécessaire de sélectionner les plus pertinents afin d'améliorer la qualité de classification.

Mots-clés : sélection d'attributs, apprentissage semi-supervisé, scores de contraintes, textures couleur, analyse de données.

Abstract :

Within the framework of this thesis , we are interested in feature selection methods based on graph theory in different unsupervised, semi-supervised and supervised learning contexts. We are particullary interested in the feature rankig scores based on must-link et cannot-link constraints. Indeed, these constraints are easy to be obtained on real applications. They just require to formulize for two data samples if they are similar and then must be grouped together or not, without detailed information on the classes to be found.

Constraint scores have shown good performances for semi-supervised feature selection. However, these scores srongly depend on the given must-link and cannot-link subsets built by the user. We propose then a new semi-supervised constraint score that uses both pairwize constraints and local properties of the unconstrained data.

Experiments on artificial and real databases show that this new score is less sensitive to the given constraints than the previous scores while providing similar performances.

Semi supervised feature selection was also successfully applied to the color texture classification. Indeed, among many texture features which can be extracted from the color images, it is necessary to select the most relevant ones to improve the quality of classification.

Key words : feature selection , semi-supervised learning, constraint scores, color texture, data analysis.