Catégorisation par mesures de dissimilitude et caractérisation d'images en multi échelle

Agata Manolova

THÈSE EN CO TUTELLE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE ET L'UNIVERSITÉ TECHNIQUE DE SOFIA, BULGARIE

Spécialité : **Signal Image Parole Télécoms**

Arrêté ministériel : 7 août 2006

Présentée et soutenue par

Agata Hristova MANOLOVA

Co-directeur de thèse : Anne Guérin-Dugué
Co-directeur de thèse : Roumen K. Kountchev

préparée au sein du **Laboratoire GISPA-Lab**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du signal**

Catégorisation par mesures de dissimilitude et caractérisation d'images en multi échelle

Thèse soutenue publiquement le **11.10.2011**,
devant le jury composé de :

Monsieur Jocelyn CHANUSSOT,	**Président**
Monsieur Michel VERLEYSEN,	**Rapporteur**
Monsieur Alain RAKOTOMAMONJY,	**Rapporteur**
Monsieur Patrick LAMBERT,	**Examinateur**
Monsieur Gilles CELEUX,	**Examinateur**
Monsieur Roumen K. KOUNTCHEV,	**Co-directeur de thèse**
Madame Anne GUERIN-DUGUE,	**Co-directeur de thèse**

Acknowledgements

This dissertation is a result from my work at the Technical University of Sofia, Bulgaria and the University of Grenoble, France in the GIPSA laboratory, thanks to the scholarship of AUF (Agence Universitaire de la Francophonie).

I am most grateful to my advisors Prof. Dr. Anne Guerin-Dugue and Prof. Dr. Roumen Kountchev who have always supported my research and gave me many good advises. Thank you for your patience and help. You gave the taste of the research and now I hope I will be able to follow in your footsteps and achieve if possible as much as you have.

My big gratitude goes to Elżbieta Pękalska and Robert Duin for their help and encouragement for my work. Elżbieta Pękalska was always available for questions and without her help this thesis could not be what it is.

I am most grateful to Ladan Amini for her support and care. She took care of me when even I forgot about it. She gave me courage when I had none and always believed in me. So thank you a thousand times.

I would like to express my thanks to Peter Vassilev his invaluable help with C programming and for his patience. At the beginning I was ignorant but he showed me that there is nothing that I could not do and finally I was not so bad at programming. I even began to enjoy it.

In addition, I would like to thank all my colleagues from the two laboratories for fruitful discussions and support and for all the fun during the conferences.

I am greatly indebted to Maxim Vassilev for his great contribution to this work. Without him this thesis could not possible. Thank you for trying to do everything so I don't have to do it.

Finally all the gratitude goes to my family for their patience and support for all these years for always believing in me and my capabilities.

Abstract

The dissimilarity representation is an alternative for the use of features in the recognition of real world objects like images, spectra and time-signal. Instead of an absolute characterization of objects by a set of features, the expert or the system is asked to define a measure that estimates the dissimilarity between pairs of objects. Such a measure may also be defined for structural representations such as strings and graphs. The dissimilarity representation is potentially able to bridge structural and statistical pattern recognition.

In this thesis we introduce a new fast Mahalanobis-like metric the "Shape Coefficient" for classification of dissimilarity data. Our approach is inspired by the Geometrical Discriminant Analysis and we have defined decision rules to mimic the behavior of the linear and quadratic classifier. The number of parameters is limited (two per class). We also expand and ameliorate this advantageous and rapid adaptive approach to learn only from dissimilarity representations by using the effectiveness of the Support Vector Machines classifier for real-world classification tasks.

Several methods for incorporating dissimilarity representations are presented, investigated and compared to the "Shape Coefficient" in this thesis:

- Pekalska and Duin prototype dissimilarity based classifiers;
- Haasdonk's kernel based SVM classifier;
- KNN classifier.

Numerical experiments on artificial and real data show interesting behavior compared to Support Vector Machines and to KNN classifier: (a) lower or equivalent error rate, (b) equivalent CPU time, (c) more robustness with sparse dissimilarity data.

The experimental results on real world dissimilarity databases show that the "Shape Coefficient" can be an alternative approach to these known methods and can be as effective as them in terms of accuracy for classification.

Keywords: Discriminant Analysis, Dissimilarity matrix, Support Vector Machines, Classification.

Tables de Matières:

Introduction

Motivation

La reconnaissance des formes repose traditionnellement sur une représentation de caractéristiques. Les caractéristiques doivent de préférence être définies sur la base de connaissances d'experts du domaine d'application. Chaque objet est représenté donc par un vecteur multidimensionnel dans un espace de caractéristiques. La dimension de cet espace est égale au nombre de caractéristiques définies. Cette approche est très efficace, et d'autant plus si des connaissances suffisantes sont disponibles pour sélectionner un ensemble restreint et suffisant de caractéristiques pour bien distinguer des objets. Sinon dans la grande majorité des cas, on utilise un nombre plus important de caractéristiques et la sélection pourrait s'effectuer a posteriori. Ainsi, la dimensionnalité de l'espace des caractéristiques peut être importante.

Depuis quelques années, Pekalska, Duin et leurs collègues ont travaillé sur le sujet de la représentation alternative des observations, par des dissimilitudes pour la reconnaissance des formes [Pekalska&Duin 2005 ; Pekalska&Duin 2001 ; Haasdonk 2005]. Selon eux, l'utilisation de représentations par dissimilitudes vient d'une motivation au centre de la notion de dissimilitude. En effet, si l'on suppose que les objets dits «similaires» peuvent être regroupés pour former une classe, une «classe» n'est rien de plus qu'un ensemble de ces objets « similaires ». S'appuyant sur cette idée, Duin et ses collègues affirment que la notion de proximité (similitude ou de dissimilitude) est en fait plus fondamentale que celle de caractéristique pour définir une classe [Duin et al. 2004].

Les travaux de Pekalska et de Duin ont donné un essor important à l'approche de classement basé sur de matrice de dissimilitudes [Pekalska&Duin 2005 ; Pekalska&Duin 2006].

Ainsi, les dissimilitudes sont un moyen de définir des classifieurs entre des classes, qui ne sont pas basés sur des caractéristiques des profils individuels, mais plutôt sur une mesure de dissimilitude adéquate directement entre ces profils ou entre leurs caractéristiques. De plus, les problèmes d'apprentissage dans des espaces à grandes dimensions rendant peu robuste l'estimation des paramètres des classifieurs, sont ainsi contournés.

Objectifs de la thèse

Le point de départ pour cette thèse était le travail de Guérin-Celeux [Guérin&Celeux 2001] qui pose les fondations de l'approche « Coefficient de forme » (« *Cs* »), basée sur des matrices de dissimilitudes. Un premier stage de fin d'études dans le laboratoire CLIPS-IMAG a permis de progresser sur le développement de cet indice et de se poser les premières questions sur le problème d'optimisation des coefficients d'ajustement. Dès le début, on a reconnu que la méthode proposée, basée sur les matrices de dissimilitudes, est limitée dans sa partie d'optimisation des paramètres par classe. Après une étude extensive, on s'est orienté vers la méthode de classement par machines à vecteurs de support (SVM) comme méthode de classement non-paramétrique, se rapprochant ainsi des travaux de [Pekalska&Duin 2005] et [Haasdonk 2005] qui ont montré que même si beaucoup de mesures de dissimilitude sont non métriques, on peut trouver des solutions adaptées.

Au départ de la thèse, nous avions l'ambition d'utiliser comme contexte applicatif pour la classification par dissimilitudes, la recherche d'images par le contenu ; et cela en utilisant la plateforme de recherche d'images « Scopie » qui avait été développée au laboratoire CLIPS-IMAG. Malheureusement des problèmes de mise à jour informatiques des bases de données ne nous ont pas permis de mener à terme ce projet. Aussi la partie d'étude sur la représentation des images par le contenu s'est réalisée à Sofia sans malheureusement de réelles interactions avec le développement des algorithmes de classement par dissimilitudes effectué à Grenoble.

Ainsi les buts de cette thèse sont les suivants et concernent d'une part le classement par dissimilitudes et d'autre la représentation d'images par le contenu :

1. Faire une étude détaillée sur l'indice « Coefficient de forme », introduit par [Guérin&Celeux 2001] et sur le comportement de ces paramètres d'ajustement vis-à-vis de différentes statistiques des classes.
2. Proposer des algorithmes appropriés d'optimisation à deux ou plusieurs classes, de l'indice « Coefficient de forme » et étudier son comportement vis-à-vis des différentes bases de données artificielles et réelles.
3. Faire une analyse comparative entre les différentes méthodologies existantes dans la littérature et le « Coefficient de forme » sous ses différentes formes.
4. Proposer un algorithme de description des images en multi échelle, basé sur la « Pyramide Réduite Différentielle ».
5. Elaborer un système de classement et reconnaissance des images, basé sur la description proposée si dessus et l'indice « Coefficient de forme ».

Plan de thèse

Ce document est composé de quatre chapitres.

Le premier chapitre représente une brève description des différentes dissimilitudes, utilisées dans cette thèse avec leurs principales propriétés ainsi que les propriétés des matrices de dissimilitudes. Un état de l'art détaillé des méthodes de classement basées sur des matrices de dissimilitudes sera présenté. Le chapitre fini par une synthèse des approches possibles.

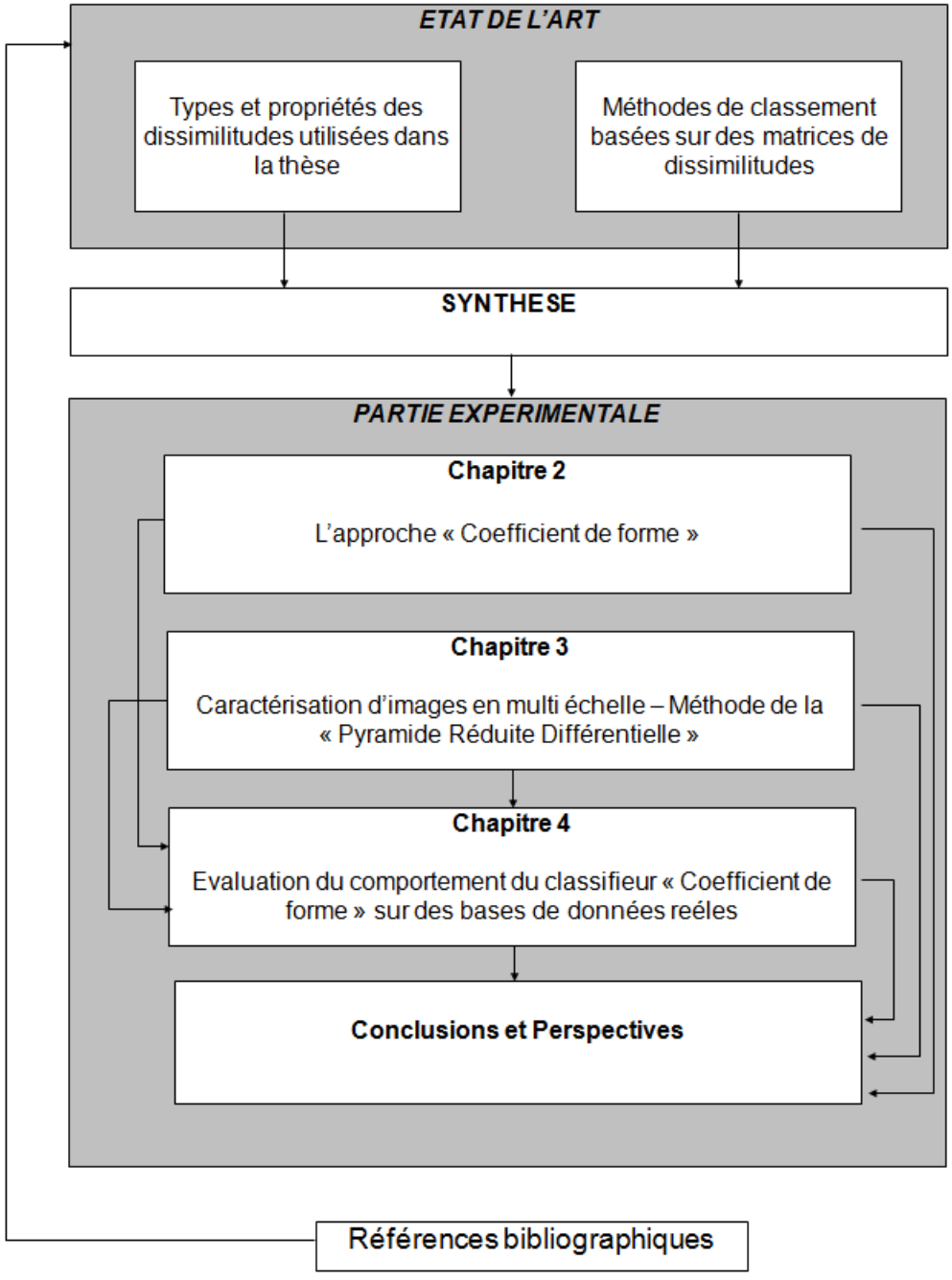
Le second chapitre débute par une présentation de l'analyse discriminante géométrique et l'introduction de l'indice « Coefficient de forme » avec ses paramètres associés de configuration. Une première approche d'optimisation de ces paramètres est présentée, utilisant des tirages « Monté Carlo » (approche « *Cs* géométrique »). Ensuite, à l'aide d'un recodage des données, on propose une optimisation de même nature que les classifieurs SVM (approches « *Cs*-SVM » et « *Cs*-LSSVM »). Le chapitre finit par une synthèse des résultats obtenus sur des données artificielles – des distributions gaussiennes bidimensionnelles. Egalement, le comportement de l'indice « Coefficient de forme » vis-à-vis des matrices creuses est présenté.

Le troisième chapitre introduit la méthode de représentation des images en multi échelle en utilisant la « Pyramide Réduite Différentielle ». Ce chapitre finit par la proposition d'un algorithme de description de visages dont le test en classement sera présenté au chapitre suivant.

Le dernier chapitre est consacré à la partie expérimentale de cette thèse. Des résultats de l'implantation de l'indice proposé (« *Cs* ») avec des bases de données issues des applications du monde réel seront présentés et comparés avec les méthodes de classement, décrites dans le premier chapitre.

Le document se termine par une conclusion, synthétisant les différents résultats obtenus. Les perspectives ouvertes par ce travail seront également présentées.

Structure de la thèse



Chapitre 1. Représentation et Classement par dissimilitudes

Ce chapitre présente les principales notions théoriques sur lesquelles nous nous sommes appuyées pour développer cette thèse. On présente certaines mesures de dissimilitudes qui ont servies dans cette thèse, puis les classifieurs les utilisant et qui ont été développés dans ces dix dernières années. Ainsi on décrit le classifieur par prototypes d'Elzbieta Pekalska et les classifieurs à vecteurs de support et leurs applications aux dissimilitudes, tels proposés par Bernard Haasdonk. Egalement, on évoque la méthode du positionnement multidimensionnel pour la transformation de l'espace des dissimilitudes vers un espace vectoriel des caractéristiques, et le classifieur aux plus proches voisins, comme étant la méthode la plus simple et la plus directe dans ce contexte. Enfin, pour pouvoir situer nos travaux dans ce panorama, on présente brièvement le classifieur « Coefficient de forme » qui est l'objet principal d'étude de cette thèse.

1.1. Représentation des données

Les systèmes automatiques de reconnaissance des objets comme des images, des vidéos, des signaux, des spectres, etc., sont conçus par apprentissage à partir d'exemples des objets étiquetés suivant leur classe d'appartenance. Ce système de reconnaissance se réalise en suivant quatre étapes principales [Duin&Pekalska 2007] :

Représentation : Les objets individuels doivent être décrits de telle sorte que leurs caractéristiques soient extraites et encodées d'une façon numérique ou symbolique. Cette représentation est en fait une simplification, mais elle doit « capter » les caractéristiques les plus importantes, les plus discriminantes des objets d'une classe à une autre. Les représentations les plus courantes utilisent des espaces vectoriels euclidiens, des chaînes de caractères, des graphes etc. Une bonne représentation doit être compacte. C'est-à-dire que dans l'espace de représentation, les points associés à des objets appartenant à la même classe doivent former des partitions compactes. C'est l'hypothèse de compacité qui stipule que les objets qui se ressemblent dans le monde physique, doivent aussi se ressembler dans leur espace de représentation [Duin&Pekalska 2007]. Les principaux types de représentation sont résumés dans le paragraphe ci-dessous.

Prétraitement : Il s'agit d'une étape optionnelle de prétraitement des données. Le plus souvent une étape de normalisation ou de réduction de dimension.

Apprentissage : À partir d'un choix de représentation, il s'agit d'apprendre la relation entre les objets à travers leur représentation et leur classe d'appartenance.

Généralisation : L'apprentissage sur les objets déjà étiquetés doit permettre le classement de nouveaux objets non étiquetés. C'est la généralisation de la règle de décision pour des nouvelles observations.

Les thèmes de l'apprentissage et la généralisation ont été intensément étudiés suivant différentes approches telles que la théorie statistique de l'apprentissage [Vapnik 1998 ; Jain et al. 2000], la reconnaissance des formes [Fukunaga 2001 ; Duda et al. 2000], les réseaux de neurones [Schalkoff 1997] et les machines à vecteurs de support [Cristianini&Shawe-Taylor 2000 ; Bishop 2006 ; Alpaydin 2004].

Les principaux types de représentation des objets sont les suivants :

- *Représentation par caractéristiques* : Les caractéristiques décrivant les objets doivent permettre de les différencier en classes. Ces caractéristiques sont définies par des experts du domaine d'application afin de mieux exploiter leurs connaissances de l'application. Un inconvénient de l'utilisation des caractéristiques est que des objets différents peuvent avoir la même représentation s'ils diffèrent principalement par des propriétés qui ne sont pas exprimées dans l'ensemble des caractéristiques choisies. Cela se traduit par une erreur de classification intrinsèque qui pourrait être corrigée par l'ajout de nouvelles caractéristiques.
- *Représentation par données brutes* : En prenant l'exemple d'un signal ou d'une image, il s'agit directement des valeurs d'acquisition, disposées en vecteur. L'utilisation directe des données brutes se fait souvent en l'absence de bonnes caractéristiques. Cette représentation est peu efficace car elle aboutit à une représentation vectorielle des objets, complètement éclatée, sans aucune invariance et dans un espace à très grandes dimensions.
- *Représentation par modèles de probabilité* : Les caractéristiques peuvent être reliées à un modèle de probabilité donné. Ces modèles peuvent être fondés sur des connaissances d'experts ou formés à partir des exemples.
- *Représentation par modèles structurels* : Au lieu d'utiliser des probabilités, des modèles d'objets peuvent être basés sur une description structurelle telle que des chaînes, des contours, des séquences de temps, des arbres et autres données dépendantes d'un ordre particulier. Des procédures automatiques pour la construction des descriptions structurelles sont encore à développer mais une première idée peut être trouvée dans [Goldfarb et al. 2004].
- *Représentation par des dissimilitudes* : Une alternative à l'utilisation des caractéristiques et des données brutes est la représentation par dissimilitudes, basée sur des comparaisons directes des objets par paires [Van Cutsen 1994]. C'est ce mode de représentation qui sera utilisé pour cette thèse.

1.2. Représentation par dissimilitudes

1.2.1. Principes de base

L'utilisation des dissimilitudes pour représenter les objets ouvre de nouvelles possibilités en apprentissage statistique [Duin et al. 2004 ; Jacobs et al. 2000 ; Devroye et al. 1996] car les dissimilitudes peuvent capturer à la fois les informations statistiques et structurelles des objets [Bunke et al. 2001]. Des techniques appropriées de classement sont alors développées [Duin et al. 1998 ; Paclik&Duin 2003 ; Pekalska et al. 2001 ; Haasdonk 2005 ; Duch 2004]. Les

dissimilitudes peuvent être construites directement à partir des données brutes (ex. dissimilitudes entre des images) ou à partir de caractéristiques extraites.

Dans la suite de ce chapitre, nous allons présenter les principales mesures de dissimilitude incluant celles qui seront utilisées dans les expérimentations sur les bases de données réelles, présentées dans le chapitre 4.

On distingue les caractéristiques suivant si elles sont binaires, catégorielles, ordinales, symboliques ou quantitatives, [Pekalska&Duin 2005]. En tenant compte de ces propriétés, on choisira la mesure de dissimilitude appropriée pour chaque type de données.

Soient $\mathcal{L}= \{l_1, l_2, \ldots, l_m\}$, un ensemble de caractéristiques et D_l un ensemble de valeurs valides pour une caractéristique $l \in \mathcal{L}$:

l est binaire : si D_l est un ensemble de deux symboles ou deux nombres comme 0 ou 1. Ces deux symboles représentent soit la présence (1) soit l'absence (0) d'une caractéristique particulière ou deux qualités opposées, par exemple petit (0) ou grand (1). Chaque objet est représenté par un vecteur binaire. Plusieurs mesures de dissimilitude pour ce type de données existent. Parmi elles, on distingue la distance de Jaccard, de Yule, Pearson, binaire euclidienne, Hamming etc.

l est catégorielle : si D_l est un ensemble de nombres finis et discrets, par exemple de 1 à 4 pour désigner la couleur des cheveux. Dans certains cas, les mesures de distance pour les données binaires peuvent être utilisées comme mesure de dissimilitude.

l est ordinale : si D_l est un ensemble fini, discret et ordonné. Un exemple sont les grades académiques américaines (A, B, C, D, F) ou le degré de préférence (le dégoût, neutre, OK, aime, formidable). Dans le cas des variables ordinales, la mesure de distance doit prendre en compte la position de chaque catégorie dans l'ordre et la distance doit être plus grande pour les catégories les plus éloignées. Des mesures spécifiques peuvent être utilisées comme la dissimilitude de Jaccard. Dans l'autre cas, on peut utiliser les dissimilitudes binaires en appliquant un recodage des vecteurs originaux en vecteurs binaires.

l est symbolique : si D_l est un ensemble fini et discret de symboles, par exemple la nationalité. Les caractéristiques symboliques représentent un ensemble de valeurs ou symboles ou modalités possibles. Leurs valeurs peuvent être comptées mais pas ordonnées. Les dissimilitudes entre les données symboliques doivent être définies en respectant l'ordre, le contenu et la portée de chaque composante (le tout normalisé entre [0, 1]) du vecteur de caractéristiques. On peut trouver dans [Pekalska&Duin 2005] plus de détails sur ce type de dissimilitudes.

l est quantitative : si l est mesuré dans un intervalle et D_l un sous-ensemble convexe de $\Re$, par exemple hauteur, longueur, température etc. Les mesures de ce type de données ou attributs sont représentées par des nombres réels.

La similitude peut être définie comme une relation entre deux objets de même nature. Plus la similitude est grande, plus les objets se ressemblent, au contraire de la dissimilitude. La similitude et la dissimilitude expriment la ressemblance entre les objets, mais leur accent est différent. Utiliser l'une ou l'autre dépend du type de données et du problème à résoudre. La différence n'est pas essentielle, puisque les mêmes raisonnements et méthodologies peuvent être

appliqués à des représentations de similitude comme à des représentations de dissimilitude, en utilisant des relations de passage, comme par exemple :

$$d(i,j) = \frac{1 - s(i,j)}{s(i,j)} \text{ ou } d(i,j) = c - s(i,j), \tag{1.1}$$

où *c* est une constante, *s(i,j)* la mesure de similitude entre deux objets *i* et *j* d'un ensemble *E* et *d(i, j)*, la mesure de dissimilitude [Pekalska&Duin 2005 ; Keshavarzi et al. 2009].

La mesure de dissimilitude a un pouvoir discriminant pour distinguer des objets appartenant à des classes séparables et elle supporte l'hypothèse de compacité, signifiant que des faibles variations d'un objet se traduiront par des faibles variations de ces mesures de dissimilitude.

Dans cette thèse, nous avons utilisé des mesures de proximité, codées par les dissimilitudes estimées à partir de données quantitatives. Ces dernières peuvent être soit les données brutes, soit des caractéristiques extraites des données. D'une manière ou d'une autre, un objet est alors caractérisé par un vecteur de dissimilitudes.

1.2.2. Définitions

Pour obtenir une description robuste des objets, une mesure de dissimilitude doit intégrer les invariances nécessaires pour un meilleur regroupement intra-classe et éloignement inter-classe. Pour ce faire, il est nécessaire que la dissimilitude intègre des invariances propres au problème posé. Par exemple, pour la reconnaissance d'objets dans les images, la dissimilitude doit être invariante à la translation, à la rotation, à l'échelle spatiale, aux variations d'illumination et robuste au bruit et aux occlusions. La mesure de dissimilitude est spécifiée par l'utilisateur expert du domaine, de manière à intégrer ses connaissances de l'application.

Dans la suite, nous utilisons les notations adoptées dans l'Encyclopédie de distances de [Deza&Deza 2009].

Notons *E* l'ensemble des objets à classer. Une dissimilitude est une application de $E \times E$ dans $\Re^+$ ayant les propriétés telles que :

- Non négativité – $d(i,j) \geq 0, \forall i, j$;
- Réflectivité - $d(i,i) = 0, \forall i$;
- Symétrie - $d(i,j) = d(j,i), \forall i, j$.

Si en plus des trois propriétés énumérées ci-dessus, la mesure de dissimilitude satisfait aussi l'inégalité triangulaire :

- $d(k,i) + d(k,j) \geq d(i,j), \forall i, j, k$,

alors la mesure de dissimilitude est métrique et on peut parler de distance. La dissimilitude ayant les propriétés de réflectivité et de symétrie et obéissant à l'inégalité triangulaire est une distance pseudo-métrique et si elle a les propriétés de non négativité, symétrie et réflectivité mais elle n'obéit pas à l'inégalité triangulaire alors elle est semi-métrique.

On peut rajouter une cinquième propriété, l'ultra métricité, pour obtenir une distance ultra-métrique :

- $d(k,i) \leq \max\{d(k,j), d(i,j)\}, \forall i, j, k$.

Cette version renforcée de l'inégalité triangulaire implique que les deux plus grandes distances entre trois points sont égales. De cette façon, tout triangle formé à partir de trois points est un triangle isocèle (voir exemple dans la section 1.4.1). Cette caractéristique est particulièrement utilisée dans le cas des méthodes hiérarchiques de classification.

Dans cette étude, on s'intéressera aux dissimilitudes métriques et non métriques.

1.2.3. Exemples de mesures de dissimilitude

L'objectif de ce paragraphe est de présenter les principales mesures de dissimilitude qui seront utilisées pour les bases de données réelles étudiées dans le chapitre 4.

Notons E l'ensemble des objets à classer. L'objet i est représenté par son vecteur de représentation $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ et l'objet j représenté par son vecteur de représentation $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$, $i, j \in E$ et $\mathbf{x}, \mathbf{y} \in \Re^m$.

Distance Euclidienne

$$D_E(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}. \tag{1.2}$$

La distance euclidienne est la mesure de dissimilitude la plus utilisée et la plus connue. Le lieu des points à distance égale d'un point d'origine est une hyper sphère centrée sur le point d'origine (voir fig.1.1). Cette distance est très sensible des petites déformations/changements des variables c'est-à-dire elle donne plus de poids à des variations importantes [Webb 2002]. Elle respecte l'inégalité triangulaire ce qui garantie l'application de l'hypothèse de compacité. Cette mesure de dissimilitude est invariante par rapport aux translations et aux rotations des données dans $\Re^m$. Appliquée directement sur les pixels de deux images, c'est une mauvaise candidate pour estimer la dissimilitude entre deux images car elle ne tient pas compte des relations spatiales entre les pixels [Wang et al. 2005].

Distance de Mahalanobis

$$D_M(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \Sigma^{-1}(\mathbf{x}-\mathbf{y})}. \tag{1.3}$$

La matrice Σ est la matrice de variance - covariance des données. Cette matrice représente une dépendance statistique du second ordre des *m* variables quantitatives. Elle diffère de la distance Euclidienne par le fait qu'elle prend en compte la corrélation de données. La distance de Mahalanobis accorde un poids moins important aux composantes les plus bruitées (variances plus grandes). Cette distance est invariante aux transformations affines des objets. Cette mesure de dissimilitude est très utile si on veut prendre en compte la dépendance statistique des données [Taguchi&Jugulum 2002]. Elle est largement utilisée dans le domaine de classification et de classement. La fig.1.1 illustre une comparaison entre la distance de Mahalanobis et la distance euclidienne.

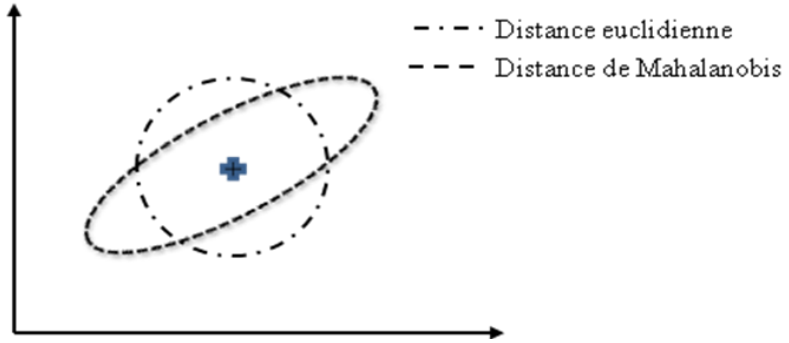


Fig.1.1 Comparaison entre la distance euclidienne et distance de Mahalanobis.

La distance de Minkowski

$$D_P(\mathbf{x},\mathbf{y}) = \left[\sum_{i=1}^{m} |x_i - y_i|^p\right]^{1/p}. \tag{1.4}$$

Pour $0 < p < 1$ (Fig.1.2) cette distance est appelée distance D_p fractionnaire. Elle est plutôt utilisée dans les cas où il y a peu d'observations avec un grand nombre de caractéristiques. Cette distance semi-métrique minimise l'impact des grandes différences entre les valeurs des caractéristiques des données à comparer. Cette propriété est importante pour la comparaison de données qui souffrent de la « malédiction de dimensionnalité ». En effet, quand le nombre de caractéristiques est grand, la probabilité que l'une au moins d'elles ait une valeur extrême peut être élevée. Dans le cadre de la métrique euclidienne, le point correspondant se retrouve aux frontières de l'espace occupé par les données, et sa distance aux autres données est grande. Cette distance fractionnaire minimise le poids des $|x_i - y_i|$ ayant des grandes contributions. Des points ayant la plus part des valeurs des caractéristiques proches seront donc proches dans cet espace.

Pour $p = 1$, on obtient la distance de Manhattan (aussi appelée distance « city-block » ou métrique absolue) :

$$D_1(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} |x_i - y_i|. \tag{1.5}$$

Cette métrique est un peu plus économique en temps de calcul que la distance Euclidienne et peut être utilisée si des contraintes de temps l'imposent.

Pour $p = 2$ on obtient la distance Euclidienne (Fig. 1.2).

Et pour p tendant vers l'infini, on obtient la métrique du « maximum » ou la distance de l'échiquier :

$$D_{\max}(\mathbf{x},\mathbf{y}) = \max_i |x_i - y_i| . \tag{1.6}$$

La distance de Minkowski permet de jouer indépendamment sur les deux puissances présentes dans l'équation (1.4) – m et p, pour trouver l'équilibre voulu entre l'importance du nombre de composantes $|x_i - y_i|$ différentes et l'importance de la différence elle-même. Le choix d'une valeur appropriée pour p dépend du poids qu'on souhaite donner aux plus grandes différences: plus grande la valeur de p accordée plus importante la différence.

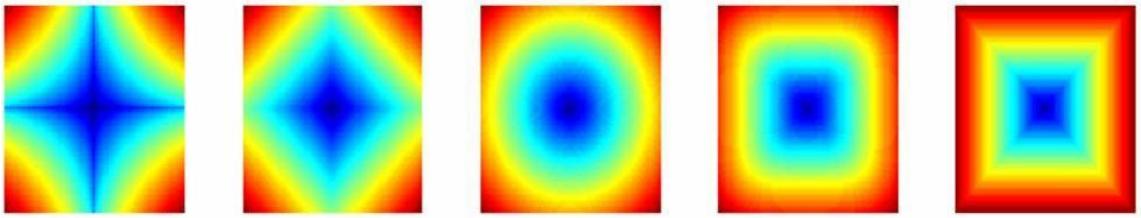


Fig. 1.2 Comportement des distances de Minkowski pour (de gauche à droite) p = 0.5, p = 1 (Manhattan), p = 2 (Euclidien), p = 4 et p = ∞ (Echiquier). La distance au centre est codée par la couleur (tous les points d'une même couleur sont équidistants du centre).

Divergence de Kullback-Leibler (entropie relative)

Beaucoup de mesures classiques expriment la différence entre deux distributions G_1 et G_2 de fonctions de densité respectives g_1 et g_2. Une de ces mesures est la divergence de Kullback-Leibler. Cette mesure est également connue sous le nom d'« Entropie Relative ».

$$D_{KL}(G_1 // G_2) = \sum_D g_1(\mathbf{x}) \log \frac{g_1(\mathbf{x})}{g_2(\mathbf{x})} . \tag{1.7}$$

La formule (1.7) donne l'entropie relative de G_1 par rapport à G_2. La divergence de Kullback-Leibler n'est pas métrique : elle n'est pas symétrique et ne satisfait pas l'inégalité triangulaire. Elle respecte cependant les propriétés de non négativité et réflectivité. Dans le domaine de l'image, elle est très utilisée pour la comparaison de distributions empiriques de caractéristiques visuelles dans le cadre de la recherche d'image par le contenu [Piro et al 2008 ; Goldberger et al. 2003]. On utilise alors souvent une version « symétrisée » en sommant avec la contribution où les rôles de G_1 et G_2 sont inversés.

La distance de Hausdorff

La distance de Hausdorff est un outil topologique qui mesure l'éloignement de deux sous-ensembles dans un espace métrique sous-jacent. Elle mesure la différence entre deux formes. Considérons des formes définies par leur contour fermé. L'idée intuitive de la distance de Hausdorff est de définir la distance entre deux contours finis $A = \{a_1, \ldots, a_p\}$ et $B = \{b_1, \ldots, b_q\}$ comme indiqué sur la figure 1.3. Alors la distance de Hausdorff est définie comme :

$$D_H(A,B) = \max\{d(A,B), d(B,A)\},$$
$$\text{où } d(A,B) = \max_{a\in A} \min_{b\in B} \|a-b\|. \tag{1.8}$$

La distance D_H est le minimum entre *d(A, B)* et *d(B, A)*. D'après [Huttenlocher et al. 1993], elle est considérée comme une mesure de similarité naturelle entre les formes. La distance de Hausdorff $D_H(A,B)$ est nulle si et seulement si $A = B$ et elle augmente lorsque des différences de plus en plus importantes apparaissent entre *A* et *B*. Dans sa forme originale (1.8), la distance de Hausdorff est trop sensible aux valeurs aberrantes.

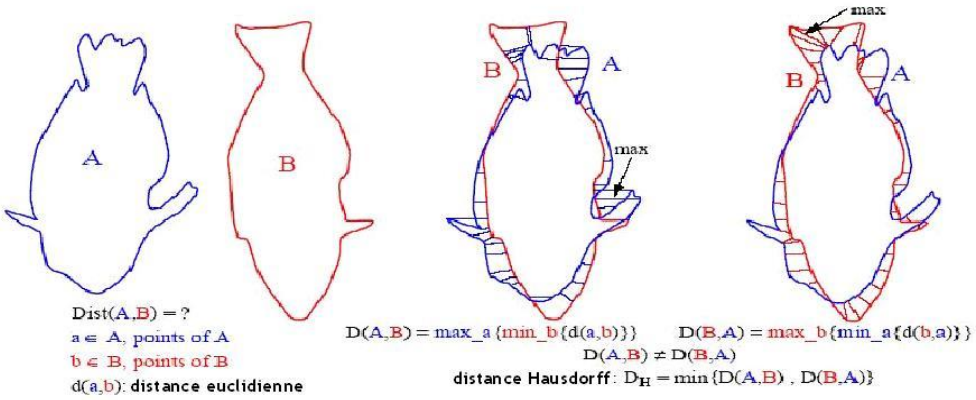


Fig. 1.3 Distance de Hausdorff entre deux contours [http://prtools.org/].

Alors afin de résoudre ce problème Dubuisson et Jain [Dubuisson&Jain 1994] proposent une modification de cette distance. Étant donné deux images binaires, leur distance de Hausdorff modifiée est définie comme le maximum de distances moyennes d'un point sur *A* et *B* :

$$D_{\text{HM}} = \max\left\{\frac{1}{A}\sum_{x\in A} d(x,B), \frac{1}{B}\sum_{x\in B} d(x,A)\right\} \tag{1.9}$$

En prenant la moyenne des distances d'un point, cette version réduit l'effet des valeurs aberrantes en la rendant plus adaptée en reconnaissance de formes car quand le niveau du bruit augmente cette distance se dégrade plus lentement. C'est cette version de la distance de Hausdorff qui sera utilisée dans les expérimentations présentées dans le chapitre 4.

Dynamic time warping

Le « Dynamic time warping » (DTW) est un algorithme permettant de mesurer la similarité entre deux séquences différentes d'échantillonnage temporel (par exemple avec une vitesse différente de déplacement d'un objet dans deux vidéos). Ces séquences peuvent être des signaux discrets ou, plus généralement, des séquences de caractéristiques échantillonnées en temps. Cette distance est souvent appliquée à la vidéo ou à l'audio.

L'objectif de cet algorithme est de trouver l'alignement optimal entre les séquences de longueur variable $T = \{t_1, t_2, \ldots, t_N\}$ et $R = \{r_1, r_2, \ldots, r_M\}$. Suivant [Muller 2007] afin de comparer les deux séquences, on a besoin d'une fonction de coût local ou mesure de distance locale $c(t, r)$. Ce coût est petit si t et r sont similaires. En évaluant la mesure des coûts locaux pour chaque paire d'éléments des séquences T et R, on obtient la matrice des coûts $C \in \Re^{N \times M}$ définie par $C(i, j) = c(t_i, r_j)$. Le but alors est de trouver l'alignement optimal entre les deux séquences ayant le plus petit coût possible. Afin de trouver cet alignement avec le plus faible coût on cumule les coûts élémentaires à travers la matrice des coûts, en commençant par le coin en haut à gauche (correspondant au début des deux signaux), et se terminant au coin inférieur droit (correspondant à la fin des deux signaux). Pour chaque cellule, la distance cumulée est calculée en choisissant la cellule voisine dans la matrice vers la gauche ou au-dessous avec la plus faible distance cumulée. On ajoute alors cette valeur à la distance totale de la cellule. Lorsque ce processus est terminé, la valeur de la cellule au coin inférieur droit représente la distance cumulée (coût total $c_p(T, R)$) entre les deux signaux en suivant le chemin le moins coûteux à travers la matrice.

Un exemple de construction d'une matrice des coûts et d'un alignement optimal entre deux signaux est illustré à la fig.1.4.

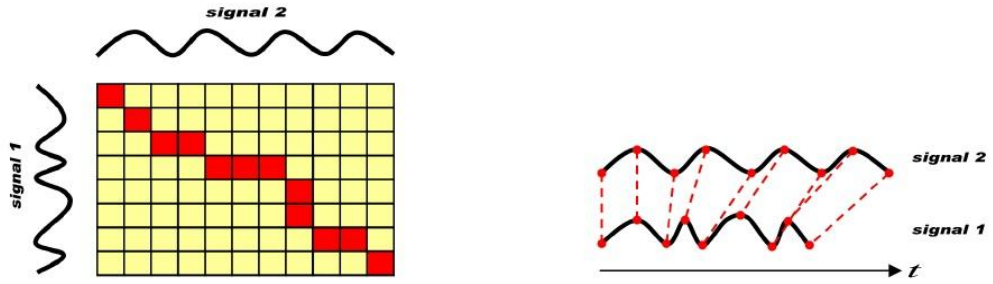


Fig. 1.4 Matrice des coûts entre les deux signaux avec le chemin moins coûteux et l'alignement de deux signaux suivant ce chemin, illustration issue de [Zinke&Mayer 2006].

Alors la distance DWT peut être définie comme suit :

$$D_{\mathrm{DWT}} = \min\{c_p(T, R)\} \tag{1.10}$$

La mesure DTW a certains points faibles. Tout d'abord elle est assez lente. Une distance appropriée (coût) doit être choisie pour les calculs ce qui n'est pas trivial en fonction des invariances à mettre en place. Généralement cette distance n'est pas définie positive et ne satisfait pas l'inégalité triangulaire même si $c(t_i, r_j)$ est métrique. La DWT est symétrique si $c(t_i, r_j)$ est symétrique. Un point intéressant de la distance DTW est qu'elle peut se calculer entre des séquences de tailles différentes. Cette propriété est très utile en reconnaissance des formes (lettres, chiffres manuscrits, …) car cela permet une comparaison sans re-échantillonnage [Niels 2004].

Distance tangentielle

Le classement des objets exige que la dissimilitude à partir des représentations soit invariante à l'égard de la position, des changements de taille, de rotation plus ou moins

importante, de distorsions, etc. Pour cela, la distance Euclidienne entre deux objets n'est pas toujours appropriée à cause de son manque de robustesse par rapport aux distorsions.

Pour introduire la notion de distance tangentielle, considérons *P* et *E* - deux images. Alors l'invariance concerne par exemple la translation, la rotation, le zoom… Soit *s*, une fonction qui transforme l'image *P* en une image $s(P,\alpha)$ en fonction du paramètre α, et telle que la fonction *s* soit dérivable par rapport à *P* et α et $s(P,0) = P$. En reprenant notre exemple, $s(P,\alpha)$ pourrait être une rotation par un angle α_θ suivie d'une translation par α_x et α_y de l'image *P*. Dans ce cas $\boldsymbol{\alpha}=(\alpha_\theta , \alpha_x , \alpha_y)$ est un vecteur 3D (voir fig. 1.5). Dans le cas général $\boldsymbol{\alpha}$ est un vecteur de dimension *m*. L'idée clé est que, lorsqu'elles sont soumises à des transformations spatiales, les images décrivent des surfaces dans un espace de dimensions *m*. Une métrique invariante à ses transformations doit mesurer la distance entre ces surfaces au lieu de la distance entre des caractéristiques extraites à partir des images. Parce que ces surfaces sont complexes, alors la minimisation de la distance entre eux est un problème d'optimisation difficile qui peut, néanmoins, être résolu en considérant la minimisation de la distance entre les hyperplans tangents à ces surfaces. On obtient alors la distance tangentielle (TD).

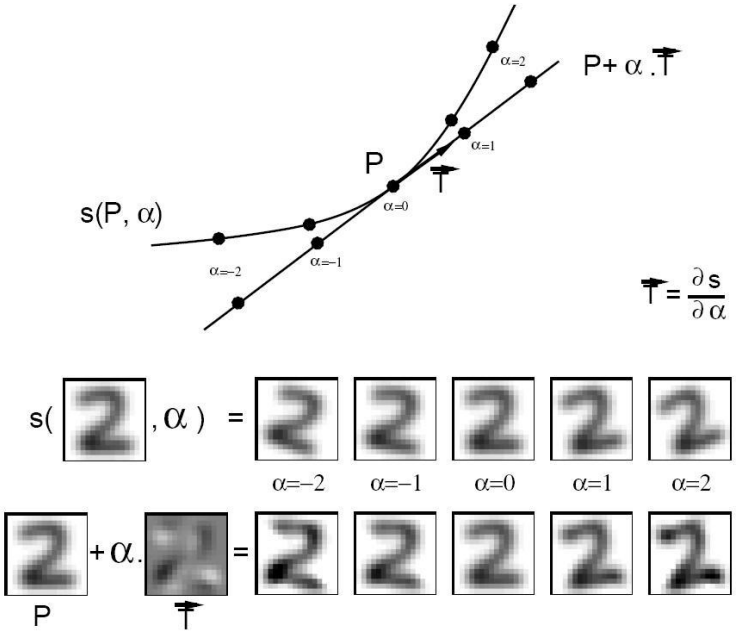


Fig. 1.5 Exemple d'une transformation par rotation d'une image du chiffre manuscrit « 2 » s(P, α) et la représentation de la courbe tangentielle associée avec la courbe de l'ensemble des images pour chaque angle [Simard et al. 1996].

Parce que $s(P,\alpha)$ est dérivable, alors l'ensemble de tous les objets transformés $S_P = \{x \mid \exists \alpha \text{ pour lequel } x = s(P,\alpha)\}$ est une variété géométrique dérivable qui par approximation du premier ordre peut être représentée par un hyperplan T_P. Cet hyperplan est tangent à S_P en *P* (voir fig. 1.6). Si on doit comparer les deux objets *P* et *E*, alors les deux courbes tangentielles T_P et T_E peuvent être utilisées pour construire la distance tangentielle [Simard et al. 1996] :

$$D_{TD}(E,P) = \min_{x \in T_P, y \in T_E} \|\mathbf{x} - \mathbf{y}\|^2 \qquad (1.11)$$

Trouver la distance minimale entre deux variétés géométriques linéaires est un problème d'optimisation par les moindres carrés. Cette distance est invariante pour des transformations locales mais pas globales.

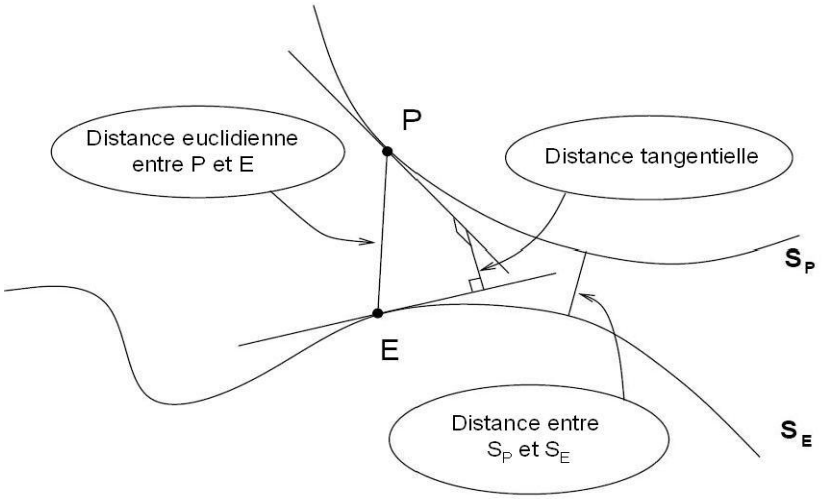


Fig. 1.6 Illustration de la distance Euclidienne et la distance tangentielle entre P et E. Les courbes S_E et S_P sont obtenues après une transformation choisie de P et E. Les lignes passant par P et E sont les tangentes de ces courbes [Simard et al 1996].

"Earth Mover's Distance" (EMD) et "Proportional Transportation Distance" (PTD)

La distance EMD est introduite récemment comme une mesure de distance pour la recherche des images par le contenu [Rubner et al. 2000], mais c'est une mesure beaucoup plus ancienne, issue de la recherche opérationnelle. Plus particulièrement, il s'agit d'une distance entre deux ensembles de points pondérés. Elle mesure la quantité minimum de travail nécessaire pour transformer l'un dans l'autre.

Soit un ensemble de points pondérés $A = \{a_1, a_2, ..., a_m\}$ tel que $a_i = \{(x_i, w_i)\}, i = 1, ..., m$ où $x_i \in \Re^k$ et $w_i \geq 0$ est son poids correspondant. Soit $W = \sum_{j=1}^{n} w_j$ le poids total de l'ensemble *A*. Idem pour l'ensemble *B*, avec *U* son poids total. De manière intuitive, un point pondéré peut être considéré comme une quantité de terre dont son volume est sa pondération et son emplacement est la position de ce point dans l'espace. Pour les points du second ensemble, l'analogie est semblable en considérant cette fois-ci le point comme un trou. La distance EMD s'interprète comme la quantité minimale de travail à faire pour déplacer les tas de terre afin de combler les trous [Typke et al. 2003].

La distance EMD peut être formulée comme un problème de programmation linéaire où l'on minimise la somme des produits des flux de terre [h_{ij}] par les distances de déplacements [d_{ij}]. Etant donnés les deux ensembles *A* et *B* et une distance *d*, on note avec h_{ij} le flux élémentaire de la masse de x_i à y_j sur la distance d_{ij}, *W* le poids total de l'ensemble *A* et *U*, celui de l'ensemble *B*. L'ensemble des tous les flux possibles $H = [h_{ij}]$ est défini par des contraintes (non négativité, ne pas émettre ou recevoir un flux plus important que sa pondération). Alors, le poids total transporté est le minimum du poids total des deux ensembles, et la distance EMD entre ces deux ensembles est la suivante :

$$D_{\mathrm{EMD}}(A,B) = \frac{\min_F \sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij} d_{ij}}{\min(W,U)} \tag{1.12}$$

Cette distance est très utilisée en indexation d'images par le contenu, à partir des premiers travaux de Rubner [Rubner et al. 2000]. Elle apporte un gain certain pour la comparaison de distributions empiriques de caractéristiques, par rapport aux mesures de distances L_2 ou L_1 classiquement utilisées dans ce contexte. Le point essentiel est la prise en compte d'une distance d_{ij} qui représente ici l'écart entre les centres des intervalles des histogrammes estimant les distributions empiriques. Les poids sont naturellement les effectifs associés à chaque intervalle. Un autre exemple de l'application de la distance EMD est la recherche des contours similaires dans des bases de données d'images [Grauman&Darell 2004].

Suivant [Typke 2011] les propriétés les plus importantes de cette distance sont :

- EMD est métrique si la distance d_{ij} est métrique et si elle est appliquée à des ensembles dont les poids globaux sont égaux.
- Elle est continue c'est-à-dire des petits changements de position ou de poids des points provoquent des petits changements de sa valeur.
- Elle ne respecte pas la propriété de non négativité si les sommes des poids des deux ensembles sont différentes.
- Dans le cas où les sommes des poids des deux ensembles sont différentes, elle ne respecte pas la propriété d'inégalité triangulaire.

La distance PTD "Proportional Transportation Distance" [Typke 2011 ; Giannopoulos&Veltkamp 2002] est une modification de la distance EMD où l'excédent ou l'insuffisance de poids entre deux ensembles est pris en compte et la propriété de l'inégalité triangulaire est alors respectée. Elle porte le nom de « Proportional Transportation Distance » car tout excédent ou insuffisance de poids doit être retiré de telle façon que les proportions soient sauvegardées et puis on calcule l'EMD.

$$D_{PTD}(A,B) = \frac{\min_F \sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij} d_{ij}}{W} \tag{1.13}$$

Il en résulte une distance pseudo-métrique car elle obéit à l'inégalité triangulaire. Mais elle ne dispose toujours pas de la propriété de positivité car la distance entre des ensembles dont la position spatiale coïncide avec les mêmes pourcentages de poids aux mêmes positions est nulle, ces ensembles diffèrent seulement par leur poids correspondant individuel. Toutefois, il s'agit du seul cas dans lequel la distance entre deux ensembles de points non identiques est nulle. Cette distance possède toutes les autres propriétés de la distance EMD pour des ensembles avec les poids totaux égaux.

1.3. Méthodologie de classement par dissimilitudes

La représentation des objets par dissimilitudes ouvre des nouvelles possibilités pour classement dans le domaine de l'apprentissage statistique. Un certain nombre de chercheurs ont apporté des fortes contributions dans ce domaine [Jacobs et al. 2000]. On citera plus particulièrement Elzbieta Pekalska et Robert Duin [Pekalska&Duin 2005 ; Pekalska et al. 2006 ; Pekalska&Duin 2008 ; Pekalska&Duin 2002] ainsi que Bernard Haasdonk [Haasdonk&Burkhardt 2007 ; Haasdonk&Bahlmann 2004] et également la collaboration entre Pekalska et Haasdonk [Pekalska&Haasdonk 2010 ; Pekalska&Haasdonk 2009 ; Pekalska&Haasdonk 2008].

Soient un ensemble d'apprentissage étiqueté de N objets $X = \{x_1, x_2, \ldots, x_N\}$, et la matrice de dissimilitudes $D(X, X)$ telle que :

$$D(X,X) = \begin{array}{c} \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & \cdot\,\cdot & \cdot\,\cdot\,\cdot & x_N \end{array} \\ \begin{pmatrix} d_{11} & d_{12} & \cdot & \cdot & d_{1N} \\ d_{21} & d_{22} & \cdot & \cdot & d_{2N} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ d_{N1} & d_{N2} & \cdot & \cdot & d_{NN} \end{pmatrix} \end{array}. \qquad (1.14)$$

La plupart des algorithmes de classement supposent une matrice de dissimilitudes D symétrique, à valeurs non négative et à diagonale nulle. Le problème est de classer un nouvel objet non étiqueté. Cet objet est représenté par les dissimilitudes $d_x = \{d_1, d_2, \ldots, d_N\}$ de cet objet à tous les points de la base d'apprentissage.

Les différentes approches pour le classement de ce point peuvent se regrouper en trois familles, illustrées à la figure 1.7.

1. *L'approche par recodage des données de dissimilitudes*. La méthode proposée par Pekalska et Duin est de réduire l'ensemble d_x par un choix de prototypes représentatifs de la base d'apprentissage. La dimension de l'espace vectoriel de recodage est égale au nombre de prototypes. Les classifieurs sont ensuite appliqués après cette étape de recodage vectoriel. Différents classifieurs sont considérés, comme par exemple le classifieur linéaire ou le classifieur quadratique [Pekalska et al. 2002].

2. *L'approche par le positionnement multidimensionnel*. Il s'agit de rechercher par des techniques linéaires ou non linéaires de positionnement multidimensionnel, l'espace vectoriel sous-jacent aux données de dissimilitude. A partir de là, les classifieurs standard sur données vectorielles peuvent s'appliquer.

3. *L'approche directe dans l'espace des dissimilitudes*. La règle de classement au plus proche voisin est la plus populaire dans ce contexte. Les classifieurs à vecteurs de support peuvent être également être utilisés avec des noyaux appropriés.

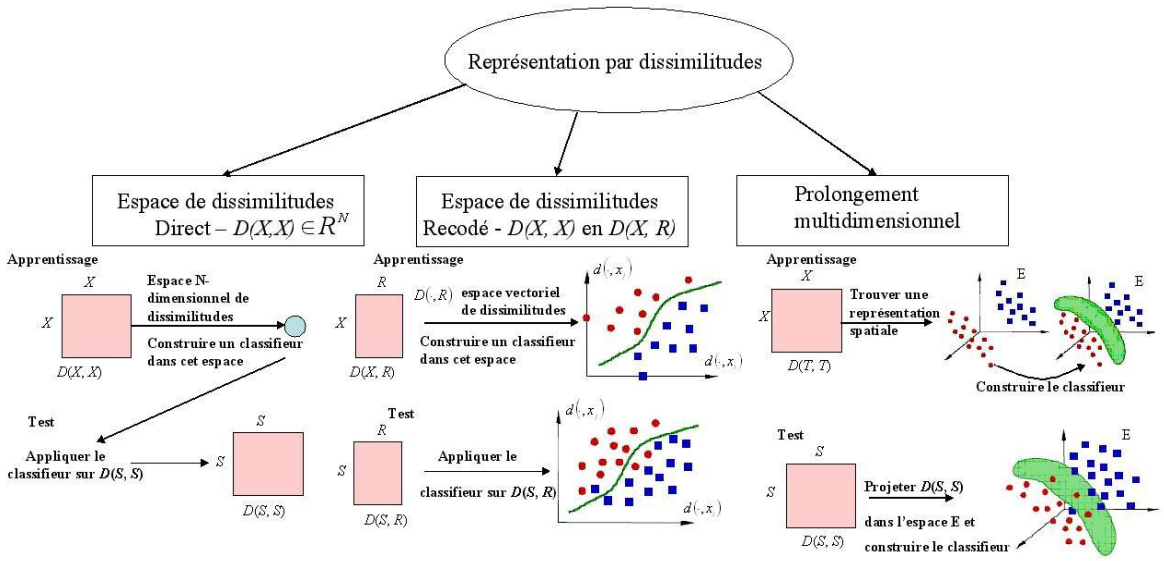


Fig 1.7 Représentation visuelle des trois approches de classement sur des données de dissimilitudes.

Dans le paragraphe qui suit on présentera les caractéristiques essentielles des matrices de dissimilitudes afin de pouvoir introduire les différentes approches de classement basées sur de telles matrices. On présentera d'abord les techniques de recodage de l'espace des dissimilitudes en espace vectoriel des dissimilitudes ou espace des caractéristiques. Après des modifications appropriées, cela permet l'application des classifieurs connus tels que le classifieur linéaire, quadratique, classifieur de Fisher, classifieur de Parzen et autres. Il a été expérimentalement démontré que beaucoup de techniques de classement classiques peuvent être utilisées après une transformation appropriée sur les mesures de dissimilitudes et donner des résultats satisfaisants [Pekalska&Duin 2006 ; Pekalska&Duin 2010]. On présentera l'approche par prototypes de Pekalska et Duin [Pekalska et al. 2006 ; Pekalska&Duin 2002].

Puis on introduira la technique de prolongement multidimensionnel pour des distances métriques et non métriques.

Finalement on présentera les classifieurs directement dans l'espace des dissimilitudes qui font objet de cette étude - la règle du plus proche ou des K plus proches voisins, la technique la plus répandue et la plus simple à appliquer, les machines à vecteurs de support linéaires et les machines à vecteurs de support aux moindres carrés. Finalement on introduira les machines à vecteurs de support non linéaires avec les noyaux de distances appropriés de Haasdonk [Haasdonk 2005 ; Haasdonk&Bahlmann 2004].

1.4. Recodage de l'espace des dissimilitudes

1.4.1. Propriétés de la matrice de dissimilitudes

Ce paragraphe vise à préciser certaines propriétés des matrices de dissimilitudes utiles à la mise en place des algorithmes de classement. La théorie présentée ci-dessous provient de [Gower&Legendre 1986] et [Pekalska 2005].

On considère une matrice de dissimilitudes $D = (d_{ij})$ (1.14) symétrique, à valeurs non négatives et à diagonale nulle. La matrice D a N^2 coefficients mais seulement $N(N - 1)/2$ différents.

Propriété 1. D est métrique si et seulement si l'inégalité triangulaire $d_{ij} + d_{ik} \geq d_{jk}$ est valable pour tous les triplets (i, j, k).

Quelques conclusions simples suivent de cette propriété. Considérant le triplet (i, j, j) on peut démontrer que $d_{ij} \geq 0$ pour toutes les paires (i, j). Considérant les triplets (i, j, i) et (j, i, j), alors $d_{ij} \geq d_{ji}$ et $d_{ji} \geq d_{ij}$ c'est-à-dire toutes les matrices de dissimilitudes sont symétriques avec des éléments non négatifs. Supposons $d_{ij} = 0$ alors en considérant les triplets (i, k, j) et (j, k, i) on a $d_{ik} = d_{jk}$ pour tous les k. Cette dernière relation démontre que si deux objets sont similaires (d_{ij} proche de zéro) alors un troisième objet, k, aura une relation similaire avec ces deux objets (d_{ik} et d_{jk} ne diffèrent que légèrement). Cela signifie que l'un d'eux peut devenir un prototype pour représenter tous les deux. Cette propriété permet une construction de la recherche approximative du plus proche voisin dans un espace euclidien. Chaque triplet métrique d_{ij}, d_{ik} et d_{kj} est euclidien, c'est-à-dire ils constituent un triangle euclidien. Toutefois pour $N > 3$, certaines $N \times N$ matrices de distances métriques D n'ont pas une représentation euclidienne.

Propriété 2. Si D est semi-métrique alors il est possible de trouver une matrice D' qui le soit : $D' = D + c\left(\mathbf{1}\mathbf{1}^T - I\right)$ avec $c \geq \max_{p,q,r} \left| d_{pq} + d_{pr} - d_{qr} \right|$.

C'est une propriété utile car elle permet de transformer une matrice quasi métrique en une matrice métrique. Si la correction c appliquée à chaque terme est relativement faible, alors la matrice de dissimilitudes D initiale n'est que légèrement non métrique. Si, toutefois, c est grand, alors la distorsion devient importante et c'est plutôt en aval, à l'algorithme de classement de tenir compte de ces propriétés non métriques.

Une question importante est la transformation d'une matrice de dissimilitudes de telle façon que les vertus métriques soient préservées. La propriété suivante montre des exemples de telles transformations. Cette propriété est nécessaire dans le cas où on doit faire des normalisations de la matrice de dissimilitudes pour les classifieurs SVM par exemple ou pour le prolongement multidimensionnel.

Propriété 3. Si D est métrique et f, une fonction non décroissante et concave telle que $f(0) = 0$ et $f(x) > 0$ pour $t > 0$ alors $D_f = [f(d_{ij})]$ est aussi métrique. Par conséquent si D est métrique, alors pour $c > 0$, les matrices de dissimilitudes définies comme : ($c\ d_{ij}$), (min {1, d_{ij}}), (d_{ij}^r) avec $r \in (0, 1]$, ($d_{ij}/(d_{ij} + c)$), (sign (d_{ij})), (log(1 + d_{ij})) sont aussi métriques;

Propriété 4. La matrice D de dimensions $N \times N$ est euclidienne si elle peut être projetée dans un espace euclidien $\Re^m$ tel que $m < n$. Cela signifie qu'il existe une configuration $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ dans $\Re^m$ telle que $||\mathbf{x}_i - \mathbf{x}_j||^2 = d_{ij}$.

Cette propriété est à l'origine des techniques de prolongement multidimensionnel, que l'on décrira dans la suite de ce chapitre.

Propriété 5. Supposons N vecteurs $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. En développant l'équation de la distance euclidienne, on montre facilement que :

$$d^2(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i ; \mathbf{x}_j \rangle + \langle \mathbf{x}_i ; \mathbf{x}_j \rangle - 2\langle \mathbf{x}_i ; \mathbf{x}_j \rangle = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\langle \mathbf{x}_i ; \mathbf{x}_j \rangle , \tag{1.15}$$

$$\langle \mathbf{x}_i ; \mathbf{x}_j \rangle = -\frac{1}{2}\left[d^2(\mathbf{x}_i, \mathbf{x}_j) - \|\mathbf{x}_i\|^2 - \|\mathbf{x}_j\|^2 \right]. \tag{1.16}$$

En se basant sur les propriétés du produit scalaire et la formule (1.16) et après des transformations appropriées (voir [Pekalska&Duin, 2005]), on obtient :

$$\langle \mathbf{x}_i ; \mathbf{x}_j \rangle = -\frac{1}{2}\left[\begin{array}{l} d^2(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{N}\sum_{s=1}^{N} d^2(\mathbf{x}_i, \mathbf{x}_s) - .... \\ -\frac{1}{N}\sum_{s=1}^{N} d^2(\mathbf{x}_s, \mathbf{x}_j) - \frac{1}{N^2}\sum_{p,s=1}^{N} d^2(\mathbf{x}_p, \mathbf{x}_s) \end{array} \right]. \tag{1.17}$$

Cette propriété est nécessaire pour les noyaux des SVM de Haasdonk [Haasdonk&Bahlmann 2004], décrits plus tard dans le chapitre. Aussi il y a une liaison directe entre la matrice des distances euclidiennes au carré et la matrice de Gram qui est calculée dans le paragraphe suivant pour le positionnement multidimensionnel (voir [Pekalska 2005]).

On doit noter que le même raisonnement est valable pour un espace pseudo euclidien, car la formulation linéaire entre les distances au carré et les produits scalaires dans les deux espaces est identique.

1.4.2. Technique de représentation par prototypes de Pekalska & Duin

L'équipe d'Elzbieta Pekalska et de Robert Duin à l'Université de Delft au Pays Bas a développé de nombreuses méthodes de classement à partir des matrices de dissimilitudes. Mais leur principal apport dans l'analyse discriminante est la technique de sélection de prototypes. L'idée est d'étendre les relations de proximité « point à point » pour caractériser la proximité d'un objet avec un groupe d'objets.

Supposons une collection de n objets $R = \{p_1, p_2, \ldots, p_n\}$, appelée ensemble de représentation ou ensemble de prototypes et une mesure de dissimilitude d. Cette mesure peut être non métrique mais il est nécessaire qu'elle soit non négative et que la propriété de réflectivité soit vérifiée. Une représentation par dissimilitude d'un objet x est un ensemble de dissimilitudes entre x et les objets de R. Cette représentation est exprimée en tant que vecteur $D(x, R) = [d(x, p_1), d(x, p_2), \ldots, d(x, p_n)]$ dans $\Re^n$. Par conséquent pour un ensemble T de N objets, on peut construire un tableau de dissimilitudes $D(T, R)$ de taille $N \times n$. L'intérêt de l'approche réside dans le fait où la taille de

l'ensemble *R* des prototypes est relativement petite vis-à-vis de la complexité du groupe d'objets qu'il est sensé représenter. *R* est soit un sous ensemble de *T* et dans ce cas les prototypes seront sélectionnés dans *T* (voir fig. 1.8) ou soit, un ensemble totalement distinct.

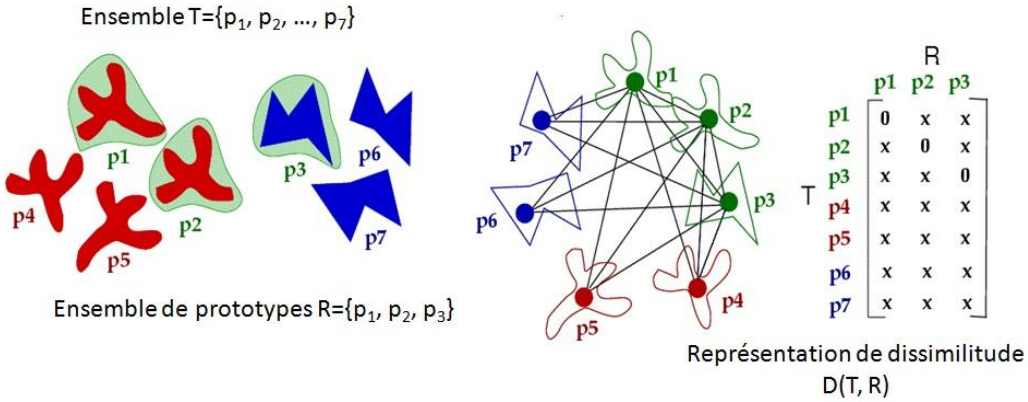


Fig. 1.8 La représentation à partir des dissimilitudes D(T, R), les prototypes sont des éléments de l'ensemble T [Pekalska 2005].

Dans l'approche de Pekalska et Duin [Pekalska&Duin 2005], une représentation par dissimilitudes $D(T, R)$ est vue comme une projection $D(\cdot, R): X \rightarrow \Re^n$, d'un objet x dans un espace initial, vers l'espace des dissimilitudes spécifié par l'ensemble R. Dans un tel espace, chaque dimension correspond à la dissimilitude avec un prototype de R, c'est à dire $D(\cdot, p_i)$. Parce que les dissimilitudes sont positives, tous les objets sont projetés comme des points dans le quadrant des coordonnées positives de l'espace vectoriel à n dimensions (voir fig. 1.9).

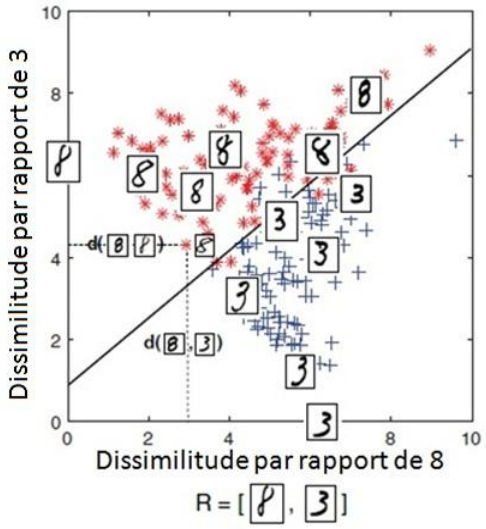


Fig. 1.9 Exemple d'un espace de dissimilitude 2D et un classifieur linéaire sur un sous-ensemble de chiffres manuscrits « 3 » et « 8 » extrait de la base NIST [Federal Register Document Image Database]. La mesure de dissimilitude D(T, R) utilise la distance euclidienne entre des images binaires. R est choisi arbitrairement et contient deux prototypes, un pour chaque classe [Pekalska 2005].

Les classifieurs seront implémentés dans cet espace. La discrimination vient du fait que les dissimilitudes sont faibles pour les objets similaires (appartenance à la même classe) et

grandes pour les objets distincts (appartenant à des classes différentes). Ainsi l'ensemble de représentation doit être choisi afin que pour deux objets similaires x et y, leurs vecteurs représentatifs $D(x, R)$ et $D(y, R)$ soient corrélés, c'est-à-dire proches dans l'espace des dissimilitudes. Les prototypes p_i sont des observations typiques des classes et non pas des observations aberrantes.

Notons que cet espace de représentation peut être utilisé avec des dissimilitudes métriques et non métriques. Il construit un espace de caractéristiques, qui sont toutes de même nature (dissimilitude à un prototype donné), contrairement à un espace de caractéristiques standards où sont présentées très souvent des caractéristiques de nature différente.

Une partie importante du travail de Pekalska et Duin concerne le choix de l'ensemble des prototypes. Le choix d'un ensemble de représentation prévu pour la construction des classifieurs dans un espace de dissimilitudes sert un objectif : la minimisation de l'ensemble des dissimilitudes à mesurer pour le classement des nouveaux objets. La sélection de l'ensemble de représentation va définir un espace de dissimilitudes dans lequel l'ensemble d'apprentissage entier est utilisé pour former un classifieur. Pour cette raison, même un ensemble de représentation choisi au hasard peut être utile [Pekalska&Duin 2005]. Les sept différentes procédures de sélection de prototypes : *Random*, *RandomC*, *KCentres*, *ModeSeek*, *LinProg*, *FeatSel*, et *EdiCon*, proposées par Pekalska, sont comparées dans [Pekalska et al. 2006 ; Pekalska&Duin 2002 ; Sang&Oommen 2007]. Les premières deux procédures sont des procédures de choix aléatoire des prototypes. Les autres sont des procédures systématiques. Pekalska et ses collègues ont constaté en utilisant différentes bases de données de dissimilitudes que la performance des procédures systématiques est la plupart des fois meilleure que celle des aléatoires. Parmi les procédures qui ont un contrôle sur le nombre des prototypes sélectionnés celle avec la meilleure performance en général c'est *KCentres*. Cette procédure est presque identique à l'algorithme « k-means », appliqué dans un espace vectoriel. La méthode de programmation linéaire *LinProg* a une bonne performance dans le cas de classement à deux classes. Les autres sont utilisées dans le cas de classement multi-classes.

On décrira dans le paragraphe suivant, les principaux classifieurs utilisés par Pekalska et Duin dans cet espace de représentation par dissimilitudes. Les classifieurs linéaire et quadratique sont pour cela d'excellents candidats. En effet, la plupart des mesures de dissimilitude couramment utilisées, comme la distance euclidienne, de Minkowsky ou la distance EMD, sont calculées à partir de sommes de contributions élémentaires. Si cette somme peut être interprétée comme la somme de variables aléatoires indépendantes et identiquement distribuées, alors le théorème central limite s'applique et cette somme tend vers une loi gaussienne. Ainsi comme les classifieurs linéaire et quadratique sont optimaux pour des distributions gaussiennes décrivant chaque classe, ils sont tout naturellement utilisés dans ce contexte de dissimilitudes.

Classificateur linéaire (CL) et classificateur quadratique (CQ)

Pour simplifier l'écriture et sans réduire la généralité du propos, on décrit les équations dans le cas d'une discrimination entre deux classes (ω_1 et ω_2). Soit f la fonction discriminante permettant de décider si l'observation appartient à l'une ou l'autre des deux classes. Pour un système multi-classes, on définit autant de fonctions discriminantes f_k qu'il y a de classes.

L'observation sera assignée à la classe dont l'image par la fonction discriminante associée sera maximale. Ainsi pour un problème à deux classes, la règle de décision peut s'écrire à l'aide d'une seule fonction discriminante *f*. Soient $p(\omega_1)$ et $p(\omega_2)$, les probabilités a priori pour chacune des deux classes. Soient $\mathbf{m}_1$ et $\mathbf{m}_2$ les observations moyennes estimées pour chacune des deux classes et Σ la matrice de variance - covariance globale. Alors la fonction discriminante *f*, appliquée à une observation décrite par son vecteur de dissimilitudes $\mathbf{D}(x, R)$ sur l'ensemble *R* des prototypes, s'écrit ainsi :

$$f(\mathbf{D}(x,R)) = \left[\mathbf{D}(x,R) - \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)\right]^{\mathrm{T}} \Sigma^{-1}(\mathbf{m}_1 - \mathbf{m}_2) + \log\frac{p(\omega_1)}{p(\omega_2)} \qquad (1.18)$$

Le signe de *f* indique la classe de l'observation *x*. Cette règle d'affectation définit une séparatrice linéaire dans l'espace vectoriel à *n* dimensions (*n* étant le nombre de prototypes). Le dernier terme relatif aux probabilités a priori est issu de la théorie des classifieurs Bayesiens, il agit comme un curseur déplaçant la frontière vers la classe moins probable.

L'hypothèse sous-jacente du classifieur quadratique est que les observations pour chacune des classes suivent des distributions gaussiennes de matrices de variance - covariance différentes. Alors la fonction discriminante s'écrit ainsi :

$$\begin{aligned} f(\mathbf{D}(x,R)) &= \sum_{i=1}^{2}(-1)^i(\mathbf{D}(x,R)-\mathbf{m}_i)^T\Sigma_i^{-1}(\mathbf{D}(x,R)-\mathbf{m}_i) \\ &+ \log\frac{p(\omega_1)}{p(\omega_2)} + \frac{1}{2}\log\frac{\det(\Sigma_1)}{\det(\Sigma_2)} \end{aligned}, \qquad (1.19)$$

où Σ_1 et Σ_2 sont les matrices de covariance estimées pour chaque classe et $\Sigma = (\Sigma_1 + \Sigma_2)/2$ est la matrice de covariance totale, les trois matrices sont déterminées dans l'espace de dissimilitudes. Si Σ (ou Σ_1 et Σ_2) est singulière, son inverse ne peut pas être calculée. Alors on doit régulariser la matrice Σ. Les méthodes de régularisation sont mentionnées dans [Pekslaska&Duin 2005].

Le classifier linéaire de la moyenne plus proche (NMC)

Quand la matrice de covariance Σ est la matrice d'unité, le classifieur CL devient le classifieur linéaire de la moyenne plus proche qui attribue l'objet à la classe de son plus proche vecteur moyen au sens euclidien. Si Σ est une matrice diagonale alors la décision résultante correspond au classifieur pondéré de la moyenne plus proche.

Le classifieur de Fisher (FLC)

C'est un classifieur linéaire [Fukunaga 2001 ; Duda et al. 2000] où le vecteur de poids définissant l'hyperplan séparateur est déterminé par maximisation du critère de Fisher $J(\mathbf{w})$, défini par :

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \Sigma_B \mathbf{w}}{\mathbf{w}^T \Sigma_W \mathbf{w}}, \tag{1.20}$$

avec $\Sigma_B = \sum_{k=1}^{K} n_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T$ la dispersion inter-classe et Σ_w la dispersion intra-classe (la somme des matrices de covariance de chaque classe). Le vecteur de poids qui maximise $J(\mathbf{w})$ est $\mathbf{w} = \Sigma_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$. Il est bien connu que pour deux classes équiprobables le classifieur de Fisher correspond au classifieur linéaire. Alors pour une représentation de dissimilitude $D(T, R)$, il peut être représenté comme :

$$f(D(x,R)) = \left[D(x,R) - \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2) \right]^T \Sigma_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2) + \log \frac{p(\omega_1)}{p(\omega_2)} \tag{1.21}$$

Comme précédemment si la matrice de covariance Σ_w est singulière alors elle peut être régularisée.

1.4.3. Méthode du Positionnement Multidimensionnel

La méthode du positionnement multidimensionnel (MDS) se réfère à un groupe de méthodes de projection linéaires et non linéaires des dissimilitudes. La théorie de MDS a été initialement développée pour l'analyse des données en psychologie et sociologie [Kruskal&Wish 1978]. En effet, les objets d'étude (les individus) ne sont pas décrits en terme individuel mais les uns par rapport aux autres par la mesure de leur différence deux à deux. Ces applications ont été étendues à la reconnaissance des formes, car la méthode MDS facilite la visualisation de données et leur exploration.

Suivant [Borg&Groenen 1997], les quatre buts de MDS sont :

- Représentation des dissimilitudes ou similitudes entre les données en tant que distances dans un espace de faibles dimensions afin de rendre ces données accessibles à l'inspection visuelle et à l'exploration ;
- Evaluation de certains critères par lesquels on peut distinguer les différents objets d'intérêt : comment sont-ils modifiés par des différences empiriques correspondantes à ces objets ;
- Analyse des données qui permet de découvrir les dimensions qui structurent les jugements de similitude ou dissimilitude ;
- Modélisation en psychologie expliquant des jugements de dissimilitude du point de vue d'une règle qui imite un type particulier de distance.

La méthode de positionnement multidimensionnel vise à représenter au mieux des objets dans un espace visualisable, de façon à ce que les distances entre les objets dans cet espace soient aussi proches que possible des dissimilitudes initiales. Une telle configuration se trouve généralement dans un espace euclidien, bien que tout autre espace $\Re^p$, avec $p \geq 1$, puisse aussi être considéré. Par conséquent, le résultat d'un algorithme MDS est une représentation spatiale

des données. La plupart des concepts présentés ci-dessous, ainsi que la discussion sur les algorithmes de calcul des MDS peuvent être trouvés dans les livres [Borg&Groenen 1997 ; Cox&Cox 2001].

Parmi les techniques de positionnement multidimensionnel, les approches métriques sont couramment opposées aux approches non métriques. Les premières produisent des représentations préservant au mieux l'information quantitative contenue dans les données, alors que les secondes privilégient l'information qualitative [Borg&Groenen 1997 ; Cox&Cox 2001]. Pour le MDS non métrique, l'objectif est d'établir une relation monotone entre les distances des points et les similitudes obtenues. L'avantage de MDS non métrique est qu'aucune des hypothèses ne doit être faite sur la fonction de transformation sous-jacente. La seule hypothèse est que les données soient mesurées au niveau ordinal.

Enfin, on distingue le MDS déterministe et celui probabiliste. Pour le MDS déterministe, chaque objet est représenté par un point unique dans un espace multidimensionnel [Borg&Groenen 1997], tandis que pour le MDS probabiliste [MacKay&Zinnes 1986], chaque objet est représenté comme une distribution de probabilités dans l'espace multidimensionnel. Cette dernière approche est utile lorsque la représentation des objets est supposée être bruitée et comportant donc des indéterminations.

MDS sur des distances métriques

La technique de positionnement multidimensionnel métrique sert à fournir une représentation euclidienne des matrices de dissimilitudes supposées elles aussi métriques. Le positionnement multidimensionnel est un prolongement euclidien des mesures de dissimilitudes en recherchant la position euclidienne des observations $\mathbf{x}_i$ (vecteur) et $\mathbf{x}_i$ telle que la dissimilitude d_{ij} soit représentée par la distance euclidienne entre $\mathbf{x}_i$ et $\mathbf{x}_i$ où $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ est la représentation des objets de l'espace $\Re^n$ dans l'espace euclidien $\Re^k$ où $k < n$. Il y a différentes façons de préserver la structure des données, donnant naissance à des différentes techniques de MDS. L'algorithme linéaire MDS est la technique la plus simple. Elle sera introduite dans cette section.

Si la matrice $D \in \Re^{n \times n}$ est une matrice de distance euclidienne, le positionnement multidimensionnel classique revient à une Analyse en Composantes Principales (ACP) à une rotation près [Pekalska 2005]. C'est une technique linéaire. On trouve la position des observations comme après une ACP où il faudrait choisir la dimension de l'espace de sortie.

L'algorithme MDS métrique est le suivant :

- Carré de la matrice de distance D: $D^2 = [d_{ij}^2]$;
- Centrage en ligne et en colonnes par l'opérateur J :

$$J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \in \Re^{n \times n}, \tag{1.22}$$

avec $\mathbf{1}$ le vecteur unitaire, T opération de transposition et I, la matrice identité. L'opérateur J st un opérateur de centrage telle que les données projetées soient à moyenne nulle.

- Calcul du produit scalaire B où $B=XX^T$ – matrice définie positive des produits scalaires:

$$B = -\frac{1}{2} JD^2 J \,. \tag{1.23}$$

- Décomposition en valeurs propres de la matrice B :

$$XX^T = B = Q\Lambda Q^T \,, \tag{1.24}$$

où Λ est une matrice diagonale contenant les premières valeurs propres non négatives λ, rangées dans un ordre décroissant suivies par des valeurs zéros ; Q est une matrice orthogonale des vecteurs propres, et X est la représentation que l'on recherche.

Pour $k < n$ valeurs propres non négatives, une représentation X à k dimensions peut être trouvée:

$$X = Q_k \Lambda_k^{\frac{1}{2}}, \quad Q_k \in \Re^{n \times k}, \quad \Lambda_k^{\frac{1}{2}} \in \Re^{k \times k} \,, \tag{1.25}$$

où Q_k est la matrice des k premiers vecteurs propres et $\Lambda_k^{\frac{1}{2}}$ contient les racines carrées des valeurs propres correspondantes.

Un exemple de représentation MDS en 2D à partir de dissimilitudes entre images est représenté à la figure 1.10. Dans cet exemple, les dissimilitudes d'origine sont calculées à partir des divergences de Kullback-Leibler sur des distributions d'orientations extraites des images.

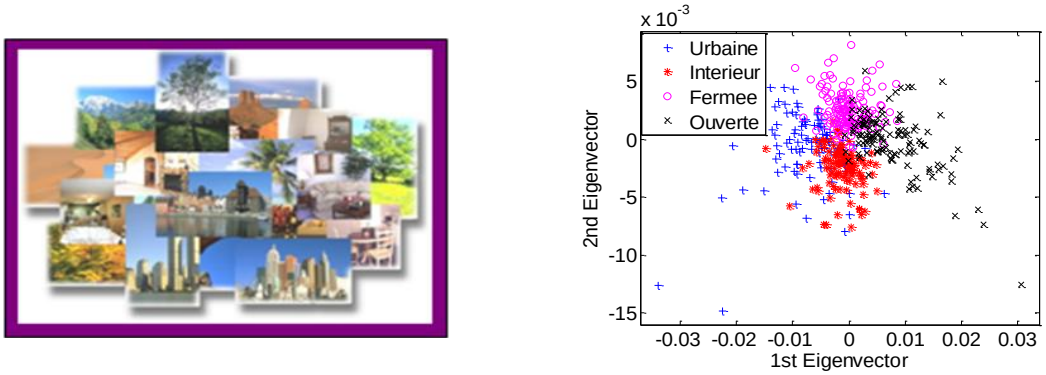


Fig. 1.10 Visualisation 2D par MSD d'une base de 473 images divisées en 4 catégories, chaque objet est représenté par un point en 2 dimensions déterminées par les deux premiers vecteurs propres de la représentation par MDS.

MDS sur distances non métriques

La matrice B est définie positive ou semi-positive si et seulement si la matrice de distances D est euclidienne. Alors pour des distances non euclidiennes, cette matrice est non définie. En effet, B aura des valeurs propres négatives, et on ne pourra pas construire une représentation des données dans l'espace euclidien car elle dépend des racines carrées des valeurs propres. Mais il est possible de corriger la matrice de distances D de telle façon que sa

matrice *B* correspondante devient définie semi-positive. Une description très détaillée des ces possibilités existe dans [Borg&Groenen 1997]. Ces approches sont utiles quand les valeurs propres négatives sont relativement petites c'est-à-dire les distances sont presque euclidiennes. Dans ce cas les valeurs négatives peuvent être interprétées comme du bruit et peuvent être négligées.

Deux approches sont possibles pour traiter ce problème dans un espace euclidien.

La première approche consiste à prendre en compte seules les *p* valeurs propres positives. On obtient alors une configuration de *p* dimensions de $X = Q_p \Lambda_p^{\frac{1}{2}}$, $p < k$, pour lequel les distances se rapprochent des distances d'origine. Étant donné que les distances sont positives, la plus grande des valeurs propres négatives en valeur absolue est plus petite que la plus grande valeur propre positive. Aussi la somme des valeurs propres positives est plus grande que la somme en valeurs absolues des valeurs propres négatives. Pekalska et Duin [Pekalska&Duin 2005] font l'hypothèse que les distances mesurées peuvent être bruitées et donc, qu'elles peuvent ne pas être tout à fait euclidiennes. Il en résulte alors la présence de valeurs propres négatives pour *B*, faibles en valeur absolue. Ainsi en les supprimant, on réduirait l'effet du bruit.

La deuxième approche suivant [Cox&Cox 2001] est qu'il existe une constante positive $c \geq 2|\lambda|$, où λ est la plus petite valeur propre négative de *B*, telle qu'une nouvelle matrice de distances euclidiennes au carré pourrait être créée de D^2 en ajoutant *c* aux éléments non diagonaux :

$$D_n^2 = D^2 + c\left(\mathbf{1}\mathbf{1}^T - I\right). \tag{1.26}$$

Alors *X* peut être de nouveau exprimé dans un espace euclidien. En pratique, les vecteurs propres restent les mêmes et la valeur $c/2$ est ajouté aux valeurs propres différentes de 0. La nouvelle matrice des valeurs propres est $\Lambda_k + \frac{c}{2} I$. Cela équivaut à la régularisation de la matrice de covariance de notre configuration *X*, c'est-à-dire : $Cov(X) = \frac{1}{n-1}\left(\Lambda_k + \frac{c}{2} I\right)$.

La question est encore ouverte sur l'utilité pour l'apprentissage de transformer les données de dissimilitudes non euclidiennes *D* en une représentation euclidienne, soit en négligeant les valeurs propres négatives, ou en augmentant les distances *D* par l'ajout d'une constante.

Dans les cas où un espace euclidien n'est pas « suffisamment grand » pour incorporer les données de dissimilitudes, Pekalska et Duin proposent d'incorporer *D* dans un espace pseudo-euclidien [Pekalska&Duin 2005]. Un espace pseudo-euclidien est une décomposition directe orthogonale de l'espace euclidien en deux espaces vectoriels $\Re^p$ et $\Re^q$. Pour ces deux espaces le produit scalaire est défini positif dans $\Re^p$ et défini négatif dans $\Re^q$. Le raisonnement est comme pour le MDS métrique sauf qu'on utilise les notions de produit scalaire dans un espace pseudo-euclidien. Les détails pour la solution de ce problème peuvent être trouvés dans [Pekalska&Duin 2005].

Ce raisonnement est très utile pour des distances basées sur les opérations de minimum et de maximum comme la distance de Hausdorff. Ce type de distances donne un grand nombre des valeurs propres négatives significatives.

En conclusion la difficulté de l'approche globale MDS réside principalement dans :

- *Le choix de l'algorithme de positionnement*. Le choix d'une technique non linéaire peut être plus adéquat si les données ont des liens non linéaires entre elles. La difficulté est liée à la qualité de l'optimisation. Une technique linéaire sera plus simple à maîtriser, mais apportera peut-être plus de distorsions dans la représentation euclidienne.
- *La validité de l'existence de la représentation euclidienne*. Si les données sont fortement des données de dissimilitude (inégalité triangulaire fortement non respectée), la représentation par distance euclidienne sera très éloignée de la « vérité » ; idem pour le non respect de la symétrie.
- *Le choix de la dimension de l'espace euclidien en sortie*. Ce choix se fait généralement par un critère global qui ne tient pas compte du pouvoir discriminant des dimensions.
- *Le coût de calcul*. Ce coût peut-être prohibitif quand le nombre d'objet *n* est important. Le nombre de calcul est $O(n^4)$ pour *n* objets.

Classement dans l'espace euclidien et pseudo-euclidien défini par MDS

Après avoir défini l'espace des représentations approprié afin de classer les données, on peut ensuite choisir parmi plusieurs classifieurs pour réaliser l'analyse discriminante : les méthodes linéaires comme le classificateur linéaire, le classificateur de Fisher, les machines à vecteurs de support, les méthodes non linéaires comme le KNN, le classifieur quadratique, …[Pekalska&Duin 2005]. Si l'espace de représentation choisi est l'espace pseudo-euclidien, il faut calculer les distances dans cet espace $\Re^{(p,q)}$. Alors, on calcule la distance euclidienne dans l'espace « positif » $\Re^p$ et soustrait la distance euclidienne calculée dans l'espace « négatif » $\Re^q$. La distance calculée dans l'espace positif est surestimée alors le but de l'espace « négatif » est de la corriger. Dans ce cas les classifieurs conventionnels doivent être redéfinis car les densités des classes peuvent ne pas être correctement définies dans cet espace. Dans [Pekalska&Duin 2010 ; Pekalska&Duin 2008 ; Duin et al. 2008] Pekalska et Duin redéfinissent dans l'espace pseudo-euclidien les classifieurs de la plus proche moyenne (Nearest Mean Classifier), le classificateur de Ficher, le classificateur quadratique et les machines à vecteurs de support. Dans ce dernier cas, concernant les SVM, la convergence et l'unicité ne sont pas garanties [Haasdonk 2005].

Dans le Chapitre 4 les résultats avec ces classifieurs seront comparés avec ceux du classifieur « Coefficient de forme ».

1.5. Classifieurs directement dans l'espace des dissimilitudes

1.5.1. La règle du plus proche et des K plus proches voisins – technique de rang

L'approche la plus simple basée sur les mesures de dissimilitudes est la règle du plus proche voisin (1NN) et son extension de K plus proches voisins (KNN) [Duda et al. 2000 ;

Hastie et al. 2009 ; Fukunaga 2001]. Dans sa forme initiale, on attribue chaque nouvel objet à la classe de son plus proche voisin. Soient $X=\{x_1,\ldots,x_N\}$ de n objets et $\Omega=\{\omega_1,\ldots,\omega_c\}$ l'ensemble des classes possibles. Soit x' un objet appartenant à X, et voisin le plus proche d'un point de test x. Alors suivant la règle du plus proche voisin, on affecte le point x à la même classe que celle du point x'. Cette règle est une procédure non paramétrique. Elle permet de partitionner l'espace des caractéristiques selon une partition de Voronoi, en cellules contenant les plus proches points d'un point d'apprentissage donné x'. Tous les points dans une telle cellule appartiennent à la même classe que le point d'apprentissage (fig. 1.11).

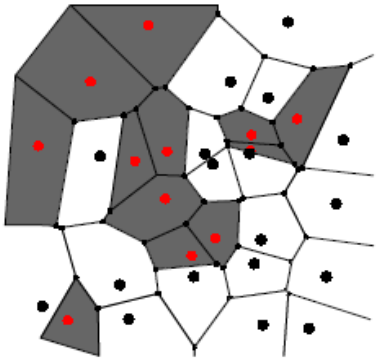


Fig. 1.11 L'algorithme du plus proche voisin dans un espace 2D qui conduit à une partition de l'espace en polygones de Voronoi, chacun étant étiqueté suivant le point d'apprentissage qu'il contient [Duda et al. 2000].

Dans le cas des K plus proches voisins, la règle est fondée sur le vote majoritaire. Un objet inconnu devient membre de la classe la plus fréquente parmi ses K plus proches voisins (fig. 1.12). Habituellement, K est supposé être impair pour éviter les conflits. Lorsque K est fixé aucune étape d'apprentissage n'est nécessaire.

La règle du KNN nécessite seulement un nombre entier K, un ensemble étiqueté d'apprentissage et une mesure de distance métrique.

La règle des K plus proches voisins peut être interprétée comme une règle de recherche de maximum de la probabilité a posteriori, estimée ainsi :

$$P(\omega_i \mid x) = \frac{K_i}{K}, \tag{1.27}$$

avec K_i, le nombre de voisins appartenant à la classe ω_i parmi les K plus proches voisins. La valeur de K est un compromis : une valeur élevée sera adaptée aux densités faibles et une valeur faible, aux densités élevées.

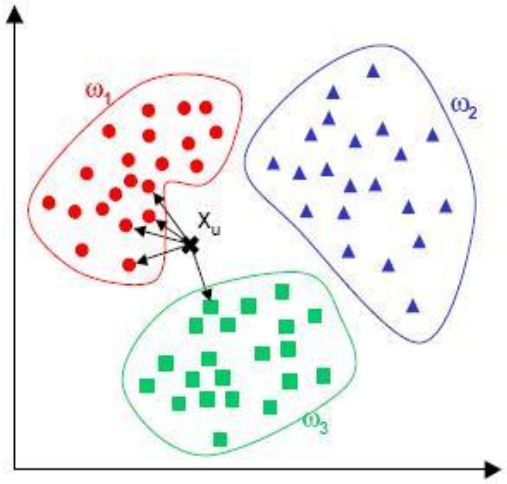


Fig. 1.12 Exemple de classement par la règle des K plus proches voisins (K = 5 voisins dans ce cas) d'un objet inconnu $\mathbf{x_u}$ *parmi 3 classes* ω_1, ω_2, ω_3. *La classe est* ω_1 *majoritaire, l'objet* $\mathbf{x_u}$ *est attribué à cette classe.*

En conclusion le classifieur KNN est très attractif, car il est simple - aucune connaissance préalable sur la distribution des données n'est nécessaire. Il utilise l'information de distance uniquement par les rangs. Mais puisqu'il s'agit d'une méthode locale, elle est sensible au bruit et plus globalement à la fluctuation des échantillons d'apprentissage et de test. De plus cette méthode est mal adaptée aux mesures qui ne respectent pas l'inégalité triangulaire.

1.5.2. Les machines à vecteurs de support à deux classes

Les machines à vecteurs de support (SVM) sont introduites au début des années 90. Elles sont issues de la théorie de l'apprentissage statistique [Abe 2005 ; Cristianini&Shawe-Taylor 2000]. Aujourd'hui, nous pouvons dire sans exagérer que ces classifieurs ont supplanté les réseaux de neurones et autres techniques d'apprentissage. En effet, elles sont largement répandues en apprentissage statistique pour le classement et la régression et ont eu beaucoup de succès dans quasiment tous les domaines où elles ont été appliquées. De nombreux codes sont disponibles en ligne, ce qui facilite encore plus leur utilisation. Les machines à vecteurs de support exploitent la théorie des bornes de Vapnik et Chervonenkis pour aborder d'une façon nouvelle, la question du dilemme biais-variance en apprentissage [Vapnik 1995]. Le compromis entre la capacité d'apprentissage et la capacité de généralisation pour ces machines est réalisé en minimisant l'erreur empirique et dans le même temps, en essayant de maximiser une marge géométrique.

Les machines à vecteurs de support sont des méthodes de noyaux. Dans une certaine mesure, la notion de dissimilitude est liée à l'utilisation de noyaux. La différence principale est que les noyaux sont définis dans des espaces vectoriels pour remplir les conditions de Mercer (la fonction du noyau doit être symétrique avec des valeurs réels et définie positive ou définie positive conditionnelle) [Cristianini&Shawe-Taylor 2000]. Les valeurs des noyaux peuvent être interprétées à partir des produits scalaires entre les vecteurs de caractéristiques et sont, à ce titre, des similitudes.

A notre connaissance, principalement trois approches ont été proposées pour l'utilisation des données de distance dans les SVMs. Une approche consiste à représenter chaque objet de l'ensemble d'apprentissage en tant que vecteur de ses distances à tous les objets de l'ensemble des prototypes et on applique les SVM standard sur ces données, [Pekalska et al. 2002]. La seconde méthode consiste en l'incorporation de données de distance dans un espace vectoriel euclidien ou pseudo - euclidien, la régularisation de celui-ci et l'interprétation de classement dans cet espace avec les SVM linéaires [Pekalska&Duin 2008 ; Duin et al. 2008]. Ces approches ont l'inconvénient de perdre la parcimonie dans le sens où tous les objets de l'ensemble d'apprentissage doivent être retenus pour le classement. Cela les rend peu pratiques pour les données à grande échelle.

La troisième approche est inspirée par l'utilisation de noyaux dans les machines à vecteurs de support et l'utilisation des dissimilitudes, Pekalska, Duin et Haasdonk ont commencé à expérimenter la construction d'autres noyaux en utilisant les mesures de distance spécifiques aux problèmes de classement [Haasdonk&Bahlmann 2004 ; Haasdonk&Pekalska 2010 ; Haasdonk&Pekalska 2009 ; Haasonk&Pekalska 2008]. Leur objectif était d'élaborer des procédures pour tout type de matrices de dissimilitudes, générées dans les applications de reconnaissance de formes.

Dans le paragraphe ci-dessous sera expliqué l'algorithme d'optimisation des SVM (cas séparable et non séparable) et à la fin on introduira les noyaux de distance de Haasdonk.

Cas des classes séparables

Le cas des deux classes linéairement séparables est le plus facile dans la théorie des machines à vecteurs de support [Burges 1998 ; Han&Kamber 2006 ; Alpayadin 2004]. Le but est de prévoir la classe y d'un vecteur $\mathbf{x}$ en p dimensions. Nous supposons que les données d'apprentissage sont des couples $\{\mathbf{x}_i, y_i\}_{1\le i\le l}$, $y_i \in \{-1,1\}$, $\mathbf{x}_i \in \Re^p$, et l est la taille de l'ensemble d'apprentissage. L'appartenance d'une observation $\mathbf{x}_i$ à une classe ou à une autre est codée par la valeur -1 ou 1 de son étiquette y_i. Supposons que nous ayons un hyperplan séparateur séparant les observations négatives des positives. Les points $\mathbf{x}$ qui se trouvent sur l'hyperplan satisfont l'équation suivante : $\boldsymbol{\beta}^T\mathbf{x} + \beta_0 = 0$, où $\boldsymbol{\beta}$ est la normale de l'hyperplan, $|\beta_0|/\|\boldsymbol{\beta}\|$ est la distance perpendiculaire de l'hyperplan à l'origine et $\|\boldsymbol{\beta}\|$ est la norme euclidienne de $\boldsymbol{\beta}$. Soit d_+ (d_-) la plus courte distance de l'hyperplan séparateur à sa plus proche observation positive (négative). La «marge» est alors la valeur $d_+ + d_-$. Pour le cas linéairement séparable, l'algorithme des machines à vecteurs de support recherche l'hyperplan séparateur avec la plus grande marge. C'est un problème d'optimisation convexe garantissant la convergence vers une solution optimale. Cela peut être formulé comme suit.

Puisque le problème est linéairement séparable, toutes les données d'apprentissage satisfont les contraintes suivantes:

$$\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0 \ge +1 \text{ pour } y_i = +1 \tag{1.28}$$

$$\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0 \le -1 \text{ pour } y_i = -1. \tag{1.29}$$

Ces deux inégalités combinées donnent :

$$y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i+\beta_0\right)-1\geq 0 \text{ pour } \forall \text{i} . \tag{1.30}$$

Considérons maintenant les points pour lesquels l'égalité (1.28) est vraie. Ces points se trouvent sur l'hyperplan: $\boldsymbol{\beta}^T\mathbf{x} + \beta_0 = 1$. De même, les points pour lesquels l'égalité (1.29) est vraie, se trouvent sur l'hyperplan: $\boldsymbol{\beta}^T\mathbf{x} + \beta_0 = -1$. Les deux hyperplans sont parallèles et aucun point de l'ensemble ne se situe entre eux. La largeur de marge est alors $2/\|\boldsymbol{\beta}\|$. Ainsi, nous pouvons trouver la paire d'hyperplans qui donne la marge maximale en minimisant $\|\boldsymbol{\beta}\|^2$, sous les contraintes (1.30). La solution d'un cas typique à deux dimensions est illustrée sur la figure 1.13. Les points pour lesquels l'égalité (1.30) est vraie, et dont le retrait changera la solution trouvée, sont appelés vecteurs de support. Ces vecteurs sont indiqués sur la figure 1.13, par des cercles supplémentaires.

S'agissant d'un problème d'optimisation avec contraintes, il est résolu par la technique des multiplicateurs de Lagrange. Les contraintes (1.30) sont remplacées par des contraintes sur les multiplicateurs de Lagrange, qui sont beaucoup plus faciles à manipuler. De plus dans cette reformulation du problème, les données d'apprentissage n'apparaissent que sous la forme de produits scalaires entre des vecteurs. C'est une propriété essentielle qui permettra de généraliser la procédure au cas non linéaire des machines à vecteurs de support.

Ainsi, on introduit les multiplicateurs de Lagrange positifs α_i, $i = 1, \ldots, l$, un pour chacune des contraintes d'inégalité (1.30). Rappelons que la règle est que pour les contraintes de la forme $c_i > 0$, les équations des contraintes sont multipliées par les multiplicateurs de Lagrange positifs et soustraites de la fonction objective à minimiser pour former le Lagrangien. Pour les contraintes de l'égalité, les multiplicateurs de Lagrange sont sans contraintes. Cela donne le Lagrangien primaire :

$$L_P\left(\boldsymbol{\beta},\beta_0,\boldsymbol{\alpha}\right)=\frac{1}{2}\|\boldsymbol{\beta}\|^2-\sum_{\text{i=1}}^{\text{l}}\alpha_i y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i+\beta_0\right)+\sum_{i=1}^{l}\alpha_i \tag{1.31}$$

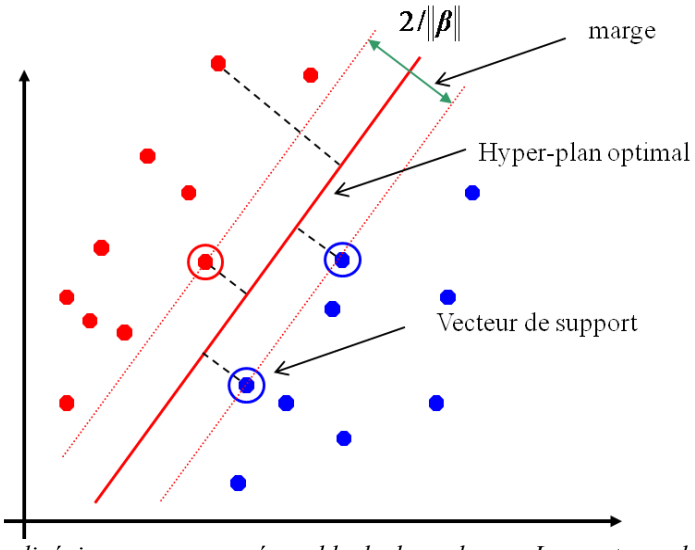


Fig. 1.13 Le plan séparateur linéaire pour un cas séparable de deux classes. Les vecteurs de support sont encerclés.

Nous devons maintenant optimiser* L_P par rapport à $\boldsymbol{\beta}$, β_0 et aux multiplicateurs α_i. C'est un problème de programmation quadratique convexe, puisque la fonction objective elle-même est convexe, et les points qui satisfont les contraintes forment également un ensemble convexe. A la convergence de la solution, les dérivées de L_P sont nulles :

$$\frac{\partial}{\partial \boldsymbol{\beta}} L_P = 0 \Leftrightarrow \boldsymbol{\beta} = \sum_i^{N_S} \alpha_i y_i \mathbf{x}_i \tag{1.32}$$

$$\frac{\partial}{\partial \beta_0} L_P = 0 \Leftrightarrow \sum_i \alpha_i y_i = 0 . \tag{1.33}$$

Le N_S est le nombre des vecteurs supports. On introduit les conditions de Karush-Kuhn-Tucker (KKT) qui reprennent celles-ci-dessus. Elles jouent un rôle important dans la théorie et la pratique des problèmes d'optimisation avec contraintes. Pour le problème primaire, les conditions KKT sont les suivantes :

$$\frac{\partial}{\partial \beta_v} L_P = \beta_v - \sum_i \alpha_i y_i x_{iv} = 0 \quad v = 1, ..., p \tag{1.34}$$

$$\frac{\partial}{\partial \beta_0} L_P = -\sum_i \alpha_i y_i = 0 \tag{1.35}$$

$$y_i \left(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0 \right) - 1 \geq 0 \text{ pour } \forall i \tag{1.36}$$

$$\alpha_i \geq 0 \quad \forall i \tag{1.37}$$

$$\alpha_i \left(y_i \left(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0 \right) - 1 \right) \geq 0 \text{ pour } \forall i . \tag{1.38}$$

Le p indique les dimensions des données. Cela donne des équations supplémentaires qui permettent une reformulation de ce problème d'optimisation, en utilisant les équations (1.34) et (1.35) dans l'équation « primaire » (1.31). Ainsi on obtient la nouvelle formulation « duale » ne dépendant que des multiplicateurs α_i sous les contraintes $\alpha_i \geq 0$, pour toutes les observations :

$$L_D(\boldsymbol{\alpha}) = -\sum_{i=1}^{l} \alpha_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i ; \mathbf{x}_j \rangle . \tag{1.39}$$

Cette équation fait apparaître les termes de la matrice Hessienne ($\langle \mathbf{x}_i ; \mathbf{x}_j \rangle$) représentant la matrice des produits internes et le problème d'optimisation quadratique. Alors l'apprentissage des machines à vecteurs de support équivaut à une minimisation quadratique de L_D par rapport à α_i et respectant les conditions de Karush-Kuhn-Tucker. Notons que l'application des conditions KKT aboutit naturellement à la fonction L_D à maximiser par rapport aux multiplicateurs de Lagrange. L'équation (1.39) est équivalente pour la minimisation de la formulation opposée. On a choisi cette formulation conformément à [Osuna et al. 1997] et [Joachims 1999] puisqu'on a implémenté le processus d'optimisation des SVM dans l'environnement SVM^{Light} (http://svmlight.joachims.org, réf. paragraphe 2.4.3).

Dans la solution, tous les points avec $\alpha_i > 0$ sont nommés « vecteurs de support », et ils se trouvent soit sur l'hyperplan $\boldsymbol{\beta}^T \mathbf{x} + \beta_0 = 1$ soit sur $\boldsymbol{\beta}^T \mathbf{x} + \beta_0 = -1$. Tous les autres points ont des

*C'est une minimisation par rapport à $\boldsymbol{\beta}$ et β_0 et une maximisation par rapport les multiplicateurs de Lagrange α_i.

valeurs α_i = 0. Les vecteurs de support sont ainsi les éléments essentiels de l'ensemble d'apprentissage. Ils sont les plus proches de la frontière de décision et si tous les points de l'ensemble d'apprentissage sont supprimés hormis ces points là, et l'apprentissage répété, on trouvera le même plan séparateur.

Le vecteur normal $\boldsymbol{\beta}$ à l'hyperplan séparateur est déterminé par la combinaison linéaire des vecteurs de support (1.32 et 1.34). Il reste à déterminer le biais β_0. Il est obtenu considérant que tous les vecteurs de support appartiennent aux hyperplans positionnés sur les marges. On a donc autant d'équations qu'il y a de vecteurs de support : $y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0\right) - 1 = 0$, pour les indices i des points vecteurs de support. La valeur de β_0 est la moyenne des solutions obtenues.

Une fois que nous avons formé le classifieur SVM, on détermine simplement de quel côté de l'hyperplan de décision ($\boldsymbol{\beta}^T\mathbf{x} + \beta_0 = 0$) se trouve un point de test $\mathbf{x}$. L'étiquette de la classe correspondante est attribuée : $y = sign(\boldsymbol{\beta}^T\mathbf{x} + \beta_0)$.

Cas des classes non séparables

L'algorithme présenté ci-dessus pour des données séparables ne trouvera de solution quand il est appliqué à des données non séparables. Alors pour l'étendre aux cas non séparable, il est nécessaire de relâcher un peu les contraintes (1.28) et (1.29) dans le but d'autoriser quelques erreurs de classement mais seulement si c'est nécessaire. On introduit alors un coût supplémentaire. Pour se faire des variables d'écart positives ξ_i, $i = 1, \ldots, l$ sont introduites dans les contraintes (1.28) et (1.29). Elles codent la distance des points à la marge :

$$\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0 \geq +1 - \xi_i \text{ pour } y_i = +1 \tag{1.40}$$

$$\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0 \leq -1 + \xi_i \text{ pour } y_i = -1 \tag{1.41}$$

$$\xi_i \geq 0 \ \forall i\,. \tag{1.42}$$

Pour un point mal classé, il y aura une erreur et la variable ξ_i correspondante sera supérieure à un. Cette situation est montrée à la figure 1.14. L'égalité à un correspond à un point sur la frontière. Ainsi $\sum_i \xi_i$ est une borne supérieure des erreurs d'apprentissage. Une façon naturelle d'attribuer un coût supplémentaire pour les erreurs consiste à modifier la fonction objective ainsi :

$$\text{Minimiser}_{\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}} \ \ \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{l}\xi_i \tag{1.43}$$

$$\text{avec } y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0\right) \geq 1 - \xi_i,\ i = 1,\ldots,l \tag{1.44}$$

$$\xi_i \geq 0 \ \forall i \tag{1.45}$$

Autrement dit, on cherche à maximiser la marge en s'autorisant pour chaque contrainte une erreur positive ξ_i, la plus petite possible. Le paramètre supplémentaire C qui apparaît ici est une constante positive fixée à l'avance qui permet de contrôler l'importance de l'erreur que l'on

s'autorise par rapport à la taille de la marge. Ce paramètre peut être fixé par validation croisée. Plus C est important, plus les erreurs seront pénalisées.

La seule différence par rapport au problème séparable est la majoration des α_i par C. On montre que si l'ensemble d'apprentissage est linéairement séparable et quand C est suffisamment grand, les deux problèmes (1.31) et (1.43) deviennent équivalents.

Le Lagrangien primaire a la forme suivante :

$$L_P(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_i \xi_i - \sum_{i=1}^{l} \alpha_i \{y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) - 1 + \xi_i\} - \sum_i \mu_i \xi_i \,, \tag{1.46}$$

avec μ_i les multiplicateurs de Lagrange, introduits afin de considérer la positivité de ξ_i.

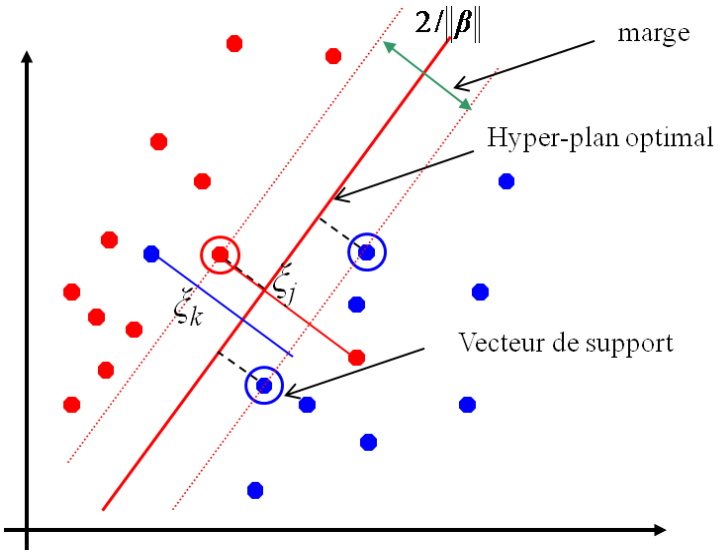


Fig. 1.14 Le plan séparateur linéaire pour un cas non séparable de deux classes.

De même, on peut déduire les conditions de KKT :

$$\frac{\partial}{\partial \beta_v} L_P = \beta_v - \sum_i \alpha_i y_i x_{iv} = 0 \quad v = 1, \ldots, p \tag{1.47}$$

$$\frac{\partial}{\partial \beta_0} L_P = -\sum_i \alpha_i y_i = 0 \tag{1.48}$$

$$\frac{\partial}{\partial \xi_i} L_P = C - \alpha_i - \mu_i = 0 \tag{1.49}$$

$$y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) - 1 + \xi_i \geq 0 \tag{1.50}$$

$$\alpha_i \geq 0 \tag{1.51}$$

$$\xi_i \geq 0 \tag{1.52}$$

$$\mu_i \geq 0 \tag{1.53}$$

$$\alpha_i \left\{ y_i \left(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0 \right) - 1 + \xi_i \right\} = 0 \tag{1.54}$$

$$\mu_i \xi_i = 0 \tag{1.55}$$

En suivant la même démarche de reformulation, nous aboutissons à la forme duale :

$$\text{Minimiser } L_D(\boldsymbol{\alpha}) = -\sum_i \alpha_i + \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i ; \mathbf{x}_j \rangle \tag{1.56}$$

$$\text{avec } 0 \le \alpha_i \le C \tag{1.57}$$

$$\sum_i \alpha_i y_i = 0 . \tag{1.58}$$

La minimisation quadratique fournit les multiplicateurs Lagrangiens α_i. Et comme dans le cas séparable, les conditions de KKT permettent de calculer l'hyperplan séparateur par son vecteur normal ($\boldsymbol{\beta}$) et son biais (β_0).

1.5.3. Les machines à vecteurs de support à moindres carrés à deux classes – Least Square Support Vector Machines (LS-SVM)

Comme montré précédemment, l'optimisation des SVM est une optimisation quadratique convexe qui peut être lourde en temps de calcul et à convergence difficile. Afin de résoudre ces problèmes, d'une part des solutions algorithmiques ont été apportées (comme par exemple par [Osuna et al. 1997 ; Joachims 1999]), d'autre part, des formulations alternatives sont proposées. On présente ici la simplification des SVM telle que proposée par [Suykens et al. 2002 ; Suykens et al. 1999]. Ils proposent une modification de la fonctionnelle à minimiser aboutissant à trouver la solution par la résolution d'un système d'équations linéaires (« Least Square SVM » – LS-SVM). A la place de la somme linéaire des variables d'écart dans la fonction objective (1.41), Suykens introduit la somme au carré de ces variables et transforme les contraintes d'inégalité en égalité. Les motivations du développement de l'approche LS-SVM sont les suivantes [Suykens et al. 2003] :

- La formulation des méthodes SVM en termes de moindres carrés des fonctions de coût et avec les contraintes d'égalité au lieu contraintes d'inégalité conduit à la résolution des systèmes linéaires. Ces types de problèmes sont mieux compris d'un point de vue théorique, algorithmique et numérique que de résoudre un problème de programmation quadratique.
- Par l'incorporation des deux formulations des SVM (le Lagrangien primaire et sa forme duale) et les contraintes d'égalité, les LS-SVM permettent des extensions vers des techniques d'un large contexte interdisciplinaire telles que les réseaux de neurones récurrents [Suykens&Vandewalle 2000], contrôle optimal non linéaire [Suykens et al. 2001], « kernel PCA », « Gaussian Progress Regression », analyse discriminante de Fisher, « kernel Canonical Correlation Analysis » et « kernel Partial Least Squares » [Suykens et al. 2003].

On introduit la formulation suivante :

$$\text{Minimiser}_{\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}} \ \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{l}\xi_i^{\,2} \tag{1.59}$$

$$\text{avec } y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0\right) = 1-\xi_i,\ i = 1,...,l \tag{1.60}$$

En introduisant les multiplicateurs lagrangiens $\alpha_i \geq 0$, on obtient :

$$L_P(\boldsymbol{\beta},\beta_0,\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_i \xi_i^{\,2} - \sum_{i=1}^{l}\alpha_i\left\{y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0\right) - 1 + \xi_i\right\} \tag{1.61}$$

Pour la solution optimale, les conditions KKT suivantes sont remplies:

$$\frac{\partial}{\partial\boldsymbol{\beta}}L_P = 0 \rightarrow \boldsymbol{\beta} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{1.62}$$

$$\frac{\partial}{\partial\beta_0}L_P = 0 \rightarrow \sum_i \alpha_i y_i = 0 \tag{1.63}$$

$$\frac{\partial}{\partial\xi_i}L_P = 0 \rightarrow \alpha_i = C\xi_i \tag{1.64}$$

$$\frac{\partial}{\partial\alpha_i}L_P = 0 \rightarrow y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0\right) - 1 + \xi_i = 0\,. \tag{1.65}$$

Ces équations peuvent être réécrites sous une forme d'un système linéaire :

$$\begin{bmatrix} I_{p\times p} & \mathbf{0}_{p\times 1} & \mathbf{0}_{p\times l} & -Z^T \\ \mathbf{0}_{1\times p} & 0 & \mathbf{0}_{1\times l} & -Y^T \\ \mathbf{0}_{l\times p} & \mathbf{0}_{l\times 1} & CI_{l\times l} & -I_{l\times l} \\ Z & Y & I_{l\times l} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\beta} \\ \beta_0 \\ \boldsymbol{\xi} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1}_{l\times 1} \end{bmatrix}, \tag{1.66}$$

où $Z = [\,\mathbf{x}_1^T y_1\,,\ \ldots,\ \mathbf{x}_l^T y_l\,]^T$, $Y = [y_1,\ \ldots,\ y_l]^T$, $\mathbf{1}_{l\times 1} = [1,\ldots,1]^T$, $\boldsymbol{\xi} = [\xi_1,\ \ldots,\ \xi_l]^T$, $\boldsymbol{\alpha} = [\alpha_1,\ \ldots,\ \alpha_l]^T$;

La solution de ce système est alors :

$$\begin{bmatrix} 0 & -Y^T \\ Y & ZZ^T - C^{-1}I_{l\times l} \end{bmatrix}\begin{bmatrix} \beta_0 \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1}_{l\times 1} \end{bmatrix}. \tag{1.67}$$

Ainsi par ces transformations du problème d'optimisation, la solution est obtenue par la résolution d'un ensemble d'équations linéaires (1.67) au lieu d'un problème de programmation quadratique. Le nombre d'équations dans ce système est égal au nombre de point d'apprentissage plus un. De plus par cette formulation, la faible densité de $\boldsymbol{\alpha}$ n'est pas garantie. Ainsi, pour l'algorithme LS-SVM, toutes les données d'apprentissage deviennent des vecteurs de support.

Suivant ces équations on constate que les Lagrangiens sont soit positifs soit négatifs mais il n'y a pas des Lagrangiens égaux à 0. Tous les points de l'ensemble d'apprentissage sont des vecteurs de support avec des valeurs des Lagrangiens plus ou moins grandes. La distribution des points avec différentes valeurs de Lagrangiens est repartie partout dans l'espace des caractéristiques, contrairement au SVM standard, autour de la frontière de décision – l'hyperplan séparateur optimal. Cette méthode est plus facile de calculer et plus rapide avec le noyau linéaire qu'avec des autres types de noyau. La méthode est sensible aux données bruitées et aux points atypiques [Abe 2005]. Plusieurs techniques sont proposées afin de résoudre ses problèmes [Jiao et al. 2007] et des vastes comparaisons empiriques [Van Gestel et al. 2004] montrent que les LS-SVM obtiennent de bonnes performances sur des différents problèmes de classement.

1.5.4. Les machines à vecteurs de support pour le classement non linéaire – noyaux de distance de Bernard Haasdonk

Les machines à vecteurs de support présentées ci-dessus ne concernent que les modèles linéaires. Pour résoudre des problèmes de séparation non linéaire, comme cela est illustré très simplement à la figure 1.15, la proposition est de transformer le problème dans un espace à plus haute dimension [Herbrich 2002 ; Schlökopf 2000 ; Schlökopf et al. 2000].

En effet, si on considère la résolution des problèmes (1.39) et (1.56), seuls les produits scalaires $\langle \mathbf{x}_i ; \mathbf{x}_j \rangle$ sont nécessaires. Les SVM peuvent être étendues pour traiter le cas non-linéaire. La ruse qui fait vraiment la force des SVM repose l'utilisation des fonctions « noyau » *k(.)* appliquées aux produits scalaires $\langle \mathbf{x}_i ; \mathbf{x}_j \rangle$, $k(\langle \mathbf{x}_i ; \mathbf{x}_j \rangle)$. Cela revient à plonger l'espace des observations $\mathbf{x}_i$, dans un espace de Hilbert $\mathcal{T}$ de dimension plus élevé (voire infinie) que la dimension initiale, à l'aide d'une fonction non-linéaire $\varphi : \Re^p \rightarrow \mathcal{T}$ [Boser et al. 1992]. L'espace $\mathcal{T}$ ainsi obtenu est appelé « espace transformé ». L'hyperplan séparateur obtenu dans l'espace $\mathcal{T}$ est appelé hyperplan optimal généralisé.

Tout l'intérêt réside dans le fait que les nouvelles caractéristiques ne sont pas à calculer explicitement, puisque l'on a :

$$k(\mathbf{x}_i ; \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i) ; \varphi(\mathbf{x}_j) \rangle \tag{1.68}$$

Le noyau peut être toute fonction symétrique définie positive. Tout ce qu'il nous reste à faire c'est de résoudre le problème (1.39) ou (1.56) dans l'espace $\mathcal{T}$, en remplaçant $\langle \mathbf{x}_i ; \mathbf{x}_j \rangle$ par $k(\langle \mathbf{x}_i ; \mathbf{x}_j \rangle)$.

Dans cette thèse, on s'intéressera aux travaux de Bernard Haasdonk [Haasdonk&Bahlmann 2004 ; Haasdonk&Pekalska 2008 ; Haasdonk&Pekalska 2010] où sont représentées quelques fonctions de noyaux basées sur des distances.

Suivant Haasdonk et Pekalska, la classe des noyaux admissibles est souvent considérée à tort comme limitée en raison de l'obligation que la fonction « noyau » soit définie positive. Dans la pratique cependant, plusieurs mesures de similitudes non définies positives existent. Naturellement, des similitudes/dissimilitudes non définies positives résultent des dissimilitudes non-euclidiennes ou non-métriques, telles que la distance modifiée de Hausdorff, et la divergence de Kullback-Leibler entre des distributions de probabilité.

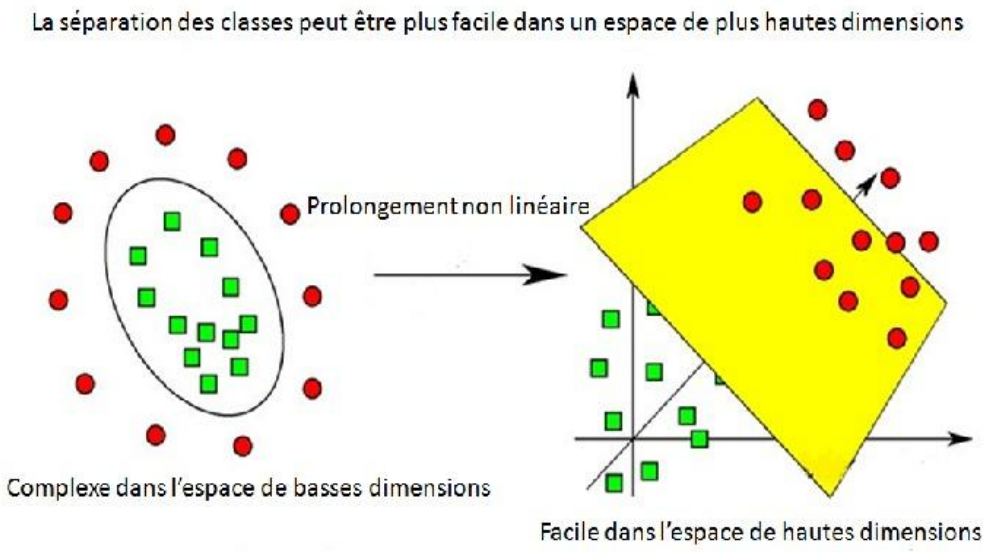


Fig. 1.15 Exemple d'un prolongement non linéaire de $\Re^2$ *à* $\Re^3$

Par conséquent, il y a une nécessité pratique de créer des noyaux non définis positifs pour gérer ces mesures correctement.

En utilisant la notation de [Haasdonk&Bahlmann 2004] et le produit scalaire représenté par (1.16), on a les noyaux de distances suivants :

$$\begin{aligned} k^{lin}(\mathbf{x}_i,\mathbf{x}_j) &= \langle \mathbf{x}_i;\mathbf{x}_j \rangle \\ k^{pol}(\mathbf{x}_i,\mathbf{x}_j) &= (1-\gamma\langle \mathbf{x}_i;\mathbf{x}_j \rangle)^p, \mathrm{p} \in \mathrm{N} \\ k^{nd}(\mathbf{x}_i,\mathbf{x}_j) &= -d(\mathbf{x}_i,\mathbf{x}_j)^\beta, \beta \in [0,2] \\ k^{rfb}(\mathbf{x}_i,\mathbf{x}_j) &= \mathrm{e}^{-\gamma d(\mathbf{x}_i,\mathbf{x}_j)^2}, \gamma \in \Re^+ \end{aligned} \tag{1.69}$$

Si la matrice de dissimilitudes est non négative, symétrique et à diagonale nulle alors le noyau polynomial k^{pol} et la fonction radiale de Gauss k^{rbf} sont des fonctions définies positives. Par contre le noyau de distance négative k^{nd} est une fonction conditionnellement définie positive pour $\beta \in [0,2]$. Si les distances sont non métriques, ces noyaux perdent ces propriétés.

Haasdonk apporte les arguments suivants [Haasdonk 2005] :

- Les SVM avec des noyaux non définis positifs peuvent être interprétés comme des classifieurs avec un hyperplan optimal séparateur dans des espaces définis par ces noyaux.
- La convergence des SVM vers une solution optimale ne peut être garantie.
- Même avec des noyaux non définis positifs des solutions uniques sont possibles.

Dans [Haasdonk&Pekalska 2008 ; Haasdonk&Pekalska 2010] sont définies les fonctions de noyaux non définis positifs de Fisher et de la distance de Mahalanobis. La partie théorique de ces fonctions dépasse le cadre de cette thèse. Nous nous sommes intéressés ici à l'existence de ces

nouveaux et aux résultats obtenus par ces auteurs sur les mêmes bases de données réelles que l'on a utilisé pour confronter notre méthode.

1.5.5. Les machines à vecteurs de support multi-classes

Les machines à vecteurs de support sont formulées pour des problèmes de classement à deux classes. Mais parce qu'elles emploient directement les fonctions de décision, une extension à des problèmes multi-classes n'est pas si simple. Selon [Abe 2005] on peut formuler plusieurs stratégies pour combiner des SVM afin de traiter les problèmes multi-classes. Mais parmi celles-ci, les deux plus intéressantes sont :

- un contre tous,
- deux à deux.

La stratégie « un-contre-tous » transforme un problème de n classes en n problèmes à deux classes. La stratégie « deux à deux » convertit un problème de n classes en $n(n\text{-}1)/2$ problèmes à deux classes, qui couvrent toutes les paires de classes. Notons que pour ces deux méthodes, des régions inclassables existent car la fonction de décision est une fonction binaire.

Dans notre étude on s'intéressera plus particulièrement à la méthode « un contre tous » qui est la plus connue, la plus utilisée dans la littérature et la plus rapide à implémenter et selon [Rifkin&Klautau 2004 ; Duan&Keerthi 2005] cette procédure est aussi performante que les autres.

SVM - un contre tous

Considérons un problème de n classes. Pour une stratégie « un contre tous », nous déterminons n fonctions de décision qui séparent les classes, entre une classe ω_i et les $n\text{-}1$ classes restantes. Soit la i-nième fonction de décision du classifieur « i contre tous », avec la marge maximale qui sépare la classe i des classes restantes telle que :

$$t_i(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{x} + \beta_{0i}\,. \tag{1.70}$$

L'hyperplan $t_i(\mathbf{x}) = 0$ est l'hyperplan optimal de séparation, et si le problème de classement est séparable, les données d'apprentissage appartenant à la classe i satisfont $t_i(\mathbf{x}) \geq 1$ et ceux qui appartiennent aux classes restantes, satisfont $t_i(\mathbf{x}) \leq -1$.

Si pour un vecteur $\mathbf{x}$ d'entrée, on a :

$$t_i(\mathbf{x}) > 0\,, \tag{1.71}$$

alors l'observation est affectée à classe ω_i. Si (1.71) est satisfait pour plusieurs classifieurs « i contre tous » ou s'il n'y a aucun qui satisfait (1.71), alors $\mathbf{x}$ est inclassable. Pour éviter cela, au lieu des fonctions de décision binaires (signe), on utilise des fonctions continues pour le classement. À savoir, la donnée $\mathbf{x}$ est affectée telle que :

$$\arg\max_{i=1,..,n} t_i(\mathbf{x}). \quad (1.72)$$

1.6. Synthèse

Les dissimilitudes sont un moyen naturel de représenter des objets. Certains les considèrent comme plus fondamentales que les caractéristiques [Edelman 1999]. Les dissimilitudes ont été étudiées dans [Pekalska&Duin 2005] pour l'apprentissage supervisé et non supervisé comme une alternative à l'utilisation des caractéristiques pour la construction des représentations. Elles sont particulièrement utiles dans deux contextes : tout d'abord, en l'absence de propriétés claires qui peuvent devenir des caractéristiques et, d'autre part, lorsque les objets peuvent être comparés au niveau global comme par exemples des formes dans les images, les signaux ou de spectres.

Il existe trois principales approches pour le classement des données de dissimilitudes.

La première est le recodage des données de dissimilitudes avec le choix d'un ensemble représentatif de prototypes comme proposé par Pekalska et Duin. La construction de ces espaces vectoriels de dissimilitudes exige un choix approprié de l'ensemble de représentation. C'est un problème similaire à la sélection des caractéristiques [Pekalska&Duin 2006]. Pour le choix de l'ensemble de représentation, on peut utiliser la nature intrinsèque des dissimilitudes, par exemple les objets plus proches ou plus similaires sont susceptibles à appartenir à la même classe. Pour la construction de classifieurs dans cet espace, des dissimilitudes peuvent être utilisées de la même manière que les caractéristiques. Notons alors que l'on néglige le caractère originel de dissimilitudes, comme comparaison par paire. Avec l'approche de Pekalska, on obtient des règles de décisions globales, mais cela suppose en pratique une sélection de prototypes. Les différentes techniques de choix des prototypes sont illustrées dans [Pekalska&Duin 2002]. Si le prototype est une observation ayant toutes les caractéristiques de sa classe alors le pouvoir discriminant de l'ensemble de représentation est fort et si ce prototype est non typique pour sa classe alors le pouvoir discriminant est faible. Tous les classifieurs traditionnels peuvent y être appliqués comme ceux déjà mentionnés dans les paragraphes précédents.

La deuxième approche c'est la méthode de la projection linéaire ou non linéaire dans l'espace des caractéristiques – le Positionnement Multidimensionnel. Cet algorithme s'appuie sur un plongement linéaire ou non linéaire de la matrice de dissimilitudes. Avec la projection d'une matrice de dissimilitudes dans un espace avec une métrique donnée, la nature des dissimilitudes est préservée. Il est naturel de chercher un prolongement dans un espace euclidien car la métrique euclidienne est supposée, soit implicitement ou explicitement dans de nombreux systèmes de classement et dans ce cas-là, on peut utiliser les classifieurs traditionnels. Les distances non-euclidiennes ne peuvent être prolongées que d'une manière approximative dans un espace euclidien. Parfois ce n'est pas suffisant et on arrive à construire un espace pseudo-euclidien. Certains classifieurs peuvent être définis dans cet espace pseudo-euclidien, comme le classifieur aux plus proches voisins, le classificateur de Parzen, ou le classifieur linéaire et quadratique.

La troisième approche c'est le classement directement dans l'espace des dissimilitudes. Le classifieur le plus répandu est basé sur la règle des plus proches voisins. Les machines à vecteurs de support peuvent aussi y être appliquées.

La règle de K plus proches voisins est la technique la plus simple et facile à appliquer directement dans l'espace des dissimilitudes. La règle KNN fonctionne bien, mais c'est une technique locale, qui a besoin d'un vaste ensemble d'apprentissage pour être efficace, elle est sensible au bruit et ne convient pas pour des distances qui ne respectent pas l'inégalité triangulaire.

L'apprentissage par les SVM est très attractif car c'est une méthode simple et non-paramétrique [Vapnik 1995 ; Burges 1998]. De plus, les règles de décision ne dépendent que des produits scalaires entre les observations ou d'une fonction « noyau ». C'est par les fonctions « noyaux » que l'on peut construire des hyperplans séparateurs dans des espaces de dimensions très élevées à temps de calcul constant. Ainsi les SVM contournent les deux formes de la «malédiction de la dimensionnalité» : l'augmentation des paramètres amenant à une extrême complexité du modèle et à son instabilité, et la prolifération des paramètres produisant le sur apprentissage. Peut-être la plus grande limitation de l'approche des SVM réside dans le choix du noyau. Ce choix est à faire par l'expérimentateur, il est primordial pour la bonne performance des SVM. Une fois le noyau fixé, des classificateurs SVM n'ont qu'un seul paramètre choisi par l'utilisateur – la pénalité pour des erreurs. Une deuxième limite est la rapidité et la dépendance de la taille des données durant l'étape de l'apprentissage. Il y a plusieurs méthodes proposées pour résoudre le problème d'optimisation des machines à vecteurs de support sur des bases des données à grand nombre d'observations. On a choisi dans notre travail la méthode d'Osuna [Osuna el al. 1997], Joachims [Joachims 1999], [Suykens et al. 2002], [Suykens et a. 1999].

Trouver des classifieurs pour des dissimilitudes non-euclidiennes est directement lié à l'étude des noyaux non définis positifs. Même si dans ces cas-là, la convergence n'est pas garantie, ces approches ont été appliquées avec succès. On note alors les études concernant les noyaux basés sur la distance tangentielle [Haasdonk&Keysers 2002], la distance « Dynamic Time Warping », la divergence de Kullback – Leibler, la distance de Mahalanobis [Haasdonk&Pekalska 2009] et autres distances [Hammer&Vilmann 2005].

Dans la présente étude, on s'intéresse particulièrement à la troisième approche – classement directement dans l'espace des dissimilitudes. La méthode du « Coefficient de forme » qui sera présenté dans le chapitre suivant, se situe comme un compromis alternatif entre les méthodes de KNN, le choix des prototypes de Pekalska et le positionnement multidimensionnel. On cherche à réduire l'espace de représentation par dissimilitudes tout en gardant une approche globale à partir de la connaissance de toutes les dissimilitudes entre les observations. Le « Coefficient de forme » (*Cs*) est défini à partir de statistiques simples (moyenne et variance) sur les données de dissimilitudes. Les règles de décision qui sont proposées dans le chapitre 2, sont des règles simples qui reprennent d'une part les démarches des classifieurs gaussiens et les SVM.

Chapitre 2. L'approche « Coefficient de forme »

Dans ce chapitre on présentera le classifieur « Coefficient de forme » utilisant les matrices de dissimilitudes. On introduira l'analyse discriminante géométrique dont nous avons repris le principe pour développer la méthode proposée. Dans une première démarche, notre *approche en est directement inspirée et nous avons défini des règles de décision afin d'imiter les comportements des classifieurs linéaire et quadratique. Le nombre de paramètres dans ce cas est limité (deux par classe). Suivant une deuxième approche, la règle de décision se formalise comme pour les machines à vecteurs de support. Cette approche offre une meilleure optimisation des paramètres par classe. On terminera par une synthèse des résultats obtenus sur des données artificielles – des distributions gaussiennes bidimensionnelles et une analyse du comportement du classifieur « Coefficient de forme » sur des matrices de dissimilitudes creuses.*

2.1. Analyse Discriminante Géométrique

Dans notre étude, on se concentre sur les méthodes géométriques de l'analyse discriminante. Ces méthodes ne reposent que sur des notions de distance et d'inertie associées à la partition en classes [Celeux 1990]. La règle linéaire de discrimination part du principe qu'une observation est classée dans le groupe pour lequel sa distance au sens de Mahalanobis, au centre du groupe est minimale. La fonction discriminante prend en compte la matrice de variances – covariances cumulée pour toutes les classes. Cette règle est optimale pour des partitions en classes suivant des distributions gaussiennes de même matrice de variance - covariance, mais de moyennes différentes.

Soit *E*, un ensemble de *n* individus $\mathbf{x}_i$ (*i=1* à *n*) décrits par *p* variables constituant l'ensemble d'apprentissage. Soit un problème de discrimination à deux classes ω_1 et ω_2. En reprenant l'équation 1.18 et en sans considérer l'information apportée par les probabilités a priori, l'équation de la fonction discriminante séparant les classes est :

$$f(\mathbf{x}) = \left[\mathbf{x} - \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)\right]^T \Sigma^{-1}(\mathbf{m}_1 - \mathbf{m}_2), \qquad (2.1)$$

avec Σ la matrice de variance - covariance globale. Les centres de gravité de ces deux classes sont $\mathbf{m}_1$ et $\mathbf{m}_2$ respectivement pour la classe ω_1 et ω_2. La classe est attribuée suivant le signe de la fonction discriminante. Cette règle de décision revient à attribuer la classe correspondant à la plus faible distance de Mahalanobis ($M=\Sigma^{-1}$) entre l'observation à classer et le centre de gravité $\mathbf{m}_1$ ou $\mathbf{m}_2$ (2.2). La discrimination linéaire est une des méthodes les plus utilisées parce que les calculs sont simples à mettre en œuvre avec peu de paramètres à estimer. Les frontières de décision sont linéaires (Fig. 2.1a).

$$classe(\mathbf{x}) = \arg\min_i (d_{\Sigma^{-1}}(\mathbf{x}, \mathbf{m}_i)) . \quad (2.2)$$

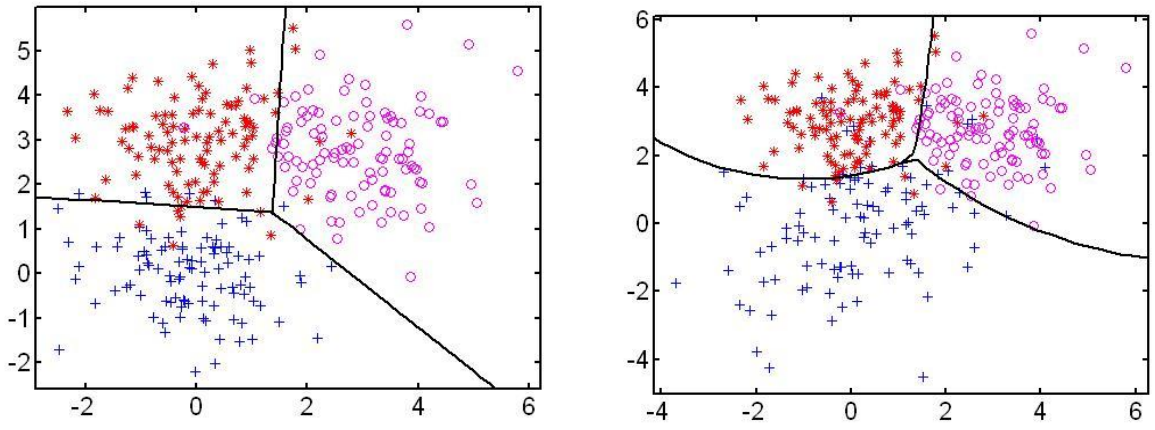
Le principe pour l'analyse discriminante quadratique est de mesurer la distance entre une observation et une classe en utilisant une métrique locale. De façon similaire à la discrimination linéaire, la métrique locale est l'inverse de la matrice de variance - covariance estimée pour chaque classe. La règle de décision devient :

$$classe(\mathbf{x}) = \arg\min_i (d_{\Sigma_i^{-1}}(\mathbf{x}, \mathbf{m}_i)) . \quad (2.3)$$

Les frontières entre les classes ne sont plus linéaires (Fig. 2.1b).

L'hypothèse de la même matrice de variance - covariance pour chaque classe s'avère à la fois forte et donc contraignante dans les applications réelles, et à la fois raisonnable, quand ces mêmes applications réelles ne permettent pas de faire des estimations robustes de tous les paramètres des matrices de variance - covariance. Cette même hypothèse est levée en utilisant la discrimination quadratique. Mais comme indiqué précédemment cette augmentation de degrés de liberté ne conduit pas systématiquement à une amélioration de la qualité du classifieur. Il est préférable alors d'engager une méthodologie graduelle aboutissant à un classifieur parcimonieux où le nombre de paramètres peut être ajusté à la complexité du problème de classification [Bensmail&Celeux 1996].

C'est en reprenant ce principe de flexibilité graduelle d'une règle de décision linéaire jusqu'à quadratique, que l'on proposé des règles de décision sur les mesures de dissimilitudes, afin de se rapprocher du comportement de ces classifieurs de base.



(a) *(b)*

Fig. 2.1 (a) Discrimination linéaire entre trois classes de distributions gaussiennes bidimensionnelles avec les moyennes [0 0], [0 1] et [1 1] respectivement et de même matrice identité de variance - covariance ; (b) Discrimination quadratique entre trois classes de distributions gaussiennes de moyennes [0 0], [0 1] et [1 1] respectivement et matrices de variance - covariance [2 1; 1 4], [1 0 ; 0 1] et [1 0 ; 0 1] respectivement.

2.2. Le « Coefficient de forme » - approche géométrique

Dans ce travail, nous proposons une approche alternative de celles déjà mentionnées dans le chapitre 1. Nous utilisons l'espace de représentation par dissimilitudes, inspiré du travail de Pekalska et nous définissons après un recodage des dissimilitudes, un ensemble de différentes règles de décision, en suivant la même démarche que l'analyse discriminante géométrique.

Si les données de dissimilitude sont isométriques à la distance euclidienne, la ressemblance entre l'approche proposée ci-dessous et le classifieur linéaire ou quadratique sera renforcée, et les règles de décision auront une interprétation géométrique. Sinon, comme pour l'approche des noyaux de distance, cette méthode peut néanmoins être appliquée et obtenir empiriquement des bons résultats.

La justification de la méthode proposée ci-dessous est la suivante : pour un classifieur linéaire, la règle de décision est uniquement basée sur la distance entre les observations et les barycentres des classes. Ces distances peuvent être estimées à partir des dissimilitudes. Dans le cas de distances euclidiennes, le théorème d'Huygens s'applique (voir sec. 2.2.1) et ces estimations sont exactes.

Pour un classifieur quadratique, nous devons estimer une métrique locale pour chaque classe. C'est-à-dire qu'il faut approximer la distance de Mahalanobis, à l'aide des dissimilitudes initiales. C'est dans ce but que nous avons défini le coefficient de forme (*Cs*) à partir de statistiques simples (moyenne et variance) appliquées aux dissimilitudes. Les règles de décision proposées sont alors basées sur ce coefficient, qui pour des distances euclidiennes peut être considéré comme une approximation de la distance de Mahalanobis.

2.2.1. Théorème d'Huygens

On considère un ensemble X de N objets, $X = \{o_i, i = 1,...,N\}$. Soit D la matrice des distances euclidiennes au carré entre ces objets. D est de taille $N \times N$ telle que $D_{ij} = d^2(o_i, o_j) : 1 \le i, j \le N$. Soit e une observation de X. La distance au carré de cette observation e à une autre observation o_i de X définie une variable aléatoire notée $d^2(e, X)$. La moyenne empirique de cette variable est :

$$\overline{d^2(e,X)} = \frac{1}{N}\sum_{o_i \in X} d^2(e,o_i). \tag{2.4}$$

Cette quantité peut aussi être interprétée comme l'inertie de X par rapport à e. L'inertie de X par rapport au centre de gravité g de X est alors $I(X) = \overline{d^2(g,X)}$. Ainsi d'après le théorème de Huygens, l'inertie se décompose ainsi :

$$\overline{d^2(e,X)} = d^2(g,e) + I(X). \tag{2.5}$$

Notons que le centre de gravite g peut être approximé par une observation $\hat{g}$ appartenant à X, telle que :

$$\hat{g} = \arg\min_{e \in X} (\overline{d^2(e, X)}) \tag{2.6}$$

De plus, on peut réécrire l'inertie comme :

$$I(X) = \frac{1}{2N^2} \sum_{o_i \in X} \sum_{o_j \in X} d^2(o_i, o_j) = \frac{1}{N^2} \sum_{o_i \in X} \sum_{\substack{o_j \in X \\ j>i}} d^2(o_i, o_j). \tag{2.7}$$

Ainsi dans le contexte euclidien et en regroupant les équations (2.5) et (2.6), on peut calculer la distance à une observation e au centre de gravité g sans connaitre explicitement la position de ce dernier :

$$d^2(g, e) = \overline{d^2(e, X)} - \frac{1}{2N^2} \sum_{o_i \in X} \sum_{o_j \in X} d^2(o_i, o_j). \tag{2.8}$$

La moyenne des distances au carré fournit donc l'information sur la position d'un point à un ensemble de points. On va suivre cette même démarche considérant maintenant la variance empirique Var$\{ d^2(e, X) \}$ des distances d'une observation quelconque e à l'ensemble X :

$$\mathrm{Var}\{d^2(e, X)\} = \frac{1}{N} \sum_{o_i \in X} \left(d^2(e, o_i) - \overline{d^2(e, X)} \right)^2 . \tag{2.9}$$

Cette quantité est plus complexe à interpréter. Cependant on peut considérer qu'elle donne en première approximation, une information sur l'orientation de l'ensemble de données X, vue depuis cette observation e, comme cela est illustré sur la figure 2.2. En effet, la variance Var$\{ d^2(e, X) \}$ est plus grande pour des observations dans la direction principale de X et plus petite dans la direction orthogonale, sous l'hypothèse de distribution connexe gaussienne du nuage des points.

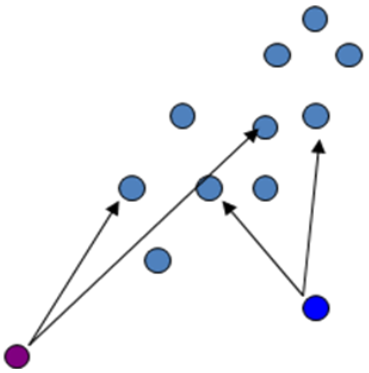
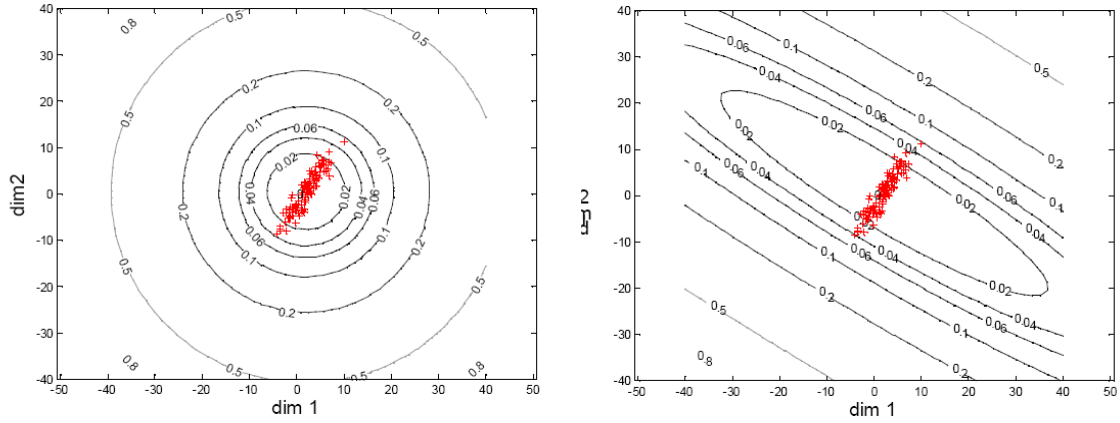


Fig. 2.2 Effet de la position des observations e sur la variance Var$\{ d^2(e, X) \}$.

Les figures 2.3a.b. illustrent l'évolution spatiale de ces statistiques de moyenne et variance relativement à un nuage de points *X* suivant une distribution gaussienne bidimensionnelle (les points sont marqués par les x), pour une observation *e* quelconque dans le plan 2D.



(a) *(b)*

Fig. 2.3 (a) Illustration du contour de l'évolution spatiale de $\overline{d^2(g,e)}$ *; (b) Illustration du contour de l'évolution spatiale de Var{* $d^2(e,X)$ *}. Les points de la distribution gaussienne bidimensionnelle X sont marqués avec des x au centre des figures.*

La moyenne est calculée suivant l'équation (2.4) et la variance, suivant l'équation (2.9). Les contours de l'évolution spatiale de la moyenne sont les contours circulaires attendus des lieux de distances constantes à l'origine (centre de gravité du nuage *X*). Les contours de l'évolution spatiale de la variance mettent en évidence l'orientation du nuage de points *X* : la variance augmente beaucoup plus rapidement quand l'observation *e* s'éloigne du centre de gravité de *X* suivant l'orientation principale du nuage (Sud-Ouest, Nord-Est), que suivant l'orientation perpendiculaire (Nord-Ouest, Sud-Est).

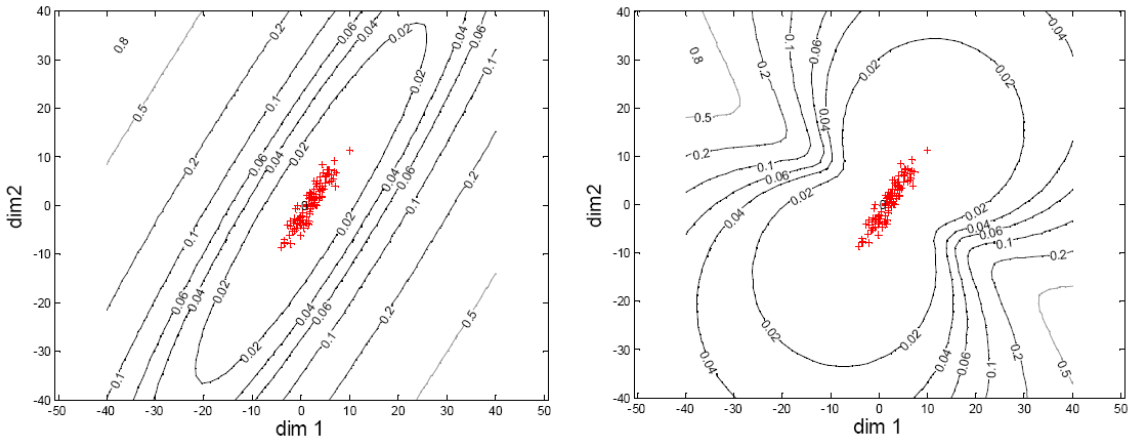
2.2.2. L'indice de proximité « Coefficient de forme »

Dans l'objectif d'en déduire une règle de décision, on retiendra que la moyenne empirique $\overline{d^2(e,X)}$ fournit une information de positionnement par rapport à un ensemble d'observations, qui pourra être une classe dans le cas de l'analyse discriminante. La variance empirique Var{ $d^2(e,X)$ } met en oeuvre des moments d'ordre supérieur de la distribution de *X* et globalement cette quantité capte une information de « forme » de la distribution des points de *X*, ainsi que la dimension de l'espace vectoriel sous-jacent (dimensions intrinsèques des donnée). En suivant le raisonnement de l'analyse discriminante, on recherche les combinaisons de variables qui permettent de séparer le mieux les différentes classes (maximiser l'inertie inter-classe) en gardant dans chaque classe une « étendue » minimale (minimiser l'inertie intra-classe). Alors, on propose un indice de proximité, appelé « Coefficient de forme », *Cs(e, X)*, basé sur les statistiques présentées dans 2.2.1.

$$Cs(e,X) = \frac{\left(\overline{d^2(e,X)} - I(X)\right)^2}{\mathrm{Var}\left\{d^2(e,X)\right\}}. \qquad (2.10)$$

La variance étant un moment statistique de deuxième ordre, on élève au carré la moyenne au numérateur, afin que le « Coefficient de forme » soit sans dimension.

Comme la discrimination géométrique reposant sur la distance de Mahalanobis est optimale dans le cas de distributions gaussiennes, on va étudier comment une règle de décision à partir du « Coefficient de forme » peut approximer celle utilisant la distance de Mahalanobis. On comparera l'évolution de ce coefficient avec l'évolution de la distance de Mahalanobis d'une observation *e* au centre de gravité d'un nuage *X* de points de distribution gaussienne. La matrice de variance - covariance de *X* est calculée sur l'ensemble des points et l'observation *e* est n'importe où dans le plan 2D. Suivant le même principe que pour les figures 2.3a.b, la figure 2.4.a illustre l'évolution spatiale de cette distance de Mahalanobis, et la figure 2.4b., celle du « Coefficient de forme ». L'indice *Cs* croît lentement pour des observations qui sont dans la direction du nuage et croît plus rapidement pour les points situés dans la direction orthogonale. Son comportement est semblable à celui de la distance Mahalanobis $d_M^2(g,e)$, calculée suivant l'équation 1.3 avec Σ la matrice de variance - covariance de *X*. Dans cette direction orthogonale, on peut observer un « pincement » dans l'évolution du *Cs*, dû au décroissement rapide de la variance.



(a) *(b)*

Fig. 2.5 (a) Illustration du contour de l'évolution spatiale de $d_M^2(g,e)$*; (b) Illustration du contour de l'évolution spatiale de Cs(e,X).*

Afin de pouvoir s'adapter par apprentissage à différentes distributions et s'approcher au mieux du comportement de la distance de Mahalanobis, on définit un coefficient de forme paramétrable, à deux paramètres γ et δ, avec $\delta > 0$ et $\gamma > 0$:

$$Cs(e) = \gamma \frac{\left(\overline{d^2(e)} - I\right)^2}{\left(\mathrm{Var}\left\{d^2(e)\right\}\right)^\delta}. \qquad (2.11)$$

Considérons la décomposition spectrale de la matrice de variance - covariance d'un nuage gaussien et en suivant les travaux de Celeux, Govaert et collaborateurs [Biernacki et al. 2006], [Celeux&Govaert 1995], ainsi cette matrice Σ_X peut être exprimée par :

$$\Sigma_X = \lambda BAB', \tag{2.12}$$

où $\lambda = |\Sigma_X|^{1/d}$, d la dimensionnalité de l'espace, B la matrice des valeurs propres, et A la matrice diagonale des valeurs propres normalisées ($|A| = 1$). La valeur scalaire λ détermine le « volume », B l'orientation et A la forme de la distribution X.

Dans notre cas, la distribution est connue comme étant normale multi-variée, alors la matrice de variance - covariance la spécifie complètement, à l'exception de la position de la moyenne. Alors les paramètres du coefficient de forme sont tels que δ puisse capter les variabilités de « forme » et de dimension et γ, les variabilités de «volume» de cette distribution.

Alors si on considère différentes distributions gaussiennes, pour chaque distribution les deux paramètres (γ, δ) peuvent être ajustés afin de minimiser l'erreur quadratique moyenne entre le Cs et la distance de Mahalanobis pour tous les points de référence e. On note l'erreur quadratique moyenne :

$$\overline{E_{err}^2(e)} = \overline{\left(\log\left(d_M^2(e,g)\right) - \log\left(Cs(e,X)\right)\right)^2}\,. \tag{2.13}$$

Les paramètres sont alors obtenus tels que :

$$\delta^* : \frac{\partial \overline{E_{err}^2(e)}}{\partial \delta} = 0 \quad \eta^* : \frac{\partial \overline{E_{err}^2(e)}}{\partial \eta} = 0 \quad \text{avec } \eta = \log(\gamma)\,. \tag{2.14}$$

Ces deux paramètres d'ajustement sont fixés afin de minimiser l'erreur quadratique moyenne entre la distance de Mahalanobis et le « Coefficient de forme ».

Pour les simulations, les distributions gaussiennes X sont paramétrées de manière aléatoire selon une procédure à partir de la décomposition spectrale. On fait varier la dimension d de 2 à 10. Pour chaque dimension, on génère vingt « formes » différentes et pour chacune des ces « formes », sept volumes différents. On aboutit ainsi à 9 x 20 x 7 = 1260 distributions différentes à moyenne nulle. Pour chaque expérience nous générons 20 distributions X de 1000 points. Des ensembles de 2500 points de référence e sont générés aléatoirement suivant une distribution uniforme.

Ainsi, en faisant varier les valeurs du volume, de l'orientation et de la forme, on obtient différents modèles pour décrire ces distributions. Ils sont divisés en trois familles:

- La famille générale où toutes les quantités descriptives (volume, forme, orientation) sont variables;

- La famille diagonale où nous supposons que la matrice de variance - covariance est diagonale, ce qui signifie que la matrice d'orientation *B* est une matrice de permutation;
- La famille sphérique où nous supposons que la matrice de variance - covariance est diagonale avec les mêmes valeurs sur la diagonale ce qui signifie que $A = I$, où I désigne la matrice identité.

Sur la figure 2.6 on montre les résultats des 20 configurations aléatoires bidimensionnelles dans la famille générale, pour $\lambda = 1$ et $d=2$.

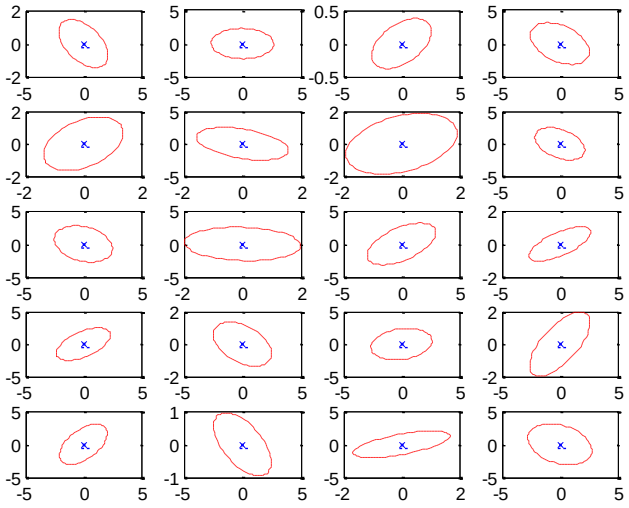


Fig. 2.6 Contours à un écart-type des 20 matrices de variance - covariance pour d = 2 et λ = 1 de la famille générale.

Du fait de la construction du *Cs*, le paramètre δ doit être indépendant du volume λ de la distribution et il doit capter les autres caractéristiques : forme et orientation. Au contraire, le paramètre γ doit être principalement lié au volume λ.

Nous avons fait les simulations avec trois modèles du coefficient *Cs* suivant si les paramètres sont fixés ou non (éq. 2.11), du plus simple au plus complexe. Dans l'équation (2.11), le « Coefficient de forme » correspond au modèle 3, le plus complexe, où les deux paramètres d'ajustement sont libres afin que l'on puisse observer la variation des deux indépendamment l'un de l'autre. Cette même procédure peut être reprise avec les modèles 1 et 2, où un paramètre est fixé, et l'autre libre.

Modèle 1

$$\delta^*: \frac{\partial \overline{E_{err}^2(e)}}{\partial \delta} = 0, \quad \gamma = \text{constante} = 1 \tag{2.15}$$

Modèle 2

$$\gamma^*: \frac{\partial \overline{E_{err}^2(e)}}{\partial \gamma} = 0, \quad \delta = \text{constante} = 1 \tag{2.16}$$

Modèle 3

$$\delta^*: \frac{\partial \overline{E_{err}^2(e)}}{\partial \delta} = 0, \gamma^*: \frac{\partial \overline{E_{err}^2(e)}}{\partial \gamma} = 0\,. \tag{2.17}$$

L'erreur quadratique moyenne $\overline{E_{err}^2(e)}$ est minimisée pour toutes les configurations des points. En prenant une des configurations illustrées à la figure 2.6, la figure 2.7 montre la qualité de l'ajustement entre l'indice *Cs* « de base » ($\delta = 1$, $\gamma = 1$) et la distance de Mahalanobis. Le coefficient de corrélation entre ces deux indices pour les 2500 points de référence est déjà très élevé (0.88).

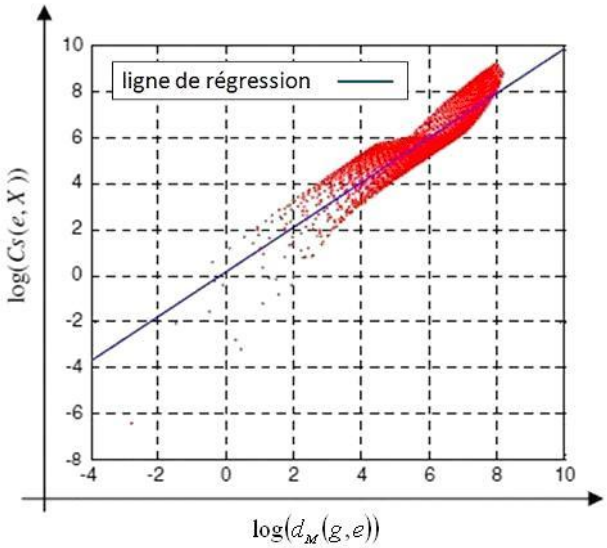
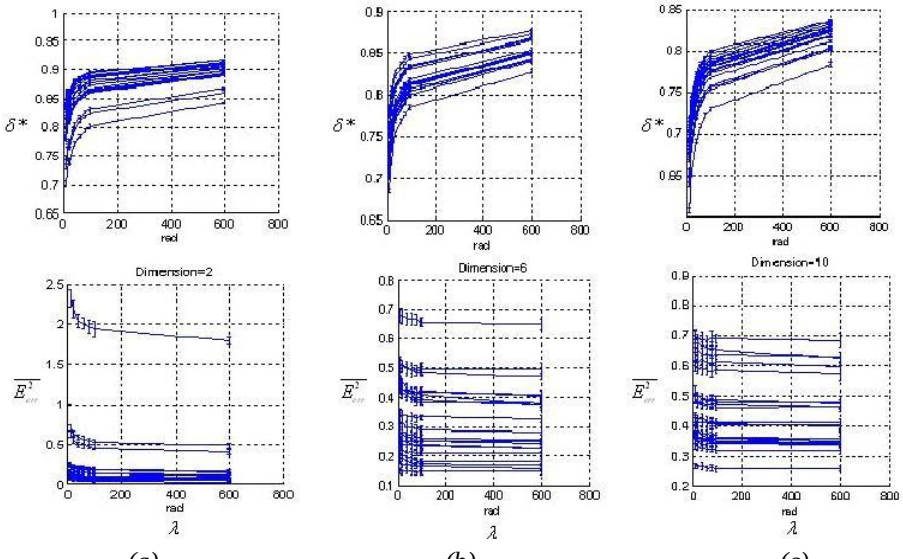


Fig. 2.7 Distribution conjointe du logarithme de l'indice Cs avec δ = 1, γ = 1 et de la distance de Mahalanobis pour tous les points de référence e, pour une des configurations Σ_X .

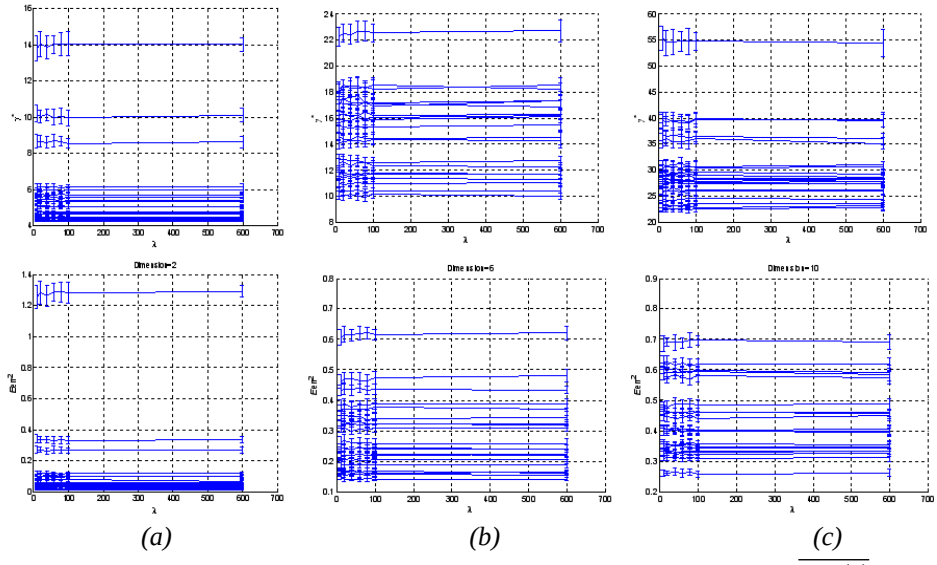
Les figures 2.8, 2.9 et 2.10 illustrent les résultats des simulations pour des modèles 1, 2 et 3, respectivement, pour les dimensions 2, 6 et 10 de la matrice de covariance et pour tous les volumes et formes.



(a) *(b)* *(c)*

Fig. 2.8 Illustrations des évolutions des moyennes et des variances du paramètre δ, et $\overline{E_{err}^2(e)}$, calculées suivant le modèle 1 pour des dimensions 2 (a), 6 (b) et 10 (c) et pour tous les volumes et formes.*

La figure 2.8 illustre des évolutions des moyennes et des variances du paramètre δ, et $\overline{E_{err}^2(e)}$, calculées suivant le modèle 1 (équation 2.15). Les valeurs de δ sont plus petites, car γ = 1 (suivant le modèle 3, fig.2.10b., les valeurs de γ sont assez grandes) et nous pouvons voir que le paramètre γ influence les valeurs de δ en comparant le comportement des deux modèles 1 et 3.



(a) *(b)* *(c)*

Fig. 2.9 Illustrations des évolutions des moyennes et des variances du paramètre γ, et $\overline{E_{err}^2(e)}$, calculées suivant le modèle 2 pour des dimensions 2 (a), 6 (b) et 10 (c), et pour tous les volumes et formes.*

La figure 2.9 illustre des évolutions des moyennes et des variances du paramètre γ, et $\overline{E_{err}^2(e)}$, calculées suivant le modèle 2 (équation 2.16). Les valeurs de γ varient beaucoup plus avec le changement du volume. Les valeurs de ce paramètre sont plus petites que dans le cas du modèle 3 (fig.2.10b.,), car δ = 1, car la limitation de δ influence les valeurs de γ. On peut aussi observer que les évolutions de $\overline{E_{err}^2(e)}$ pour les deux modèles 1 et 2 sont très semblables.

Les résultats illustrés de la figure 2.10a. représentent l'évolution de la moyenne et la variance des paramètres d'ajustement δ, γ et $\overline{E_{err}^2(e)}$, calculés suivant le modèle 3 (équation 2.17). Ces résultats expérimentaux montrent que le paramètre d'ajustement δ est indépendant du volume mais dépend de la forme, l'orientation et la dimension de la distribution de X. Pour la famille sphérique, on constate que le paramètre δ est proche de 1. On peut établir aussi que la gamme des valeurs pour δ est limitée (suivant la figure 2.10a. dans l'intervalle [0.8 ; 1.2]), cette propriété sera utilisée plus tard dans l'étape d'apprentissage pour accélérer l'apprentissage. Une seule exception est visible pour le cas d = 2, où la valeur de la moyenne de δ est proche de 1.6. On constate que ces valeurs du paramètre correspondent à une matrice de variance - covariance de dimension 2 × 2 quasi-diagonale avec une différence importante entre les 2 variances. La distribution est très étirée et aplatie. Pour ce même cas à la figure 2.10c. on remarque aussi que l'erreur quadratique moyenne dépasse de manière significative les erreurs moyennes des autres cas. Les valeurs de la moyenne de δ reviennent dans l'intervalle [0.8 ; 1.2] avec l'augmentation de la dimensionnalité de la matrice de variance - covariance. De plus, cette configuration

particulière de variances deviendra de moins en moins fréquente avec l'augmentation de la dimensionnalité.

L paramètre γ capture essentiellement les variations de volume, et aussi en partie les variations de forme et d'orientation. Cette dernière dépendance augmente avec la dimension de la distribution de *X*. Ainsi, la dimension affecte les deux paramètres. Effectivement avec l'augmentation de la dimension, le nombre de paramètres de la matrice Σ_X augmente de façon quadratique, mais le nombre de paramètres à ajuster dans le modèle *Cs* reste constant (2: δ, γ). Ainsi, dans le cas du paramètre γ on ne peut pas trouver un intervalle de variation des valeurs comme pour le paramètre δ. Le paramètre d'ajustement γ varie très fortement vers des très grandes valeurs qui peuvent tendre vers infini, comme on peut l'observer à la figure 2.10b.

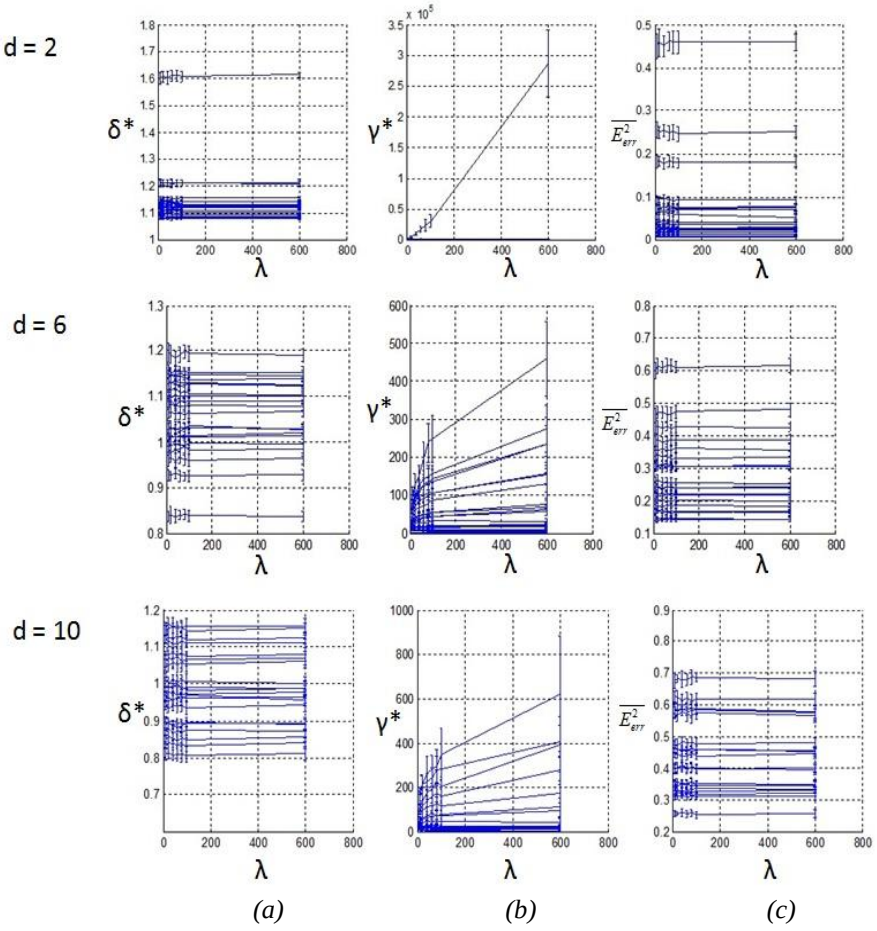
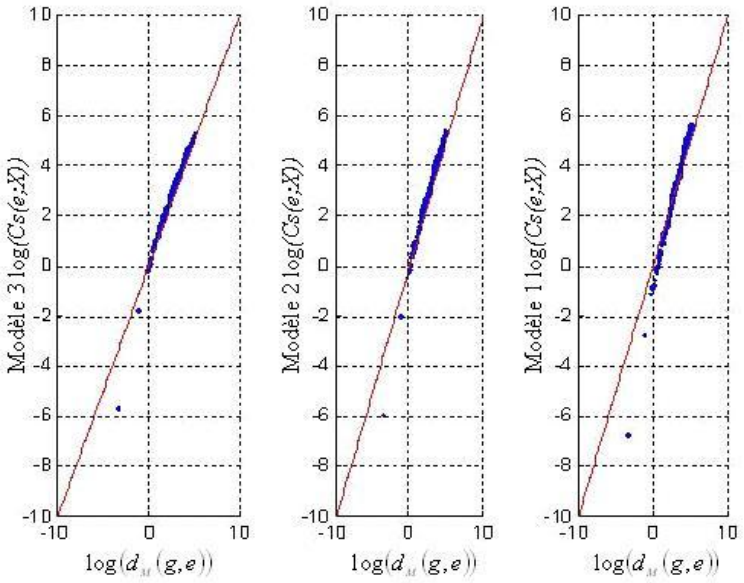


Fig. 2.10 Illustrations des évolutions des moyennes et des variances des paramètres δ (a), γ* (b) et $\overline{E_{err}^2(e)}$ (c), calculées suivant le modèle 3 pour des dimensions 2, 6 et 10 et pour tous les volumes et formes.*

La figure 2.11 illustre la qualité d'ajustement selon les trois modèles. Les paramètres d'ajustement sont fixés par minimisation de l'erreur quadratique moyenne (équation 2.13). Pour l'exemple de cette figure, l'erreur quadratique moyenne est respectivement de 0.0100 (modèle 3), 0.0143 (modèle 2) et 0.0728 (modèle 1). Cette erreur relève bien qu'il est mieux d'approximer avec deux paramètres d'ajustement libres (modèle 3) qu'avec un paramètre fixé et l'autre libre (modèles 1 et 2).

Aussi à l'aide de ces expériences, on peut déduire que les valeurs du paramètre d'ajustement δ dépendent de la forme particulière de la distribution et ne dépendent pas du volume de la distribution, elles restent aussi limitées dans un intervalle étroit, tandis que les valeurs du paramètre d'ajustement γ dépendent du volume de la distribution et ne peuvent pas être limitées.



(a) *(b)* *(c)*

Fig. 2.11 (a) Qualité d'ajustement du modèle 3 ; (b) Qualité d'ajustement du modèle 2 ; (c) Qualité d'ajustement du modèle 1.

Finalement on peut conclure que les simulations étendues sur les distributions gaussiennes de familles et de dimensions différentes conduisent aux observations suivantes:

- Le paramètre d'ajustement δ prend en compte la forme particulière de la distribution.
- Le paramètre δ peut être délimité dans un intervalle étroit (par exemple [0.8 ; 1.2]), et pour la famille sphérique, les valeurs de δ sont proche de 1.
- Le paramètre δ ne dépend pas du volume de la distribution.
- Nous ne pouvons pas déterminer des limites strictes pour le paramètre d'ajustement γ mais les valeurs de ce paramètre d'ajustement dépendent des variations de volume des distributions.
- Finalement l'erreur quadratique moyenne montre que l'ajustement avec deux paramètres est meilleur que celui avec un paramètre libre et l'autre fixé à 1.

2.3. Classement par rapport la règle géométrique linéaire

2.3.1. Règles de décision

Les règles de décision utilisant le *Cs* ont été établies à partir des expériences sur le comportement de l'indice *Cs* présentées ci-dessus. Ces règles de décision permettent de définir

une famille cohérente, imitant l'évolution de la complexité de la classification d'un classificateur linéaire ou quadratique.

La justification de ces règles de décision vient des analogies avec le classificateur linéaire en supposant que les dissimilitudes entre les données sont en fait les distances euclidiennes dans un espace euclidien sous-jacent inconnu. Sinon, l'exacte relation avec l'inertie (équations 2.5 à 2.8) n'est plus valide. Néanmoins, en considérant que l'indice *Cs* est un nouveau critère pour calculer la proximité entre une observation et un ensemble d'observations (une classe), les règles de décision sont simplement calculées à partir de statistiques (moyenne et variance) appliquées aux valeurs de dissimilitude.

Soit X un ensemble d'observations appartenant à C classes. Soit X_c l'ensemble d'observations pour la classe ω_c (c = 1, …, C). La règle la plus simple est de classifier une nouvelle observation e dans la classe qui minimise la relation suivante :

$$classe(e) = \arg\min_c \left(\overline{d^2(e, X_c)} - I(X_c) \right). \tag{2.18}$$

Cette règle est le résultat direct de l'équation (2.8) appliquée à la classe ω_c. Dans le cadre euclidien, cette règle définit la proximité de la classe comme une distance euclidienne entre l'observation e et le barycentre g_c de la classe ω_c et les frontières entre classes sont linéaires.

Pour améliorer sa performance, on peut ajouter l'information du volume de chaque classe à l'aide de l'inertie $I(X_c)$ comme suit :

$$classe(e) = \arg\min_c \left(\frac{\overline{d^2(e, X_c)} - I(X_c)}{I(X_c)} \right). \tag{2.19}$$

Enfin, la dernière règle de décision prend en compte la structure totale de la classe (volume, forme, orientation) en utilisant l'indice de proximité *Cs* :

$$classe(e) = \arg\min_c \left(\gamma_c \frac{\left(\overline{d^2(e, X_c)} - I(X_c) \right)^2}{\left(\mathrm{Var}\left\{ d^2(e, X_c) \right\} \right)^{\delta_c}} \right). \tag{2.20}$$

Le comportement de ces trois règles de décision est illustré à la figure 2.12 pour un classement en trois classes non linéairement séparables. Chaque classe est issue d'une distribution gaussienne bidimensionnelle. On compare leur comportement avec un classifieur quadratique standard. La règle utilisant le « Coefficient de forme » est paramétrée avec $\gamma_c=1$ et $\delta_c=1$, $c=1..C$. L'objectif de cette illustration n'est pas une comparaison quantitative des performances, mais plutôt qualitative de l'allure (linéaire ou quadratique) des frontières de décision.

Considérons l'équation (2.20). Deux cas sont étudiés pour l'estimation des deux paramètres d'ajustement :

- une estimation globale des deux paramètres pour toutes les classes,
- ou bien une estimation locale pour chaque classe.

Pour le premier cas, les modèles de classification sont adaptés quand il est pertinent d'examiner une métrique pour toutes les classes (comme pour l'analyse linéaire discriminante). Notons que, le paramètre γ est un facteur de proportionnalité, il n'a donc aucun impact sur une estimation globale. Pour le deuxième cas, une métrique locale doit être apprise par un couple des paramètres (γ_c, δ_c), associés à chaque classe ω_c (comme pour le classificateur quadratique).

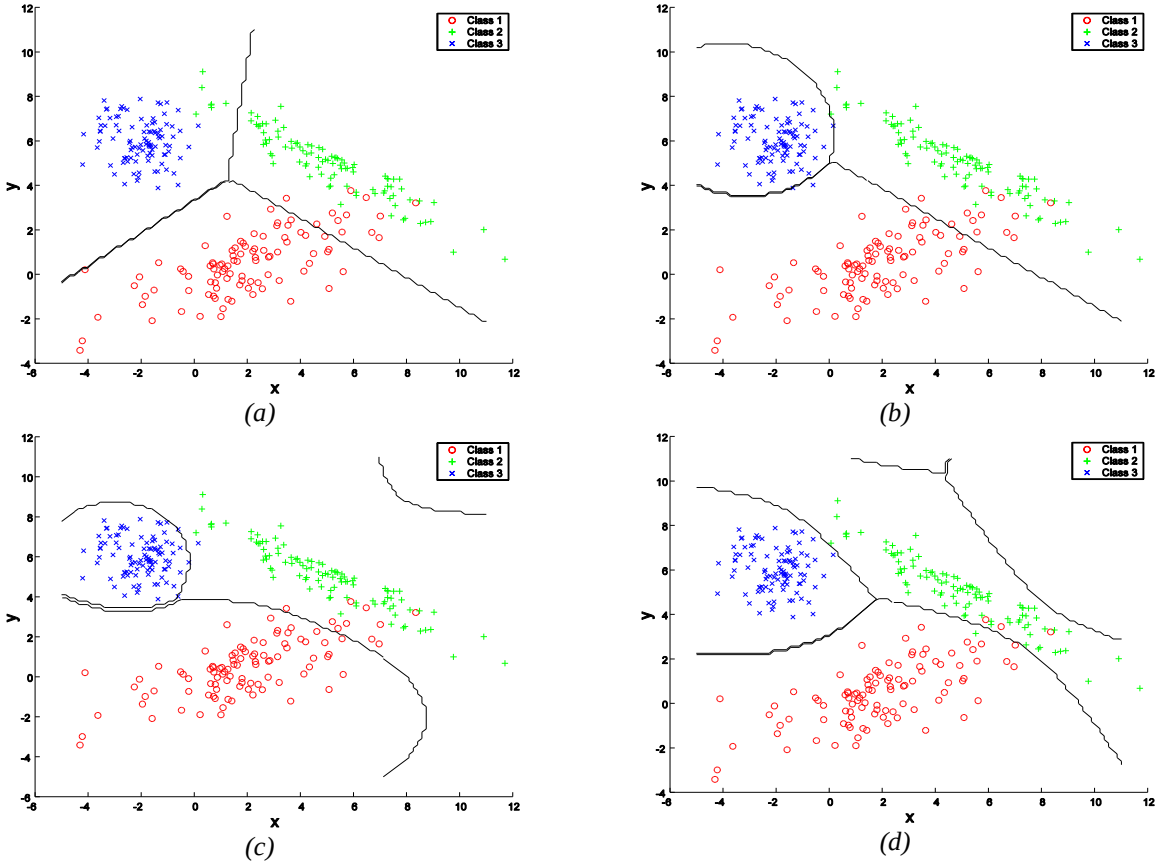


Fig. 2.12 : Exemples de frontières obtenues avec la règle de décision définie par les équations 2.18 (a), 2.19 (b), 2.20 (c), et avec le classifieur quadratique (d) pour le classement de trois distributions gaussiennes non linéairement séparables.

Le tableau 2.1 récapitule ces différentes possibilités et les équivalences entre les modèles de classement qui en résultent.

Les modèles avec l'adaptation globale sont des modèles intermédiaires dont une étude a été déjà menée dans [Guerin&Celeux 2001] et [Koning 2001]. Ils ne seront pas plus étudiés ici dans cette thèse, puisque les performances sont inférieures au modèle complet.

Dans ce travail on se concentrera sur les trois modèles suivants, avec une adaptation locale:

- *Cs*(1, 0) : l'équation (2.18) avec $\gamma_c = 1$, $\delta_c = 0$.

- *Cs* calculé suivant l'équation (2.19).
- $Cs(1, 1)$: l'équation (2.20) avec $\gamma_c = 1$, $\delta_c = 1$.
- $Cs(\gamma_c, \delta_c)$: l'équation (2.20) avec les paramètres γ_c et δ_c , ajustés pour chaque classe ω_c.

Nous allons maintenant détailler le procédure d'apprentissage pour le modèle local de classement $Cs(\gamma_c, \delta_c)$. A la fin de ce chapitre des résultats expérimentaux sur des distributions gaussiennes seront présentés et les différents modèles du « Coefficient de forme » seront comparés et aussi avec le classifieur « KNN » et celui des prototypes de Pekalska.

Tableau 2.1: L'équivalence entre les différents modèles de classement du « Coefficient de forme »

	Modèle sans adaptation, les deux paramètres sont constants	Modèle avec adaptation globale : $\forall \omega_c$, $\gamma_c = \gamma^*$, *et/ou* $\delta_c = \delta^*$	Modèle avec adaptation locale : γ_c et/ou δ_c $\neq$ pour chaque classe ω_c
Optimiser γ, quand $\delta = 0$	$\gamma = 1$ $Cs(1, 0)$	$\gamma_c = 1$ $Cs(1, 0)$	$\gamma_c \neq$ pour $\forall \omega_c$ $Cs(\gamma_c, 0)$
Optimiser δ, quand $\gamma = 1$	$\delta = 1$ $Cs(1, 1)$	$\forall \omega_c$, $\delta_c = \delta^*$ $Cs(1, \delta^*)$	$\delta_c \neq$ pour $\forall \omega_c$ $Cs(1, \delta_c)$
Optimiser γ, Quand $\delta = 1$	$\gamma = 1$ $Cs(1, 1)$	$\gamma_c = 1$ $Cs(1, 1)$	$\gamma_c \neq$ pour $\forall \omega_c$ $Cs(\gamma_c, 1)$
Optimiser : γ et δ	$\gamma = 1$, $\delta = 1$ $Cs(1, 1)$	$\forall \omega_c$, $\gamma_c = \gamma^*$, $\delta_c = \delta^*$ $Cs(\gamma^*, \delta^*)$	$\gamma_c, \delta_c \neq$ pour $\forall \omega_c$ $Cs(\gamma_c, \delta_c)$

2.3.2. Procédure d'optimisation

L'objectif de la phase d'apprentissage est l'estimation des paramètres d'ajustement (γ_c, δ_c) par rapport aux données de chaque classe. La procédure d'apprentissage consiste à trouver les paramètres (γ_c, δ_c) en maximisant le taux de reconnaissance par validation croisée de l'ensemble d'apprentissage X. Les paramètres sont estimés relativement les uns aux autres, à partir d'une valeur de référence. Sans perte de généralité, on choisit la classe ω_1 comme classe de référence $(\gamma_1 = 1, \delta_1 = 1)$. Les paramètres des autres classes seront estimés par rapport à la classe ω_1. La procédure sera donc sous-optimale, car les caractéristiques propres de la classe de référence ne seront pas ajustées. Notons que la procédure d'apprentissage par « SVM » (paragraphe suivant) lève justement cette difficulté.

Ainsi, relativement à cette approche, pour une tâche de discrimination avec C classes, il reste donc $2 \times (C - 1)$ paramètres à estimer pour le classifieur $Cs(\gamma_c, \delta_c)$. Ce problème à plusieurs classes est un problème d'optimisation difficile car la fonction à optimiser n'est pas explicitement connue, le nombre de paramètres peut être important (si le nombre de classes est grand) et le domaine de variation des paramètres n'est pas a priori borné. Alors pour résoudre ce problème, nous proposons une procédure d'optimisation avec des solutions partielles analytiques et des simulations « Monté Carlo ».

Pour expliquer la procédure d'optimisation, on commence par le problème à deux classes. La procédure sera alors étendue à plus de deux classes. L'espace des paramètres est un espace 2D (deux paramètres inconnus : γ_2, δ_2). La classe ω_1 est la classe de référence ($\gamma_1 = 1$, $\delta_1 = 1$), choisie a priori comme telle. Suivant l'interprétation de γ_2 et δ_2 (section 2.2.2), cet espace est limité en partie avec $\delta_2 \geq 0$ *et* $\gamma_2 > 0$. Pour la classe ω_2, la première étape consiste à initialiser $\gamma_2 = 1$ et $\delta_2 = 1$ pour toutes les observations. La seconde étape de classement sur l'ensemble d'apprentissage est effectuée avec la règle de décision suivante :

$$classe(e) = \begin{cases} \omega_1, \text{si} Cs(e,\omega_1) < Cs(e,\omega_2) \\ \omega_2, \text{si} Cs(e,\omega_1) > Cs(e,\omega_2) \end{cases} \quad (2.21)$$

Avec cette règle, pour toutes les observations bien classées ($X_b \subset X$), les paramètres candidats sont $\gamma_2(e) = 1$ et $\delta_2(e) = 1$. Si e est une observation mal classée, alors $\gamma_2(e)$ et/ou $\delta_2(e)$ doivent être ajustés afin de vérifier l'inégalité appropriée pour la classer correctement. Alors, pour bien classer cette observation, il faut considérer deux cas possibles suivant si l'erreur concerne la première classe ou la seconde. C'est deux possibilités correspondent aux deux inégalités ci-dessous :

- Si la véritable classe de l'observation est ω_1, mais elle est classée dans ω_2 alors nous devons avoir l'inégalité suivante: $Cs(e, \omega_1) < Cs(e, \omega_2)$;
- Si la véritable classe de l'observation est ω_2, mais elle est classée dans la classe ω_1, alors nous devons avoir l'inégalité suivante: $Cs(e,\omega_2) < Cs(e,\omega_1)$;

La ligne de partage se situe lorsque les deux coefficients de forme sont égaux. Ces quantités étant positives, l'égalité peut alors s'écrire à travers la fonction logarithmique. Pour chaque observation e mal classée, cette équation met en jeu les paramètres $\gamma_2(e)$ et $\delta_2(e)$ ainsi :

$$\begin{aligned} \log(\gamma_2) = \delta_2(e) \times \log\left(\text{Var}\left(d^2(e,\omega_2)\right)\right) \\ + \log(Cs(e,\omega_1)) - 2 \times \log\left(\overline{d^2(e,\omega_2)} - I(\omega_2)\right) \end{aligned} \quad (2.22)$$

Dans l'espace (δ_2, log(γ_2)) de $\Re^2$, nous obtenons une équation d'une droite de type $y = ax + b$. Cette droite est une frontière dans l'espace des paramètres: selon le côté choisi, l'observation e est soit bien classée, soit mal classée. En regroupant l'ensemble des observations mal classées ($X_m \subset X$, $N_m = Card(X_m)$), on obtient N_m droites. L'intersection de toutes ces droites définit des régions dans lesquelles le taux d'erreur est constant pour ce sous-ensemble d'observations X_m. Voir l'illustration à la figure 2.13. Si on considère toutes les intersections prises deux à deux, pour l'ensemble des observations mal classées (X_m), il y a $N_m \times (N_m - 1)/2$ couples d'équations. Par la résolution de tous ces systèmes de Cramer, on obtient autant de sommets définis par leurs coordonnées (δ_2, log(γ_2)). Quand le système est mal conditionné, la résolution est annulée. Cet ensemble de sommets est une première connaissance a priori pour guider l'optimisation afin de choisir la configuration optimale (γ_2, δ_2), qui minimise le taux d'erreur.

De manière empirique, on observe que cette distribution des paramètres (γ_2, δ_2) n'est pas connexe, il y a des points aberrants plus particulièrement sur la dimension γ_2. Donc à partir de ce

premier ensemble de paramètres, on va délimiter le domaine de recherche de la configuration optimale. Concernant le paramètre δ_2, et en reprenant les expérimentations expliquées au paragraphe 2.2.2, ce dernier évolue dans un intervalle borné. De plus, cette observation concerne aussi bien les expériences sur des ensembles de données réelles (voir les résultats dans chapitre 4) qu'artificielles (voir les résultats au paragraphe 2.5).

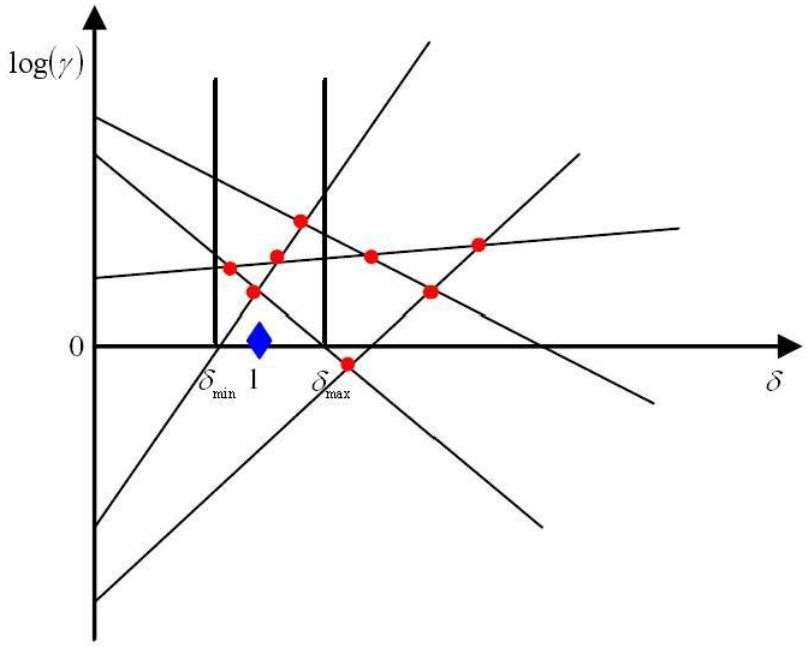


Fig. 2.13 Illustration des N_m contraintes dans l'espace 2D des paramètres avec la limitation de l'ensemble des sommets dans l'intervalle [δ_{min}, δ_{max}].

En conséquence, le domaine de variation de δ_2 sera limité à ($\delta_{2\min}$, $\delta_{2\max}$) directement dans l'espace (δ_2, log(γ_2)). Par contre, concernant le paramètre γ_2, il n'y a pas de limite supérieure. En effet, les résultats expérimentaux sur les distributions gaussiennes (voir paragraphe 2.2.2 et paragraphe 2.5) montrent que ce paramètre est principalement lié au volume de classe et il peut prendre des valeurs importantes. Ainsi, en essayant de résoudre les systèmes de Cramer, on peut aboutir à une indétermination quand log(γ_2) est très grand. Voir l'exemple illustré à la figure 2.14b. Ainsi, pour limiter le domaine de variation de γ_2, nous préférons utiliser les quartiles (25% et 75%) de la distribution au lieu des valeurs minimale ou maximale.

Dès que l'espace des paramètres est ainsi borné, on réalise une procédure de Monte Carlo, en tirant aléatoirement des couples (δ_2, log(γ_2)) dans ce domaine de recherche. Ce tirage se fait suivant une distribution gaussienne centrée sur la médiane des distributions des positions des sommets. Le nombre de tirage est fixé a priori à 1000. Il pourra être augmenté ou diminué suivant la taille de la zone de recherche. Pour chaque configuration aléatoire, l'erreur de classement est calculée sur l'ensemble d'apprentissage. Une seule configuration sera retenue, celle qui minimise l'erreur de classement parmi toutes les valeurs simulées (il peut y avoir plusieurs configurations qui minimisent l'erreur de classement alors on choisit une aléatoirement).

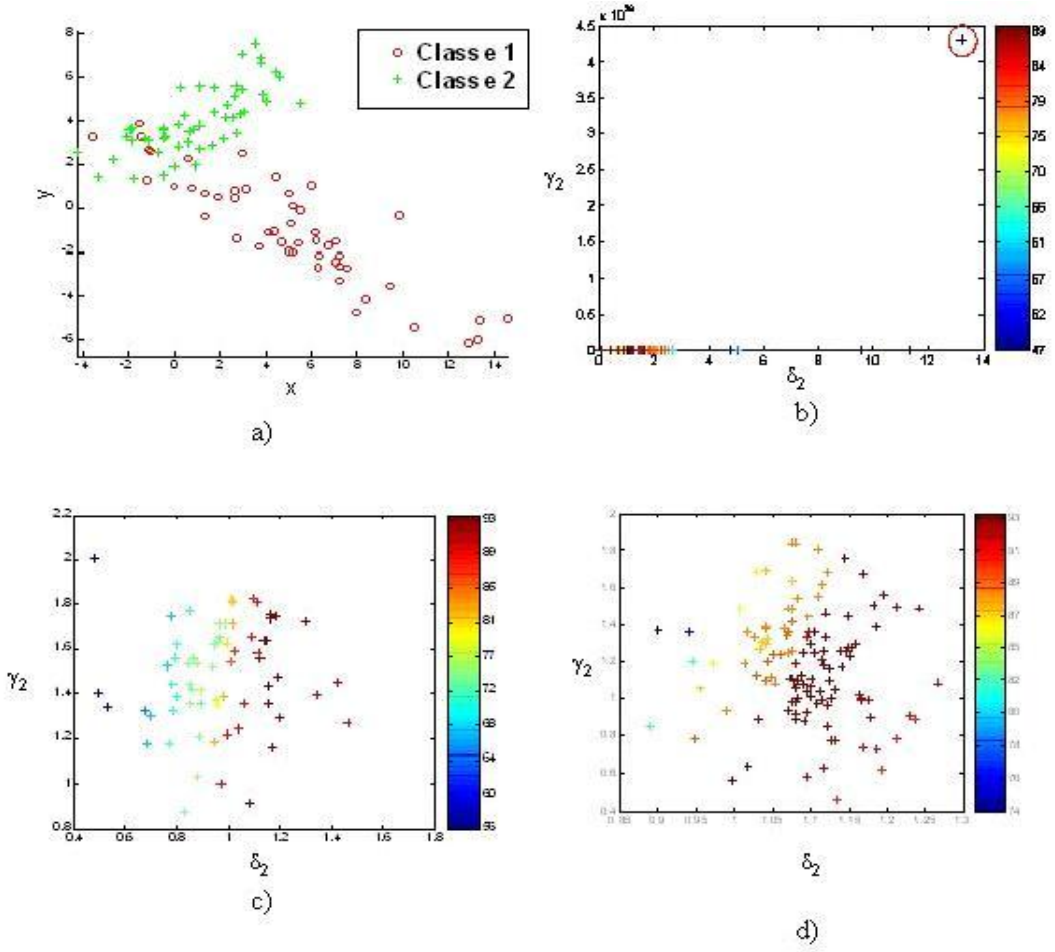


Fig. 2.14 (a) 2 classes gaussiennes non linéairement séparables avec 60 points par classe; (b) Tous les couples des paramètres (δ_2, γ_2) solution des systèmes de Cramer pour les observations mal classées avec leur taux de reconnaissance correspondant (taux maximal = 89%); (c) Tirage Monte Carlo de 60 points couples (δ_2, γ_2) dans une région délimitée par les quartiles de 25% et 75% de δ_2 et pour ces valeurs on trouve les quartiles de 25% et 75% de γ_2 alors leur taux de reconnaissance correspondant (taux maximal) = 93%; (d) Tirage Monte Carlo de 100 points couples (δ_2, γ_2) dans une région délimitée autour du voisinage du meilleur couple (δ_2, γ_2) de (c), le taux de reconnaissance varie de 74% à 93%;

Pour un problème à plus de deux classes ($C > 2$), la procédure d'optimisation est équivalente de celle à deux classe mais l'espace de recherche a $2 \times (C - 1)$ dimensions et chaque contrainte représente un hyperplan dans cet espace multidimensionnel. Pour chaque observation mal classée, il y a un ensemble de $C - 1$ contraintes (éq. 2.22). Il y a $N_m \times (N_m - 1)/2$ couples des observations mal classées. Pour chacun d'eux, on associe un système de Cramer de $2 \times (C - 1)$ équations. En résolvant ces systèmes, on obtient $N_m \times (N_m - 1)/2$ sommets. Il y a deux types d'ensembles de contraintes pour chaque observation mal classée suivant sa vraie classe d'appartenance– le premier est quand l'observation appartient à la classe ω_1, choisie comme classe de référence et le deuxième cas c'est quand l'observation appartient à toute autre classe qui est différente de la classe de référence ω_1.

Pour clarifier l'explication de l'extension de la méthode à plus de deux classes, on présente dans le tableau 2.2, pour un exemple à trois classes, l'ensemble de contraintes possibles pour chaque observation mal classée suivant l'erreur d'affection.

Tableau 2.2 : Ensemble des contraintes dans le cas d'un problème à trois classes

$e \in \omega_1$	$e \in \omega_2$	$e \in \omega_3$
$Cs(e, \omega_1) < Cs(e, \omega_2)$*	$Cs(e, \omega_2) < Cs(e, \omega_1)$*	$Cs(e, \omega_3) < Cs(e, \omega_1)$*
$Cs(e, \omega_1) < Cs(e, \omega_3)$*	$Cs(e, \omega_2) < Cs(e, \omega_3)$**	$Cs(e, \omega_3) < Cs(e, \omega_2)$**
* $\log(\gamma_i(e)) = \delta_i(e) \times \log(\mathrm{Var}(d^2(e,\omega_i))) + 2 \times \log(\overline{d^2(e,\omega_1)} - I(\omega_1)) - 2 \times \log(\overline{d^2(e,\omega_i)} - I(\omega_i)) -$ $- \log(\mathrm{Var}(d^2(e,\omega_1)))$, pour $i = 2,....,C$		
** $\log \gamma_j(e) - \log \gamma_l(e) - \delta_j(e) \times \log(\mathrm{Var}(d^2(e,\omega_j))) + \delta_l(e) \times \log(\mathrm{Var}(d^2(e,\omega_l))) =$ $= 2 \times \log(\overline{d^2(e,\omega_l)} - I(\omega_l)) - 2 \times \log(\overline{d^2(e,\omega_j)} - I(\omega_j))$, pour $l \neq j = 2,...,C$.		

En résumé, les étapes de la procédure d'optimisation sont les suivantes :

- Limitation du domaine de recherche pour les paramètres solutions des systèmes de Cramer (δ_c et γ_c où $c = 2$ à C). On calcule les quartiles de la distribution de 25% et 75% de δ_c et pour ces valeurs on trouve les valeurs de γ_c correspondantes et on calcule leurs quartiles de la distribution de 25% et 75%.
- Simulation de Monte Carlo : évaluation des règles de décision par un choix aléatoire des couples (δ_c, γ_c) dans le domaine borné de recherche.
- Choix de la meilleure configuration qui minimise le taux d'erreur pour l'ensemble d'apprentissage.

La procédure d'optimisation présentée ci-dessus est assez lente lorsque le nombre des classes augmente. La quantité de paramètres augmente linéairement avec le nombre de classes, donc la dimension des systèmes de Cramer et de l'espace de recherche augmente de la même manière. L'effectif de simulations Monté Carlo devrait croître exponentiellement avec le nombre de paramètres. Cependant pour limiter le temps de calcul nous avons choisi une évolution linéaire de la dimension de l'espace des paramètres.

Aussi, une approche alternative sera proposée dans le paragraphe suivant qui consiste à reprendre les procédures d'optimisation utilisées dans les classifieurs « SVM ». L'équation 2.22 peut être interprétée comme une règle de décision linéaire après recodage des observations comme cela sera expliqué dans le paragraphe suivant. Les motivations de cette approche concernent d'une part les problèmes de temps de calcul avec la procédure Monté Carlo et d'autre part la désignation d'une classe de référence conduisant à une méthode sous-optimale.

2.4. Classement par machines à vecteurs de support

2.4.1. Procédure de recodage

On considère un problème de classement à deux classes (ω_1, ω_2) d'un ensemble X de N objets, $X = \{o_i, i = 1,...,N\}$. L'idée est de proposer une nouvelle représentation des observations qui doit être compatible avec une règle de décision linéaire dans un nouvel espace des caractéristiques en dimensions réduites. En effet reprenons la règle de décision utilisant le « Coefficient de forme » (2.21). Les quantités $Cs(o_i, \omega_1)$ et $Cs(o_i, \omega_2)$ sont des quantités positives et nous pouvons les transformer par la fonction logarithmique, pour linéariser l'équation dans l'espace des paramètres :

$$\begin{aligned} &\log\left(\frac{\gamma_1}{\gamma_2}\right) + 2\log(\overline{d^2(o_i,\omega_1)} - I(\omega_1)) - 2\log(\overline{d^2(o_i,\omega_2)} - I(\omega_2)) \\ &- \delta_1 \log(\mathrm{Var}(d^2(o_i,\omega_1))) + \delta_2 \log(\mathrm{Var}(d^2(o_i,\omega_2))) \begin{cases} < 0 \text{ si } o_i \in \omega_1 \\ > 0 \text{ si } o_i \in \omega_2 \end{cases} \end{aligned} \tag{2.23}$$

Il s'agit en fait, d'une règle de décision linéaire dans un espace vectoriel à quatre dimensions. Ainsi selon l'équation (2.23), nous pouvons représenter chaque objet o_i par un vecteur de caractéristiques $\mathbf{x}_i$ à quatre dimensions ($\mathbf{x}_i = [x_{i1}\, x_{i2}\, x_{i3}\, x_{i4}]^T$), en adoptant le recodage ci-dessous :

$$\begin{aligned} x_{i1} &= 2 \times \log\left(\overline{d^2(o_i,\omega_1)} - I(\omega_1)\right) & x_{i3} &= -\log\left(\mathrm{Var}\left(d^2(o_i,\omega_1)\right)\right) \\ x_{i2} &= -2 \times \log\left(\overline{d^2(o_i,\omega_2)} - I(\omega_2)\right) & x_{i4} &= \log\left(\mathrm{Var}\left(d^2(o_i,\omega_2)\right)\right). \end{aligned} \tag{2.24}$$

Alors la version recodée de la règle de décision (2.23) avec deux paramètres d'ajustement (γ, δ) pour chaque classe sera:

$$\log\left(\frac{\gamma_1}{\gamma_2}\right) + x_1 + x_2 + \delta_1.x_3 + \delta_2.x_4 \begin{cases} < 0 \text{ si } o_i \in \omega_1 \\ > 0 \text{ si } o_i \in \omega_2 \end{cases}, \tag{2.25}$$

ou bien, en adoptant une notation vectorielle :

$$\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 0, \tag{2.26}$$

avec $\boldsymbol{\beta}$ = [1 1 δ_1 δ_2]^T, vecteur normal à l'hyperplan séparateur et $\beta_0 = \log\left(\gamma_1 / \gamma_2\right)$ le biais de positionnement de l'hyperplan par rapport à l'origine. Si nous étiquetons les objets pour chaque classe, y_i = -1 quand $o_i \in \omega_1$ et y_i = 1 quand $o_i \in \omega_2$, alors nous obtenons la décision linéaire classique suivante :

$$\hat{y}_i = sign(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0)\,. \tag{2.27}$$

Il s'agit de la règle classique de décision du classifieur à vecteurs de support sauf qu'ici le vecteur $\boldsymbol{\beta}$ normal à l'hyperplan optimal, est contraint d'avoir deux composantes identiques: $\beta_1 = \beta_2$ sur les quatre possibles.

Nous avons choisi d'utiliser la procédure d'optimisation utilisée dans les SVM, puisque la théorie de ce classifieur ne suppose des propriétés particulières des distributions de points. Les observations que l'on a, sont issues de données de dissimilitudes, recodées comme expliqué dans l'équation (2.24). C'est ce cadre non-paramétrique qui convenait pour traiter les données après recodage. Nous allons donc décliner la procédure d'apprentissage des SVM dans ce contexte de connaissance partielle du vecteur normal à l'hyperplan séparateur. A l'issue de l'apprentissage, on obtiendra alors les trois paramètres indépendants : δ_1, δ_2, γ_1/γ_2.

Pour ce faire, on considère les vecteurs $\mathbf{x}_i$ dans deux sous-espaces orthogonaux :

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{x}'_i \\ \mathbf{x}''_i \end{bmatrix} \text{ avec } \mathbf{x}'_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} \text{ et } \mathbf{x}''_i = \begin{bmatrix} x_{i3} \\ x_{i4} \end{bmatrix}, \tag{2.28}$$

et

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}' \\ \boldsymbol{\beta}'' \end{bmatrix} \text{ avec } \boldsymbol{\beta}' = \|\boldsymbol{\beta}'\|\mathbf{1}_{\mathrm{N}} \text{ et } \boldsymbol{\beta}'' = \begin{bmatrix} \beta_3 \\ \beta_4 \end{bmatrix}. \tag{2.29}$$

Le vecteur $\mathbf{1}_{\mathrm{N}}$ de norme 1 et : $\mathbf{1}_{\mathrm{N}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Big/ \sqrt{2}$.

Ainsi en reprenant la procédure d'optimisation utilisée pour les SVM, présentée au chapitre 1, il s'agit ici de considérer la fonctionnelle à minimiser comme une fonction à 4 inconnus : $\|\boldsymbol{\beta}'\|$, β_3, β_4 et β_0.

On présentera alors deux propositions, une à partir de la fonction à minimiser classique, utilisée pour les SVM, que l'on note *Cs*-SVM et l'autre à partir de la fonction à minimiser des SVM à moindres carrés que l'on note *Cs*-LSSVM.

2.4.2. Le classifieur *Cs* - SVM

Comme déjà mentionné au chapitre 1 et suivant la théorie des SVM, l'optimisation de la règle de décision pour un problème de classement entre deux classes non linéairement séparables, s'écrit ainsi, avec ξ_i, les variables « ressort » d'écart entre les observations et les marges :

$$\begin{aligned} &\text{Mimimiser}_{\boldsymbol{\beta},\beta_0,\xi} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{A condition que : } y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) \geq 1-\xi_i, \text{ pour } i = 1,...,N \text{ et } \xi_i \geq 0 \end{aligned} \tag{2.30}$$

Ce problème peut être réécrit dans notre cas, puisque les vecteurs se décomposent dans deux sous espaces orthogonaux :

$$\begin{aligned} &\text{Mimimiser}_{\|\boldsymbol{\beta}'\|,\boldsymbol{\beta}'',\beta_0,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{\beta}'\|^2 + \frac{1}{2}\|\boldsymbol{\beta}''\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{A condition que : } y_i(\|\boldsymbol{\beta}'\|u_i + \boldsymbol{\beta}''^T\mathbf{x}_i'' + \beta_0) \geq 1-\xi_i, \text{ pour } i = 1,...,N \text{ et } \xi_i \geq 0 \\ &\text{où } u_i = \langle \mathbf{1}_N ; \mathbf{x}_i' \rangle \end{aligned} \tag{2.31}$$

Pour chaque observation *i*, u_i est la projection dans le premier sous espace à deux dimensions sur le vecteur normé colinéaire à $\boldsymbol{\beta}'$ (ici le $\mathbf{1}_N$). En utilisant la méthode des multiplicateurs de Lagrange, la fonction L_p du Lagrangien primaire devient :

$$\begin{aligned} L_P(\|\boldsymbol{\beta}'\|,\boldsymbol{\beta}'',\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\mu}) &= \frac{1}{2}\|\boldsymbol{\beta}'\|^2 + \frac{1}{2}\|\boldsymbol{\beta}''\|^2 + C\sum_i \xi_i \\ &- \sum_{i=1}^{N}\alpha_i\{y_i(\|\boldsymbol{\beta}'\|\cdot u_i + \boldsymbol{\beta}''^T\mathbf{x}_i'' + \beta_0) - 1 + \xi_i\} - \sum_i \mu_i\xi_i, \end{aligned} \tag{2.32}$$

avec α_i les multiplicateurs de Lagrange, introduits pour prendre en compte les contraintes du classement et avec μ_i les multiplicateurs de Lagrange, introduits pour prendre en compte la positivité des variables « ressort » ξ_i.

De la même manière qu'au chapitre 1 on utilise les conditions de KKT et en ajoutant la condition reliée à la variable supplémentaire $\|\boldsymbol{\beta}'\|$:

$$\frac{\partial}{\partial\|\boldsymbol{\beta}'\|}L_P = \|\boldsymbol{\beta}'\| - \sum_{i=1}^{N_S}\alpha_i y_i u_i = 0, \tag{2.33}$$

où le N_S indique le nombre des vecteurs support. Ainsi on a :

$$\|\boldsymbol{\beta}'\| = \sum_{i=1}^{N_S}\alpha_i y_i u_i. \tag{2.34}$$

En substituant dans la fonctionnelle du Lagrangien primaire, on obtient le Lagrangien dual :

$$\begin{aligned} &\text{Minimiser } L_D = -\sum_i \alpha_i + \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j (u_i u_j + \langle \mathbf{x}_i''; \mathbf{x}_j'' \rangle) \\ &\text{sous } 0 \leq \alpha_i \leq C \end{aligned} \tag{2.35}$$

$$\sum_i \alpha_i y_i = 0$$

On peut remarquer donc que l'on aboutit à la même fonctionnelle que celle présentée au chapitre 1 (éq. (1.56)) à la différence près du terme $u_i \cdot u_j$ à rajouter au produit scalaire $\langle \mathbf{x}_i ; \mathbf{x}_j \rangle$ pour chaque couple d'observations.

Le classifieur, présenté ci-dessus a pour but de trouver l'hyperplan séparateur optimal sachant que le vecteur $\boldsymbol{\beta}$ normal à l'hyperplan optimal, est contraint d'avoir deux de ses composantes identiques: $\beta_1 = \beta_2$ et l'espace des caractéristiques de cet hyperplan est limité à quatre dimensions quelque soit la dimension intrinsèque sous-jacente des observations, et quelque soit le nombre des observations.

2.4.3. Procédure d'optimisation du classifieur *Cs* – SVM

L'apprentissage du classifieur à vecteurs de support aboutit à une procédure d'optimisation quadratique. En particulier, l'apprentissage avec un grand nombre de données devient rapidement difficile à résoudre (capacité mémoire et temps de calcul) car la taille du problème d'optimisation dépend du nombre N des objets dans l'ensemble d'apprentissage. Osuna [Osuna et al. 1997] et Joachims [Joachims 1999] ont développé une technique pour résoudre ces difficultés reliées avec les grandes bases de données. Leur approche est de décomposer le problème d'optimisation quadratique en une série de petits sous-problèmes. Le SVM^{Light} [http://svmlight.joachims.org], le logiciel d'optimisation des SVM, développé par Joachims, utilise la décomposition proposée dans [Osuna et al. 1997]. Cette décomposition consiste à diviser le problème du Lagrangien primaire en deux parties – ensemble de travail et ensemble inactif.

Ci-dessous on présentera cette technique et les modifications apportées afin d'intégrer la méthode *Cs*-SVM dans SVM^{Light}.

On utilisera une représentation plus simplifiée du problème d'optimisation de *Cs*-SVM (2.35) :

$$\begin{aligned} &\text{Minimiser} \quad -\boldsymbol{\alpha}^T \mathbf{1} + \frac{1}{2}\boldsymbol{\alpha}^T (V+Q)\boldsymbol{\alpha} \\ &\text{sous} \quad \boldsymbol{\alpha}^T \mathbf{y} = 0 \quad , \\ &\qquad 0 \le \boldsymbol{\alpha} \le C\mathbf{1} \end{aligned} \tag{2.36}$$

où $V(i,j) = y_i y_j u_i u_j$ et $Q(i,j) = y_i y_j \langle \mathbf{x}_i ; \mathbf{x}_j \rangle$ sont des matrices des produits scalaires. Les deux sont définies positives et les contraintes de (2.36) sont des contraintes linéaires alors le problème d'optimisation (2.36) est un problème d'optimisation convexe. Pour ce type de problème les conditions de KKT suivantes sont nécessaires et suffisantes pour atteindre une solution optimale.

On note le multiplicateur Lagrangien pour la contrainte linéaire de (2.36) avec λ^{eq} et les multiplicateurs Lagrangiens pour les limites inférieure et supérieure de la contrainte

conditionnelle de (2.36) avec $\lambda^{\inf}$ et $\lambda^{\sup}$, $\boldsymbol{\alpha}$ est optimal pour (2.36) s'ils existent tels λ^{eq}, $\lambda^{\inf}$ et $\lambda^{\sup}$ que les conditions de KKT sont satisfaites :

$$g(\boldsymbol{\alpha}) + \left(\lambda^{\text{eq}}\mathbf{y} - \lambda^{\inf} + \lambda^{\sup}\right) = \mathbf{0} \tag{2.37}$$

$$\forall i : \lambda_i^{\inf}\left(-\alpha_i\right) = 0 \tag{2.38}$$

$$\forall i : \lambda_i^{\sup}\left(\alpha_i - C\right) = 0 \tag{2.39}$$

$$\lambda^{\inf} \geq \mathbf{0} \tag{2.40}$$

$$\lambda^{\sup} \geq \mathbf{0} \tag{2.41}$$

$$\boldsymbol{\alpha}^T\mathbf{y} = 0 \tag{2.42}$$

$$0 \leq \boldsymbol{\alpha} \leq C\mathbf{1}\ , \tag{2.43}$$

où $g(\boldsymbol{\alpha})$ est le vecteur des dérivées partielles de $\boldsymbol{\alpha}$ et pour (2.36) :

$$g(\boldsymbol{\alpha}) = -\mathbf{1} + (V + Q)\boldsymbol{\alpha}\,. \tag{2.44}$$

Suivant l'algorithme de décomposition de l'ensemble d'apprentissage, proposé par Osuna et Joachims, dans chaque itération nous divisons l'ensemble des multiplicateurs de Lagrange α_i en deux catégories :

- Un ensemble **B** des multiplicateurs de Lagrange libres
- Un ensemble **N** des multiplicateurs de Lagrange fixes

Les multiplicateurs de Lagrange libres sont ceux qui peuvent être mis à jour durant l'itération courante, tandis que les multiplicateurs de Lagrange fixes sont temporairement fixés à une valeur particulière. L'ensemble des multiplicateurs libres sera également appelé l'ensemble de travail. L'ensemble de travail a une taille constante q beaucoup plus petite que N.

Si les conditions KKT sont violées alors l'algorithme décompose le problème (2.36) en deux ensembles – l'ensemble de travail **B** et l'ensemble des multiplicateurs fixes **N**. On arrange les **y**, Q et V correctement par rapport à **B** et **N** et on décompose aussi $\boldsymbol{\alpha}$ en $\boldsymbol{\alpha}_{\mathbf{B}}$ et $\boldsymbol{\alpha}_{\mathbf{N}} = 0$, de sorte que :

$$\boldsymbol{\alpha} = \begin{vmatrix}\boldsymbol{\alpha}_{\mathbf{B}} \\ \boldsymbol{\alpha}_{\mathbf{N}}\end{vmatrix} \tag{2.45}$$

$$\mathbf{y} = \begin{vmatrix}\mathbf{y}_{\mathbf{B}} \\ \mathbf{y}_{\mathbf{N}}\end{vmatrix} \tag{2.46}$$

$$Q = \begin{vmatrix}Q_{\mathbf{BB}} & Q_{\mathbf{BN}} \\ Q_{\mathbf{NB}} & Q_{\mathbf{NN}}\end{vmatrix} \tag{2.47}$$

$$V = \begin{vmatrix}V_{\mathbf{BB}} & V_{\mathbf{BN}} \\ V_{\mathbf{NB}} & V_{\mathbf{NN}}\end{vmatrix} \tag{2.48}$$

Parce que Q et V sont symétriques ($Q_{\mathbf{BN}} = Q_{\mathbf{NB}}^T$ et $V_{\mathbf{BN}} = V_{\mathbf{NB}}^T$), nous pouvons réécrire (2.36):

$$\begin{aligned} \text{Minimiser} \quad & -\boldsymbol{\alpha}_{\mathbf{B}}^T\left(\vec{\mathbf{1}} - \left(Q_{\mathbf{BN}} + V_{\mathbf{BN}}\right)\boldsymbol{\alpha}_{\mathbf{N}}\right) + \frac{1}{2}\boldsymbol{\alpha}_{\mathbf{B}}^T\left(Q_{\mathbf{BB}} + V_{\mathbf{BB}}\right)\boldsymbol{\alpha}_{\mathbf{B}} \\ \text{sous} \quad & \boldsymbol{\alpha}_{\mathbf{B}}^T\mathbf{y}_{\mathbf{B}} + \boldsymbol{\alpha}_{\mathbf{N}}^T\mathbf{y}_{\mathbf{N}} = 0 \\ & 0 \le \alpha_i \le C \;\forall i \end{aligned} \tag{2.49}$$

Parce que les des multiplicateurs de Lagrange de l'ensemble **N** sont fixés alors les termes $\frac{1}{2}\boldsymbol{\alpha}_{\mathbf{N}}^T Q_{\mathbf{NN}}\boldsymbol{\alpha}_{\mathbf{N}}$ et $\frac{1}{2}\boldsymbol{\alpha}_{\mathbf{N}}^T V_{\mathbf{NN}}\boldsymbol{\alpha}_{\mathbf{N}}$ et $-\boldsymbol{\alpha}_{\mathbf{N}}^T\mathbf{1}$ sont constants, ils peuvent être supprimés, sans changer la solution du problème. C'est un sous-problème de programmation quadratique semi-défini positif qui est suffisamment petit pour être résolu par la plupart des méthodes existantes. Nous pouvons remplacer $\alpha_i = 0$, $i \in \mathbf{B}$, avec $\alpha_j = 0$, $j \in \mathbf{N}$, sans changer la fonction de coût ni la faisabilité du sous-problème (2.49) et du problème d'origine (2.36). Le déplacement d'une variable qui viole les conditions d'optimalité de **N** à **B** donne une amélioration stricte dans la fonction de coût lorsque le sous-problème est optimisé de nouveau. Selon la proposition précédente, cet algorithme améliorera strictement la fonction objective à chaque itération sans tomber dans une boucle infinie. Puisque la fonction objective est bornée (elle est convexe quadratique et la région réalisable est bornée), l'algorithme doit converger vers la solution globale optimale en un nombre fini d'itérations. Les preuves de ces proposions sont détaillées dans [Osuna et al. 1997].

Le progrès rapide de l'algorithme dépend fortement de savoir s'il peut choisir des bons ensembles de travail. Lors de la sélection d'ensemble de travail, il est souhaitable de sélectionner un ensemble de multiplicateurs de Lagrange tels que l'itération courante fera faire des évolutions significatives vers le minimum de L_D. En utilisant une stratégie, basée sur la méthode de Zoutendijk [Zoutendijk 1970], l'idée est de trouver la direction de descente la plus raide $\boldsymbol{\upsilon}$ qui a seulement q éléments non négatifs. Les variables qui correspondent à ces éléments feront partie de l'ensemble de travail courant. Cette approche conduit à un nouveau problème d'optimisation :

$$\text{Minimiser} \quad g\left(\boldsymbol{\alpha}^{(t)}\right)^T\boldsymbol{\upsilon} \tag{2.50}$$

$$\text{sous} \quad \mathbf{y}^T\boldsymbol{\upsilon} = 0 \tag{2.51}$$

$$\upsilon_i \ge 0 \text{ pour } i : \alpha_i = 0 \tag{2.52}$$

$$\upsilon_i \le 0 \text{ pour } i : \alpha_i = C \tag{2.53}$$

$$-\mathbf{1} \le \boldsymbol{\upsilon} \le \mathbf{1} \tag{2.54}$$

$$\left|\left\{\upsilon_i : \upsilon_i \ne 0\right\}\right| = q\,. \tag{2.55}$$

La fonction objective (2.50) indique quelle direction de descente est désirée. Une direction de descente a un produit scalaire négatif avec le vecteur des dérivées partielles $g\left(\boldsymbol{\alpha}^{(t)}\right)$ au point courant $\boldsymbol{\alpha}^{(t)}$. Les contraintes (2.51), (2.52) et (2.53) s'assurent que la direction de descente est projetée le long de la contrainte d'égalité (2.36) et obéit aux contraintes. La contrainte (2.54) normalise le vecteur de descente afin de bien poser le problème d'optimisation. Enfin, la dernière contrainte (2.55) indique que la direction de descente ne doit impliquer que q variables. Les

variables avec des valeurs $\upsilon_i \neq 0$ sont inclus dans l'ensemble de travail **B**. De cette façon, nous choisissons l'ensemble de travail avec la plus rapide évolution de décroissance.

Il a deux conditions requises pour l'algorithme proposé ci - dessus :

- l'algorithme se termine seulement si on a retrouvé une solution optimale ;
- si non alors on arrive à une nouvelle itération vers la solution optimale.

La première condition est remplie par vérification des conditions (nécessaires et suffisantes) optimales (2.37) et (2.43) pour chaque itération. Pour la deuxième condition on suppose que $\boldsymbol{\alpha}^{(t)}$ courant est non optimal, alors la stratégie de sélection de l'ensemble de travail revient au problème d'optimisation de type (2.49). Puisque par construction de ce problème d'optimisation, il existe un $\boldsymbol{\upsilon}$ qui est une direction possible de descente, on sait suivant [Joachims, 1998] que le problème courant n'est pas optimal. Alors l'optimisation de (2.49) va conduire vers une valeur inférieure de la fonction objective de ce problème. Puisque la solution pour (2.49) est aussi une solution possible pour (2.36) à cause de la décomposition proposée, on a aussi une valeur inférieure pour (2.36). Cela vaut dire que l'on a une descente stricte de la fonction objective de (2.36) à chaque itération.

L'algorithme fonctionne ainsi:
Tandis que les conditions d'optimalité KKT (équations de (2.37) à (2.43)) sont violées

- Sélectionner les q variables pour l'ensemble de travail **B**. Le reste $(N - q)$ variables sont fixées à leur valeur courante.
- Décomposer le problème et résoudre le sous-problème quadratique : optimiser L_D sur **B**.
- Terminer et renvoyer $\boldsymbol{\alpha}$.

Alors les quantités nécessaires pour le fonctionnement de cet algorithme pour chaque itération sont :

- Les vecteurs des dérivées partielles $g(\boldsymbol{\alpha})$ pour l'ensemble de travail courant ;
- Les critères de fin – si (2.50) devient 0 alors (2.36) est résolu pour $\boldsymbol{\alpha}^{(t)}$ courant comme solution. SVM^{Light} utilise aussi des critères de fin, dérivés des conditions (2.37) à (2.43) ;
- Les matrices $Q_{\mathbf{BB}}$, $Q_{\mathbf{BN}}$, $V_{\mathbf{BB}}$ et $V_{\mathbf{BN}}$.

L'algorithme d'optimisation, présenté ci-dessus, offre la possibilité de travailler avec des grandes bases de données. Osuna prouve qu'il converge vers une solution optimale et il est implémenté dans le logiciel SVM^{Light} qu'on utilise dans ce travail avec les modifications présentées ci-dessus afin d'intégrer l'algorithme du « Coefficient de forme ».

2.4.4. Procédure d'optimisation du classifieur *Cs* – LSSVM

En utilisant le raisonnement des LS-SVM on peut réécrire le problème d'optimisation présenté dans le chapitre 1 par :

$$\begin{aligned}
&\text{Mimimiser}_{\|\boldsymbol{\beta}'\|,\boldsymbol{\beta}'',\beta_0,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{\beta}'\|^2+\frac{1}{2}\|\boldsymbol{\beta}''\|^2+\frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\
&\text{A condition que : } y_i(\|\boldsymbol{\beta}'\|u_i+\boldsymbol{\beta}''^T\cdot\mathbf{x}_i''+\beta_0)=1-\xi_i,\ \text{pour } i=1,...,N \text{ et } \xi_i\geq 0 \\
&\text{où } u_i=\langle \mathbf{1}_N;\mathbf{x}_i'\rangle
\end{aligned} \tag{2.56}$$

On appellera ce classifieur *Cs*-LSSVM.

A partir de l'équation (2.56), la formulation du problème d'optimisation en utilisant les multiplieurs de Lagrange s'écrit ainsi :

$$\begin{aligned}
&L_P\left(\|\boldsymbol{\beta}'\|,\boldsymbol{\beta}'',\boldsymbol{\xi},\boldsymbol{\alpha}\right)=\frac{1}{2}\|\boldsymbol{\beta}'\|^2+\frac{1}{2}\|\boldsymbol{\beta}''\|^2+\frac{C}{2}\sum_i\xi_i^{\,2} \\
&-\sum_{i=1}^{N}\alpha_i\{y_i\left(\|\boldsymbol{\beta}'\|u_i+\boldsymbol{\beta}''^T\mathbf{x}_i''+\beta_0\right)-1+\xi_i\}.
\end{aligned} \tag{2.57}$$

Les conditions de Karush–Kuhn–Tucker seront alors les suivantes :

$$\begin{aligned}
&\frac{\partial}{\partial\boldsymbol{\beta}''}L_P=\boldsymbol{\beta}''-\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i''=0\Rightarrow\boldsymbol{\beta}''=\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i'' \\
&\frac{\partial}{\partial\|\boldsymbol{\beta}'\|}L_P=\|\boldsymbol{\beta}'\|-\sum_{i=1}^{N}\alpha_i y_i u_i=0\Rightarrow\|\boldsymbol{\beta}'\|=\sum_{i=1}^{N}\alpha_i y_i u_i \\
&\frac{\partial}{\partial\beta_0}L_P=-\sum_{i=1}^{N}\alpha_i y_i=0 \\
&\frac{\partial}{\partial\xi}L_P=C\xi_i-\alpha_i=0 \\
&\frac{\partial}{\partial\alpha}L_P=y_i\left(\|\boldsymbol{\beta}'\|u_i+\boldsymbol{\beta}''^T\mathbf{x}_i''+\beta_0\right)-1+\xi_i=0
\end{aligned} \tag{2.58}$$

Ainsi le système linéaire à résoudre peut alors s'écrire sous la forme suivante :

$$\begin{bmatrix}
1 & \mathbf{0}_{1\times2} & 0 & \mathbf{0}_{1\times N} & -W^T \\
\mathbf{0}_{2\times1} & \mathrm{I}_{2\times2} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times N} & -Z^T \\
0 & \mathbf{0}_{1\times2} & 0 & \mathbf{0}_{1\times N} & -Y^T \\
\mathbf{0}_{N\times1} & \mathbf{0}_{N\times2} & \mathbf{0}_{N\times1} & CI_{N\times N} & -I_{N\times N} \\
W & Z & Y & I_{N\times N} & \mathbf{0}_{N\times N}
\end{bmatrix}
\begin{bmatrix}
\|\boldsymbol{\beta}'\| \\ \boldsymbol{\beta}'' \\ \beta_0 \\ \boldsymbol{\xi} \\ \boldsymbol{\alpha}
\end{bmatrix}=
\begin{bmatrix}
0 \\ \mathbf{0}_{2\times1} \\ 0 \\ \mathbf{0}_{N\times1} \\ \mathbf{1}_{N\times1}
\end{bmatrix}, \tag{2.59}$$

avec $Z=[y_1\mathbf{x}_1''^T,\ldots,y_N\mathbf{x}_N''^T]^T$, $W=[y_1u_1,\ldots,y_Nu_N]^T$, $Y=[y_1,\ldots,y_N]^T$.
Comme on a déjà indiqué au chapitre 1, tous les points de l'ensemble d'apprentissage sont des

vecteurs de support avec des valeurs des Lagrangiens plus ou moins grandes. La distribution des points avec différentes valeurs des Lagrangiens est repartie partout dans l'espace des caractéristiques et pas comme avec les SVM standard, autour de la frontière de décision de l'hyperplan séparateur optimal. Parce que tous les points deviennent des vecteurs de support si le problème de classement est gros, on a certaines difficultés le calcul sous Matlab mais le temps de calcul reste satisfaisant.

2.5. Synthèse sur des résultats des données artificielles

Le comportement des méthodes d'optimisation pour les règles de décision avec le « Coefficient de forme » *Cs* présentées ci-dessus, sera comparé sur des bases de données artificielles (distributions gaussiennes 2D) afin d'en extraire une première synthèse. Les résultats sur des données réelles seront présentés dans le chapitre 4.

L'évaluation de la performance du classifieur est réalisée à partir du nombre d'observations de l'ensemble de test qui sont incorrectement classées. Ce taux d'erreur sera estimé par validation croisée (« Leave-One-Out » ou « k-fold ») pour les expérimentations sur les données réelles (chapitre 4) et par « apprentissage-test » pour les expérimentations sur les bases de données artificielles. Dans ce cas là, on génère des bases de données pour l'apprentissage et des bases de données pour le test. Les points pour l'apprentissage ne sont jamais inclus dans les bases de test et vise versa.

L'approche générale est rappelée à la figure 2.15.

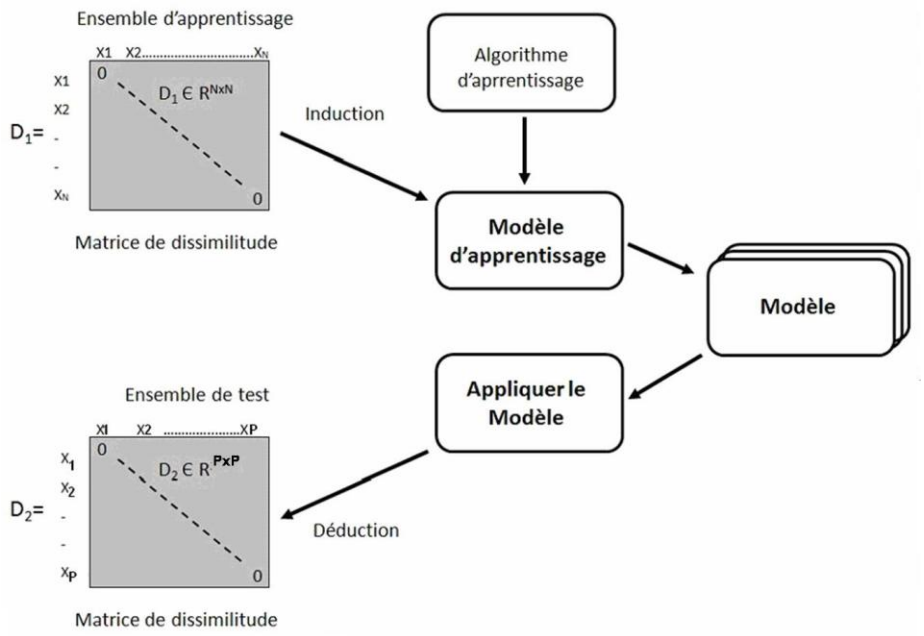


Fig. 2.15 : Approche générale pour l'apprentissage et le test d'un modèle de classement

Pour toutes les expériences, deux environnements différents de programmation ont été nécessaires. Pour les méthodes appelées « *Cs*-SVM », nous avons développé en langage C, en apportant des modifications au logiciel « SVM^{Light} » [http://svmlight.joachims.org], version 6.02, écrit en C par Thorsten Joachims. Pour toutes les autres méthodes, le code a été écrit en utilisant Matlab® avec la bibliothèque « PRTools 4.0» [www.prtools.org], développé par le groupe « Pattern Recognition » de l'Université de Delft, et la bibliothèque « LS-SVMLab 1.5 », développée par l'équipe de J.A.K. Suykens de l'Université K.U.Leuven.

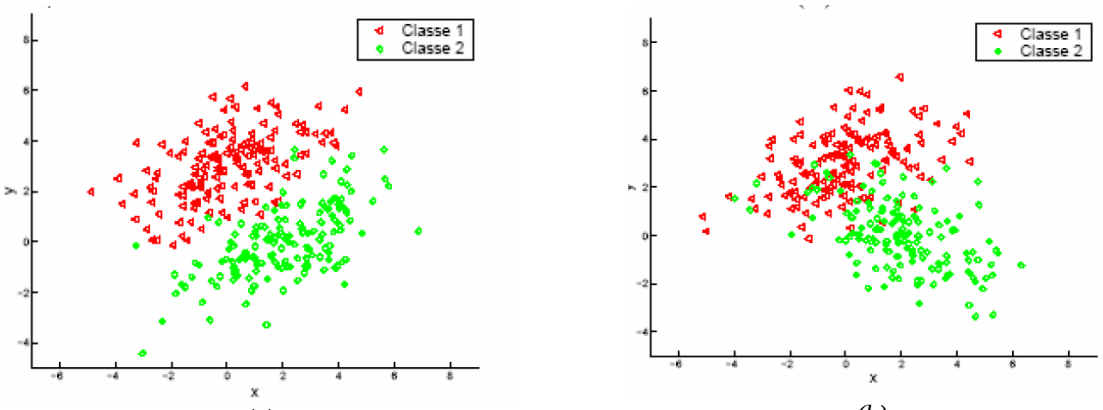
On génère des bases de données artificielles. Chaque classe est issue d'un tirage d'une loi normale de moyenne et de matrice de variance - covariance parfaitement connues. On se place dans deux cas différents selon si la matrice de variance - covariance est commune aux deux classes (cas 1) ou différente pour les deux (cas 2). Les bases de données sont générées en deux dimensions. On peut voir l'allure des nuages de points aux figures 2.16a. et 2.16b. Pour l'apprentissage, les bases de données contiennent 100 observations par classe et pour le test, elles contiennent 200 observations par classe. Pour chaque expérimentation, dix ensembles d'apprentissage et un ensemble de test sont générés. Les distances euclidiennes sont calculées pour toutes observations. À la suite de la procédure d'apprentissage, les paramètres d'ajustement sont calculés pour chaque ensemble d'apprentissage et le couple minimisant le taux d'erreur pour cette base d'apprentissage est choisi et ensuite ce couple de paramètres est utilisé pour le test. Cette procédure est répétée pour toutes les bases d'apprentissage.

Cas 1 : La matrice de variance - covariance des classes ω_1 et ω_2 est identique pour les deux classes, les moyennes sont différentes (voir fig. 2.16a.).

$$\begin{aligned}(\mu_1\ \mu_2) &= \begin{pmatrix} 0 & 2 \\ 3 & 0 \end{pmatrix} \\ \Sigma &= \begin{pmatrix} 3.25 & 1.3 \\ 1.3 & 1.75 \end{pmatrix}\end{aligned} \tag{2.60}$$

Cas 2 : Les matrices de variance - covariance et les moyennes pour les deux classes sont les suivantes (voir fig. 2.16b.) :

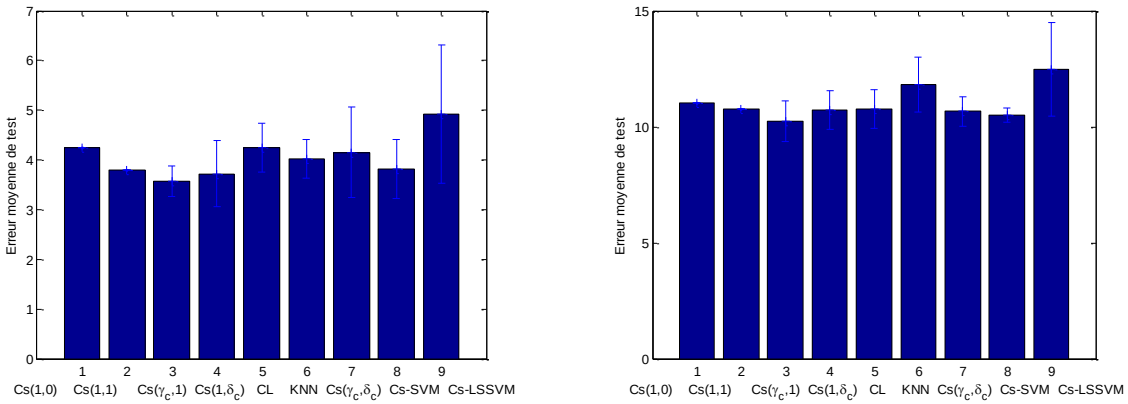
$$\begin{aligned}(\mu_1\ \mu_2) &= \begin{pmatrix} 0 & 2 \\ 3 & 0 \end{pmatrix} \\ \Sigma_1 &= \begin{pmatrix} 3.25 & 1.3 \\ 1.3 & 1.75 \end{pmatrix} \\ \Sigma_2 &= \begin{pmatrix} 3.25 & -1.3 \\ -1.3 & 1.75 \end{pmatrix}\end{aligned} \tag{2.61}$$



(a) *(b)*

Fig. 2.16 Données artificielles pour deux classes suivant des distributions gaussiennes, exemple de la base de test à 200 points par classe ; (a) cas 1 et (b) cas 2

Pour les expériences menées sur ces deux cas, on comparera les taux d'erreur de test en utilisant les différents modèles du « Coefficient de forme », le classifieur des K plus proches voisins et le classifier linéaire de Pekalska (1.18) dont l'ensemble de prototypes est choisi suivant la procédure *KCenters*. Ce dernier classifieur est noté « CL ». La figure 2.17 illustre les taux d'erreur de test pour les Cas 1 et 2 pour les différents classifieurs.



(a) *(b)*

Fig. 2.17 Moyenne et variance des taux d'erreur de test pour les modèles Cs(1,0), Cs(1,1), Cs(γ_c,1), Cs(1,δ_c), Cs(γ_c, δ_c), KNN, classifieur linéaire (CL), Cs-SVM et Cs-LSSVM a) Cas 1 ; b) Cas 2.

Notons que pour les modèles *Cs*(1,0) et *Cs*(1,1) qui sont sans paramètres à ajuster, il n'y à qu'une seule estimation sur la base de test, puisqu'il n'y a pas d'apprentissage. Afin de pouvoir comparer la performance des modèles Cs-SVM et Cs-LSSVM avec le modèle Cs géométrique sur la fig 2.17a.b on a fixé une des deux classes comme classe de référence ($\delta_1 = 1$ et $\gamma_1 = 1$ alors $\boldsymbol{\beta} = [1\ 1\ 1\ \delta_2]$ et $\beta_0 = \log(1/\gamma_2) = -\log(\gamma_2)$).

De plus, on a calculé l'erreur théorique de classement dans le Cas 1 car les matrices de variance - covariance pour chaque classe sont égales. On utilise la formule de [Celeux&Turlot 1989] qui donne le taux d'erreur de classement d'un élément de la classe ω_1 dans la classe ω_2 :

$$Err[\omega_2/\omega_1] = \Phi\left(-\frac{D_{\omega_1\omega_2}}{2}\right) + \frac{\log\left(p_{\omega_2}/p_{\omega_1}\right)}{D_{\omega_1\omega_2}}, \tag{2.62}$$

où Φ représente la fonction de répartition d'une loi normale centrée réduite, $D_{\omega_1\omega_2}$ est la distance de Mahalanobis de la classe ω_1 à la classe ω_2 et les p_{ω_1} et p_{ω_2}, les probabilités a priori des deux classes (pour l'exemple présenté dans ce chapitre, les classes sont équiprobables).

Dans notre cas l'erreur théorique en test calculée est de 3.65. On peut déduire suivant les résultats présentés à la figure 2.17a. que la moyenne du taux d'erreur du *Cs*-SVM (3.825%) s'approche beaucoup de cette erreur théorique.

Concernant le classifieur linéaire de Pekalska (présenté au paragraphe 1.4.2), on sait que le choix de l'ensemble de prototypes est primordial pour le classement [Pekalska et al. 2001]. Ce classifieur linéaire dans l'espace de dissimilitudes s'interprète comme une combinaison linéaire pondérée des dissimilitudes entre une observation et cet ensemble de prototypes. Un exemple est illustré sur la figure 2.18a.b avec un choix aléatoire d'un prototype par classe.

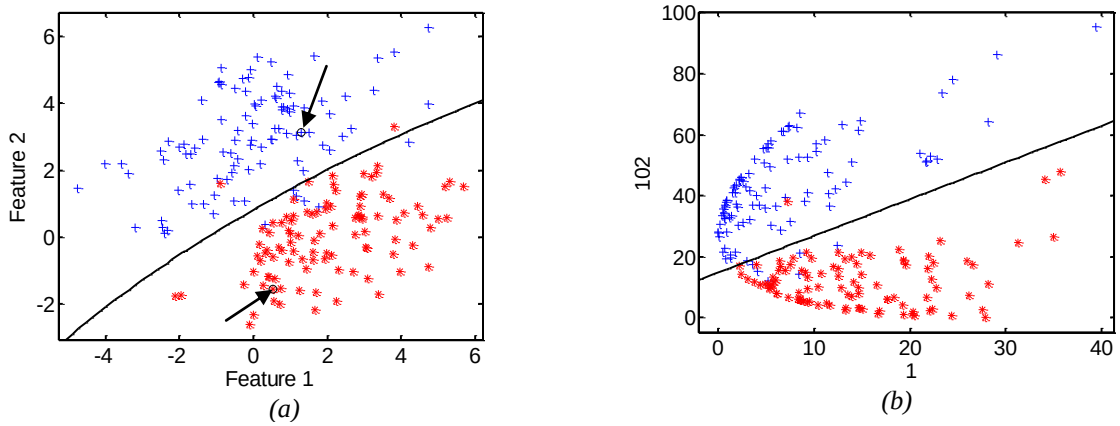


Fig. 2.18 Exemples de frontières obtenues avec (a) le classifieur quadratique (taux d'erreur de 4%), les deux observations choisies comme l'ensemble de représentation R = [1 ; 102] sont marquées avec un cercle; (b) le classifieur linéaire dans l'espace des dissimilitudes $D(\cdot, R)$ *(taux d'erreur de 3.5%).*

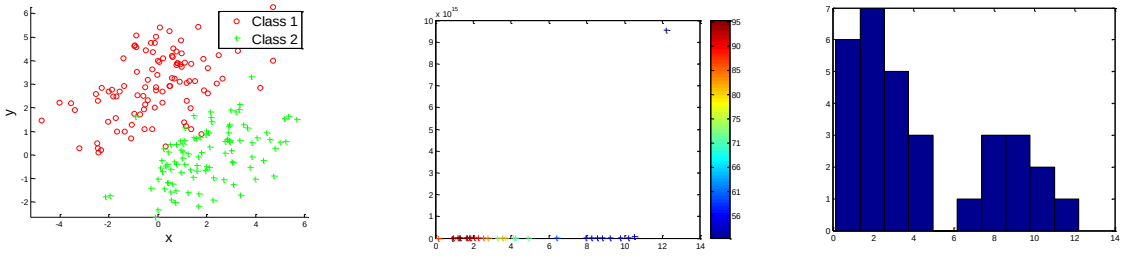
Pekalska et ses collègues proposent plusieurs méthodes de choix de cet ensemble de représentation [Pekalska et al. 2006 ; Pekalska et al. 2002]. Pour notre étude on utilise la procédure *KCenters* pour choisir des représentants pour chaque classe. Comme les probabilités a priori des classes sont égales, on choisit le même nombre de prototypes pour les deux classes. Suivant [Pekalska 2005] le choix de l'ensemble initial pour cette procédure est très important pour les résultats de classement. Afin de déterminer l'ensemble initial des prototypes de chaque classe pour cette procédure, nous partons du point plus proche du centre de gravité de la classe et progressivement d'autres prototypes sont ajoutés. L'ensemble est agrandi en divisant l'ensemble avec le plus grand rayon en deux et en remplaçant son centre par deux autres membres du group. La boucle est arrêtée lorsque le nombre désiré de *k* centres est déterminé. Ce choix est justifié dans le cadre des distributions gaussiennes car cette procédure distribue les prototypes uniformément sur les classes d'une manière spatiale reliée à l'information des dissimilitudes.

Suivant cette procédure on a choisi 4 prototypes par classe. Les résultats du classement sur les bases gaussiennes confirment les propos de Pekalska, à savoir que le classifieur linéaire est performant pour des petits ensembles de représentation et généralise mieux que le classifieur KNN. De plus, en raison de la relation linéaire entre la matrice de distances euclidiennes au carré et la matrice des produits scalaires, ce classifieur linéaire construit à partir des distances au carré est en fait, un classificateur quadratique dans l'espace sous-jacent des prototypes. Comme les distributions utilisées sont gaussiennes, il est donc également optimal.

Des expériences avec un différent nombre de prototypes ont été menées (2, 3, 4 et 10 prototypes par classe). On a constaté que la moyenne et la variance du taux d'erreur de test restent très semblables pour tous les cas.

Pour le classifieur aux K plus proches voisins, durant la phase d'apprentissage on optimise le nombre K de voisins par estimation en « Leave One Out » sur chaque base d'apprentissage. La performance de ce classifieur est très bonne pour les distributions du Cas 1. Cela est dû au fait que la distance euclidienne est métrique (inégalité triangulaire respectée) et que les deux classes sont faiblement mélangées. Cette dernière caractéristique n'est plus valable pour le Cas 2 et la performance du KNN est moins bonne. La règle du KNN étant basée sur un voisinage local, elle est donc sensible aux observations bruitées et la variance du taux d'erreur est plus importante.

Concernant l'approche géométrique du « Coefficient de forme » (décrite au paragraphe 2.3), l'apprentissage consiste à estimer deux paramètres (γ_2, δ_2) puisqu'il n'y a que deux classes. Les figures 2.19a-i illustrent les résultats sur un apprentissage dans le Cas 1. A la figure 2.19b. sont représentés tous les couples solutions (γ_2, δ_2) des systèmes de Cramer, tels que δ_2 (exposant de la variance au dénominateur dans l'équation du coefficient de forme) soit strictement positif (le taux de reconnaissance associé est indiqué par un code de couleurs). On remarque d'une part, la présence de valeurs aberrantes pour γ_2 ; cela se produit quand le système de Cramer est indéterminé. D'autre part, la distribution des valeurs pour δ_2 est bimodale (fig. 2.19c.), et le mode inférieur correspondant aux plus faibles valeurs regroupe les configurations à taux de reconnaissance élevé. Comme expliqué dans le paragraphe 2.3, le domaine de recherche des paramètres δ_2 et γ_2 est ensuite restreint entre les quartiles à 25% et 75% pour les deux distributions marginales (fig. 2.19d. et 2.19e.). Un total de 1000 simulations de Monte Carlo est alors réalisé. La figure 2.19.g illustre la distribution conjointe des paramètres pris pour ces simulations. Un code couleur indique le taux de reconnaissance associé. Le meilleur couple des paramètres (γ_2, δ_2) correspond à une valeur maximale du taux de reconnaissance.



(a) *(b)* *(c)*

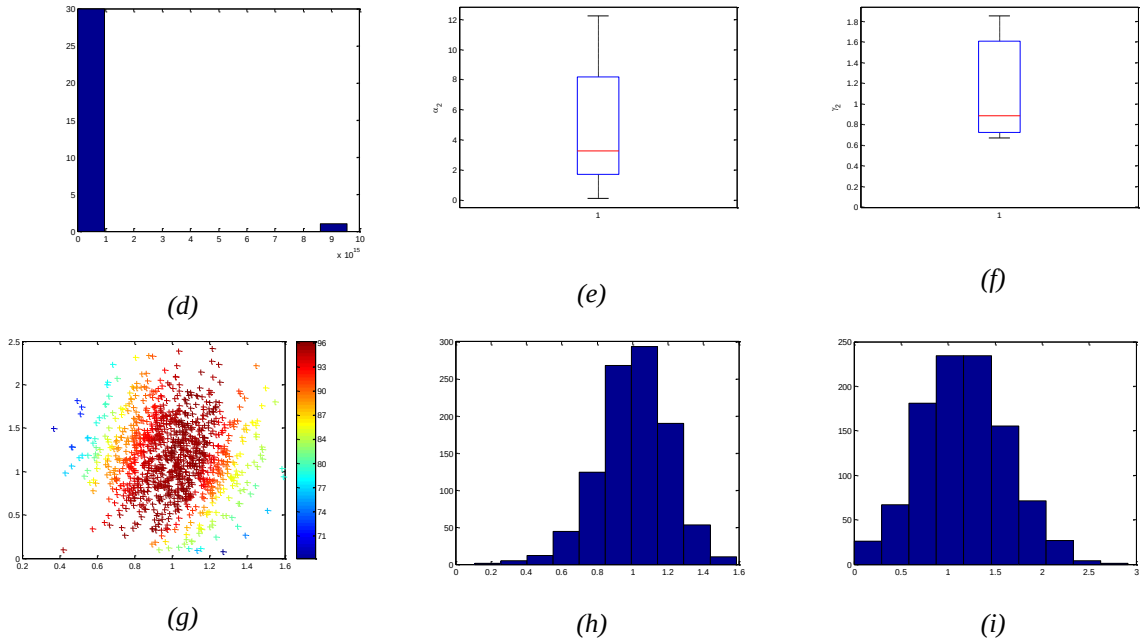


Fig. 2.19 (a) Distribution gaussienne Cas 1; (b) Couples solutions (γ_2, δ_2) du système de Cramer avec seule condition, $\delta_2 > 0$, le taux d'erreur minimal = 5% ; (c) Les distributions marginales des δ_2 candidats ; (d) Les distributions marginales des γ_2 candidats ; (e) boxplot des quartiles de 25%, 50% et 75% des δ_2 candidats; (f) boxplot des quartiles de 25%, 50% et 75% des γ_2 candidats, ayant limité les valeurs de δ_2; (g) Distribution conjointe des 1000 simulations Monte Carlo suivant une loi gaussienne, taux d'erreur minimal = 3.5% ; (h) Les distributions marginales des δ_2 candidats des simulations Monte Carlo; (i) Les distributions marginales des γ_2 candidats des simulations Monte Carlo.

En résumé, l'approche du « Coefficient de forme » géométrique est une procédure sous optimale par l'emploi de la classe de référence. Par l'utilisation des tirages Monté Carlo, le temps de calcul dépend du nombre des classes et du nombre des points mal classés. Néanmoins, on observe un bon comportement (taux d'erreur dans la moyenne des classifieurs testés et faible variance de l'estimation de l'erreur).

La méthode *Cs*-SVM présentée au paragraphe 2.4.2 est beaucoup plus rapide et on a le choix d'utiliser une classe de référence ou non ce qui augmente ou réduit le nombre de restrictions de l'espace (c'est également le cas pour *Cs*-LSSVM). Le temps de calcul et de classement pour une base de données de 200 points sur un ordinateur, muni d'un processeur Pentium M à 2GHz avec 1.5 GB de RAM sous « SVM^{Light} » est de 0.07 secondes. La constante contrôlant l'erreur de classement est fixée à C = 10 (exemple illustré à la fig.2.20a.). En comparaison le temps nécessaire sous Matlab 7 pour l'apprentissage du classifieur $Cs(\gamma_c, \delta_c)$ avec 1000 tirages Monte Carlo est de 3.33 secondes. Le classifieur *Cs*-SVM a un besoin d'une mémoire linéairement dépendant du nombre d'exemples d'apprentissage et du nombre de vecteurs de support.

Pour le Cas 1, la moyenne du taux d'erreur de *Cs*-SVM est meilleure de celle de la procédure KNN qui affiche une très bonne performance en taux d'erreur et variance. Dans le Cas 2 les performances du classifieur linéaire CL et *Cs*-SVM sont très semblables. On doit prendre en compte que pour le classifieur linéaire, on généralise dans un espace de huit dimensions (quatre prototypes par classe) et pour la procédure *Cs*-SVM l'espace est de quatre dimensions (deux paramètres par classe) dont deux fixées.

Le *Cs*-SVM est une méthode globale avec des paramètres ajustables en fonction des propriétés de la classe de sorte qu'elle fonctionne mieux que la règle KNN. Pour le Cas 1, la différence entre la performance du KNN est très proche de celle du modèle *Cs*-SVM, mais ces résultats sont dus à l'augmentation du nombre de voisins pour mieux classer les objets (une moyenne de 11 voisins) et aux classes faiblement mélangées (pour le cas 2 la différence est significative avec le test *t* de Student, $p \approx 0.002$).

On peut voir pour le cas de *Cs*-SVM et *Cs*-LSSVM qu'en fixant certaines dimensions du plan séparateur on ne diminue pas le taux de reconnaissance. Dans certains cas il y a des valeurs négatives des paramètres γ qui correspondent à des petites variances des classes (voir l'annexe).

La procédure *Cs*-LSSVM est plus simple à mettre en œuvre, car elle utilise un système linéaire pour l'optimisation, mais les résultats montrent que cette méthode est plus instable. Le taux d'erreur est plus grand en moyenne ainsi que sa variance.

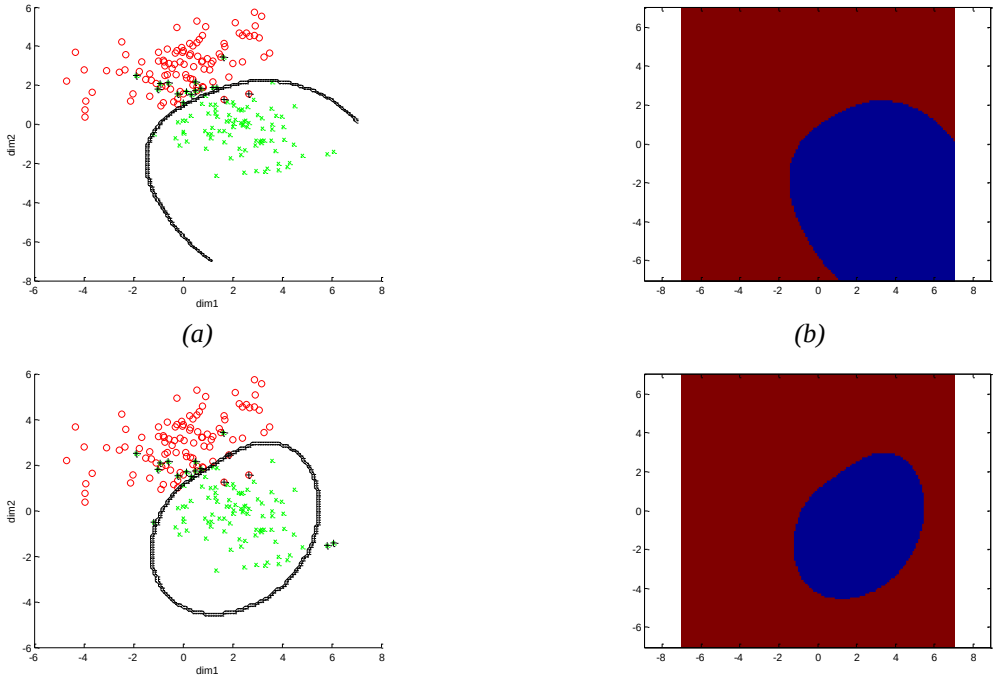


Fig. 2.20 Exemples de frontières obtenues pour le Cas 2 avec (a) le classifieur Cs-SVM, les points marqués avec une croix représentent les observations mal classées, taux d'erreur = 7.5%; (b) Illustration du contour de l'évolution spatiale Cs-SVM avec $\beta_1 = \beta_2$. (c) le classifieur Cs-LSSVM, les points marqués avec une croix représentent les observations mal classées, taux d'erreur = 9.5%; (d) Illustration du contour de l'évolution spatiale Cs-LSSVM avec $\beta_1 = \beta_2$.

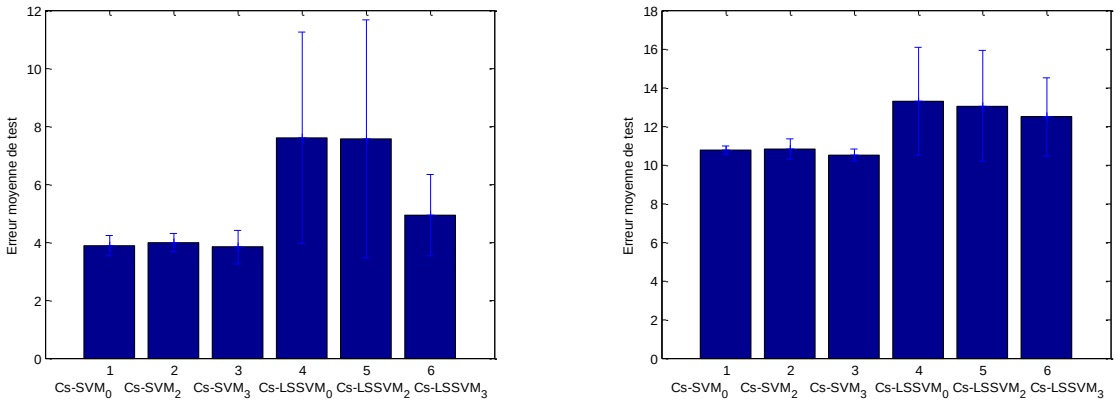
Il est prouvé que les LSSVM donnent des meilleures performances avec différents noyaux que le noyau linéaire [Abe 2005]. La méthode *Cs*-SVM a des problèmes de stabilité quand l'algorithme n'arrive pas à trouver une solution optimale dans un nombre d'itérations limité. Ce qui n'est pas le cas pour *Cs*-LSSVM qui trouve toujours une solution optimale mais si l'ensemble d'apprentissage est gros alors l'algorithme est lent. La constante contrôlant l'erreur de mauvais classement est fixée à $C = 10$ (exemple illustré à la fig.2.20b.).

Afin de comparer le comportement du *Cs*-SVM et *Cs*-LSSVM avec ou sans des contraintes, on a mené une étude sur les distributions gaussiennes du Cas 1 et Cas 2. L'étude détaillée avec tous les résultats obtenus est présentée dans l'annexe de cette thèse.

Suivant (2.26) on peut avoir trois cas possibles :

- On fixe l'une des deux classes comme classe de référence alors on fixe $\delta_1 = 1$ et $\gamma_1 = 1$ et on restreint le vecteur normal à l'hyperplan séparateur $\boldsymbol{\beta} = [1\ 1\ 1\ \delta_2]$ et le biais de positionnement de l'hyperplan par rapport à l'origine $\beta_0 = \log(1/\gamma_2) = -\log(\gamma_2)$. Les classifieurs sont notés : *Cs*-SVM$_3$ et *Cs*-LSSVM$_3$.
- Aucune classe n'est choisie comme classe de référence alors on restreint le vecteur normal à l'hyperplan séparateur $\boldsymbol{\beta} = [1\ 1\ \delta_1\ \delta_2]$, et le biais de positionnement de l'hyperplan par rapport à l'origine $\beta_0 = \log\left(\gamma_1 / \gamma_2\right)$. Les classifieurs sont notés : *Cs*-SVM$_2$ et *Cs*-LSSVM$_2$.
- On utilise juste le recodage des données suivant (2.24) et on applique le classifieur SVM ou LSSVM standard dans cet espace 4D non restreint. Les classifieurs sont notés : *Cs*-SVM$_0$ et *Cs*-LSSVM$_0$.

A la fig. 2.21a.b. sont illustrées la moyenne et la variance du taux d'erreur suivant ces trois cas pour les deux modèles des distributions gaussiennes.



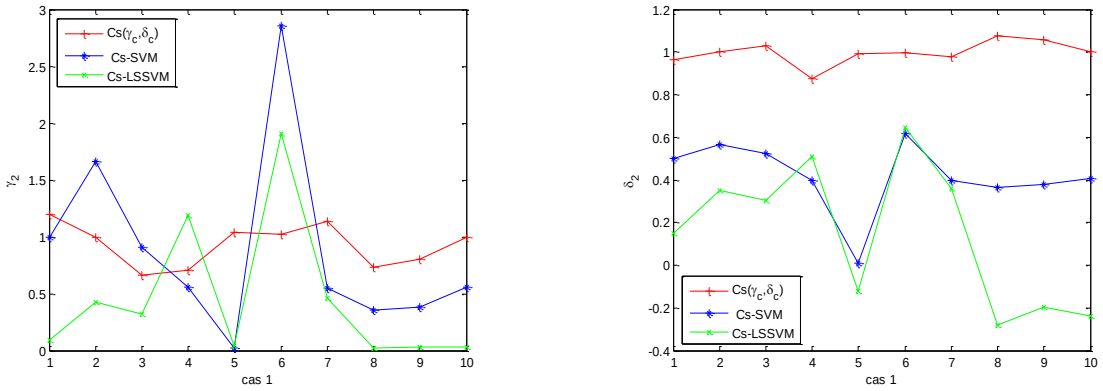
(a) *(b)*

Fig. 2.21 Moyenne et variance des taux d'erreur de test pour les modèles Cs-SVM$_0$ (espace non restreint), Cs-SVM$_2$ (sans classe de référence), Cs-SVM$_3$ (avec classe de référence), Cs-LSSVM$_0$ (espace non restreint), Cs-LSSVM$_2$ (sans classe de référence), Cs-LSSVM$_3$ (avec classe de référence) (a) Cas 1 ;(b) Cas 2.

On peut constater qu'en restreignant le vecteur normal à l'hyperplan séparateur et le biais, la performance du *Cs*-SVM varie très peu. Cela n'est pas le cas pour *Cs*-LSSVM, où l'on observe une diminution de la moyenne de l'erreur de test avec l'augmentation du nombre des dimensions restreintes. Tous les détails de cette étude sont présentés dans l'annexe.

Il est intéressant aussi de suivre l'évolution des paramètres d'ajustement correspondant au meilleur taux d'erreur suivant les différents modèles du « Coefficient de forme ». On a confirmé

les constatations de [Koening 2001] que les paramètres d'ajustement de la distance de Mahalanobis ne sont pas forcement les paramètres d'ajustement qui minimisent le taux d'erreur de classement des distributions gaussiennes. Ces derniers varient moins que ceux d'ajustement de la distance de Mahalanobis donc on peut supposer qu'ils ajustent avec moins de précision les modèles $Cs(\gamma_c,1)$, $Cs(1,\delta_c)$ pour les bases de données d'apprentissage mais ils permettent à mieux classer une observation de l'ensemble de test. Les valeurs des paramètres sont délimitées pour le classifieur *Cs* géométrique. Mais suivant le tirage des points Monte Carlo, on arrive à avoir plusieurs couples de paramètres qui optimisent le taux d'erreur. Sur la figure 2.22 sont illustrées les valeurs choisies aléatoirement parmi celles qui minimisent le taux d'erreur. Afin de pouvoir comparer les valeurs des paramètres (γ_2, δ_2) qui optimisent le taux d'erreur des *Cs*, *Cs*-SVM et *Cs*-LSSVM (les résultats sont illustrés sur la figure 2.22a.b.), on doit limiter encore plus le vecteur $\boldsymbol{\beta}$ = [1 1 1 δ_2] (la normale de l'hyperplan séparateur) et $\beta_0 = \log(1/\gamma_2) = -\log(\gamma_2)$ (le biais) pour y ajouter la classe de référence comme dans le cas du « Coefficient de forme » géométrique. Dans le cas du classifieur *Cs* géométrique, les valeurs optimales de δ_2 sont trouvées par procédure Monté Carlo dans un intervalle borné à valeurs positives. Notons alors que des valeurs négatives sont possibles pour les paramètres d'ajustement avec les classifieurs *Cs*-SVM et *Cs*-LSSVM.



(a) *(b)*

Fig. 2.22 (a) Valeurs des γ_2 optimisant le taux d'erreur pour chaque distributions gaussiennes de l'ensemble d'apprentissage Cas 1 suivant les modèles $Cs(\gamma_2, \delta_2)$, Cs-SVM et Cs-LSSVM ; (b) Valeurs des δ_2 optimisant le taux d'erreur pour chaque distributions gaussiennes de l'ensemble d'apprentissage Cas 1 suivant les modèles $Cs(\gamma_2, \delta_2)$, Cs-SVM et Cs-LSSVM

Les valeurs négatives des paramètres $-\log(\gamma_2)$ correspondent à des petites variances des classes et à la position de l'hyperplan séparateur par rapport à l'origine de l'espace des paramètres.

Dans l'annexe sont présentées les valeurs de coefficients pour les deux cas de distributions gaussiennes pour les classifieurs *Cs*-SVM et *Cs*-LSSVM sans ou avec des contraintes du plan séparateur.

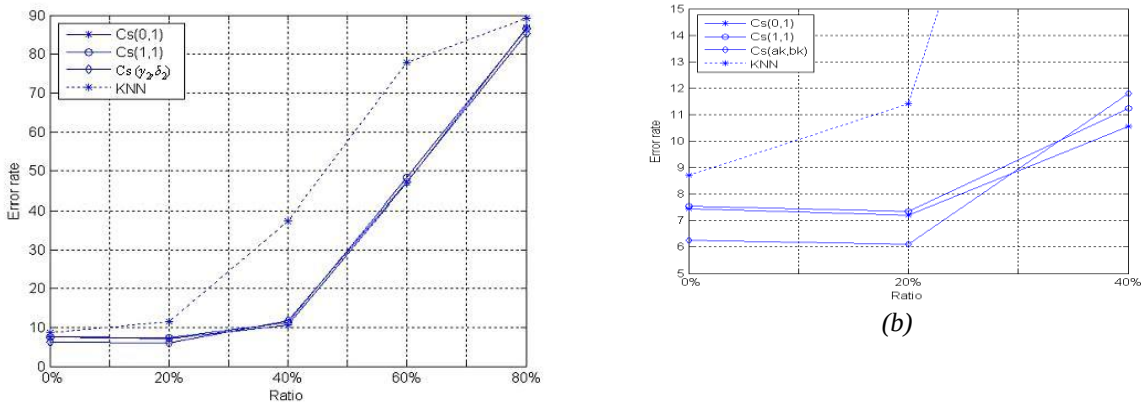
2.5.1. Matrices de dissimilitudes creuses

Un cas particulier des matrices de dissimilitudes que l'on a abordé dans cette thèse, concerne les matrices creuses. Une matrice creuse est une matrice contenant beaucoup de valeurs de

données manquantes qui, sont laissées « vides » ou remplacées par des 0. Quand on veut manipuler ou stocker des matrices creuses en mémoire, il est avantageux voire souvent nécessaire d'utiliser des algorithmes et des structure de données qui prennent en compte la structure creuse de la matrice [Duff et al. 1986 ; Tewarson 1973].

Le but de l'utilisation de ces matrices creuses est de démonter la performance des classifieurs « Coefficient de forme » vis-à-vis un volume fluctuant des données manquantes. En effet ces classifieurs s'implémentent tout naturellement en ne considérant que les dissimilitudes présentes. Nous avons expliqué que l'espace de représentation des dissimilitudes se calcul à partir des moyennes et variances sur des populations de dissimilitudes, alors dans le cas de matrices creuses, ces mêmes statistiques seront calculées, mais uniquement bien sur avec les dissimilitudes acquises.

Ici nous présentons les résultats obtenus par simulation. Pour obtenir une matrice creuse, des masques binaires sont générés sur la matrice triangulaire supérieure (0: la dissimilitude est manquante, 1: la dissimilitude est connue). Le masque de l'ensemble est fixé par symétrie - lorsqu'un coefficient d_{ij} est non nul, il en est de même pour le coefficient d_{ji}. Ce processus est répété dix fois pour un même ratio de données manquantes. Les ratios sont évalués sur le masque triangulaire. Le ratio est le même pour les dissimilitudes intra-classe de l'ensemble d'apprentissage et l'ensemble de test et pour les dissimilitudes inter-classe entre l'ensemble d'apprentissage et le test. La mise en œuvre de la procédure d'apprentissage est la même que précédemment définie, mais ici, les statistiques (moyenne et variance) pour l'indice *Cs* sont calculées pour des dissimilitudes connues. L'expérience présentée est menée sur les données gaussiennes de Cas 2. Avec la base de données artificielle, on observe (figure 2.23) que jusqu'à 40% de valeurs manquantes, le taux d'erreur augmente lentement puis plus rapidement.



(a) *(b)*

Fig. 2.23 Distributions gaussiennes creuses cas 2, erreur de classement pour les trois modèles Cs(0,1), Cs(1,1) et Cs(γ_c, δ_c), et le K-NN, selon le ratio des valeurs manquantes, (a) de 0 à 80% et (b) de 0% à 40%.

À partir de 20% de valeurs manquantes, les modèles *Cs*(0, 1) et *Cs*(1, 1) deviennent un peu plus intéressants que le modèle complet *Cs*(δ_c, γ_c). Ceci est justifié par le fait que que les premiers sont moins sensibles aux valeurs manquantes lors de l'apprentissage: seule les pseudo inerties $I(\omega_1)$ et $I(\omega_2)$ sont estimées. Le classifieurs KNN devient inefficace à partir de 20% de valeurs manquantes.

2.6. Conclusion

Suivant les expériences menées et les résultats obtenus ci-dessus on peut conclure que les différents modèles du « Coefficient de forme » se caractérisent par :

- Précision – bonne fiabilité de la règle *Cs* sous ces différentes formes en comparaison avec le classifieur linéaire et la règle KNN. La précision telle que mesurée sur les ensembles d'apprentissage et la précision telle que mesurée sur des données de test sont très cohérentes. Même dans sa forme la plus simple sans paramètres d'adaptation le « Coefficient de forme » généralise très bien. Sous cette forme là, il est particulièrement performant pour des matrices creuses.
- Vitesse – temps de calcul et de classement du modèle *Cs*-SVM pour une base de donnée de 200 points sur un ordinateur, muni d'un processeur Pentium M à 2GHz avec 1.5 GB de RAM sous « SVM^{Light} » est de 0.07 secondes. Le modèle *Cs*-LSSVM est aussi rapide à optimiser pour des bases de données faiblement mélangées. En comparaison le temps nécessaire sous Matlab 7 pour le *Cs* géométrique avec 1000 tirages de Monte Carlo est de 3.33 secondes.
- Parcimonie – avec seulement deux paramètres d'ajustement par classe, le « Coefficient de forme » est facile à mettre en œuvre et robuste.
- Inconvénients - L'approche géométrique du *Cs* est une procédure assez lente car le nombre de calculs dépend des points mal classés et le nombre de classes. La difficulté réside dans la solution du système des équitations linéaires qui n'est pas toujours défini et on obtient parfois des couples candidats de paramètres d'ajustement dont les valeurs tendent vers l'infini. Le modèle *Cs*-SVM parfois a des problèmes de stabilité quand l'algorithme n'arrive pas à trouver une solution optimale dans un nombre d'itérations limité. Le modèle *Cs*-LSSVM trouve toujours une solution optimale mais si l'ensemble d'apprentissage est gros alors l'algorithme est lent. C'est une méthode très sensible aux points « outliers ».

On a comparé le comportement du classifieur « Coefficient de forme » avec celui de la règle de KNN, de la méthode de Pekalska (pour le classifieur linéaire) sur des bases de données des distributions gaussiennes et des bases de données creuses. Les résultats présentés dans ce chapitre font partie des publications [Manolova&Guérin-Dugué 2007], [Manolova et al. 2008], [Manolova&Guérin-Dugué 2009].

Dans le chapitre suivant sera proposée une méthode de caractérisation d'images en multi échelle, développée pour une application particulière de classement des visages. Cette méthode nous permettra de décrire un système complet de reconnaissance d'image – de l'étape de la description vectorielle de l'image à l'étape de classement. Avec ce système, on peut contrôler chaque aspect de la procédure et ajuster tous types de paramètres afin d'avoir meilleurs résultats en reconnaissance. Les résultats de classement en utilisant les classifieurs « *Cs* » seront présentés dans le chapitre 4.

Chapitre 3. Caractérisation d'images en multi échelle – méthode de la «Pyramide Réduite Différentielle»

Dans ce chapitre on présente une méthode, destinée à la caractérisation d'images en multi échelle afin de pouvoir extraire les vecteurs caractéristiques, construire des matrices de dissimilitudes et classer des images. Cette méthode peut être aussi utilisée pour la recherche d'images par le contenu dans des bases de données des images. L'approche a été développée à l'Université Technique de Sofia sous la tutelle de prof. Roumen Kountchev. Elle a été proposée pour des applications du projet européen FIVES [http://fives.kau.se/] pour la recherche de contenu interdit dans des bases de données d'images.

3.1. La représentation multi échelle des images

Les utilisateurs et les développeurs dans de nombreux domaines applicatifs sont à la recherche de nouvelles méthodes et algorithmes pour accéder et manipuler des bases de données d'images. En ce qui concerne les données visuelles, l'attention se concentre généralement sur le problème du classement des objets visuels (images, flux vidéo) à partir de leur contenu, afin de permettre la meilleure gestion des bases de données visuelles pour accéder à leur contenu.

La représentation de l'image est basée sur différentes techniques, utilisant des matrices, des vecteurs de caractéristiques, des pyramides multi-résolutions, des « R-trees », des transformations orthogonales, des représentations de perception anisotropique, [Schettin et al. 2001 ; Cawkell 2000 ; Smeulders et al. 2000 ; Del Bimbo 1999 ; Nixon&Aguado 2002]. Dans ce chapitre, on présente une nouvelle approche de représentation de l'image, basée sur la décomposition pyramidale de son spectre. La méthode nous permet de construire un vecteur de représentation pour chaque image de la base et de calculer la matrice de dissimilitudes entre ces représentations afin d'appliquer différentes méthodes de classement. C'est une décomposition en multi échelle qui permet une approche « coarse to fine » pour classer et/ou identifier les objets [Gevers 2001], en reprenant ce principe du système visuel humain [Gupta&Jain 1997 ; Adelson et al. 1991a ; Adelson 1991].

L'importance d'analyser des images à plusieurs échelles découle de la nature des images elles-mêmes. Les scènes naturelles par exemple peuvent contenir des objets de diverses tailles, et ces objets peuvent avoir des caractéristiques de tailles différentes. En outre, les objets peuvent être à des distances différentes par rapport à l'observateur. En conséquence, toute procédure d'analyse qui est appliquée uniquement à une échelle peut s'avérer être incomplète ou surdimensionnée. L'analyse multi échelle est alors naturelle. La méthode proposée est la décomposition par la « Pyramide Réduite Différentielle » [Kountchev et al. 2002].

La méthode de décomposition pyramidale de l'image est appelée aussi "inverse" à cause de l'ordre suivi pour obtenir les niveaux de la pyramide : de plus grossier au plus fin, en correspondance avec l'exigence de transmission progressive d'images.

3.2. Principes de base de la “Pyramide Réduite Différentielle” (PRD)

L'essentiel de la décomposition PRD, appliquée sur une image en niveaux de gris est expliqué comme suit :

- une transformation orthogonale 2D est appliquée sur l’image d’entrée $B(i,j)$ et un filtrage (un masque binaire) par valeur des coefficients est utilisé afin de retenir un nombre limité de coefficients résultants. Ces coefficients, constituent le premier niveau de la pyramide ($p = 0$) ;
- en utilisant les valeurs de ces coefficients, l'image est restaurée avec la transformation orthogonale inverse et ensuite l’on soustrait pixel par pixel l’approximation obtenue $\tilde{B}_0(i, j)$ de l'image originale ;
- l'image de différence obtenue $E_0(i,j)$, qui est de même taille que l’image originale $B(i,j)$, est divisée en 4 sous-images et chaque sous-image est traitée avec la transformation orthogonale 2D de nouveau et le filtrage est appliqué. Les coefficients qui en résultent constituent le deuxième niveau de la pyramide ($p = 1$) ;
- le traitement se poursuit d'une manière similaire pour les niveaux de la pyramide suivants.

La figure 3.1 illustre le schéma block de la décomposition PRD avec la transformation de Mellin-Fourier.

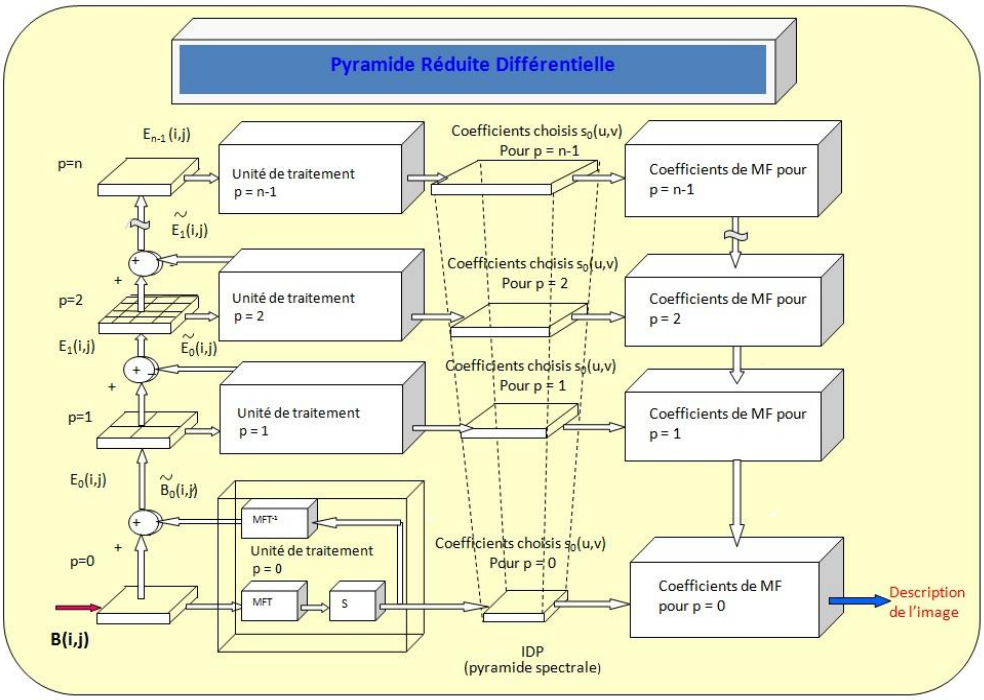


Fig. 3.1 Schéma block de la pyramide PRD avec n niveaux de décomposition avec la transformation de Mellin-Fourier.

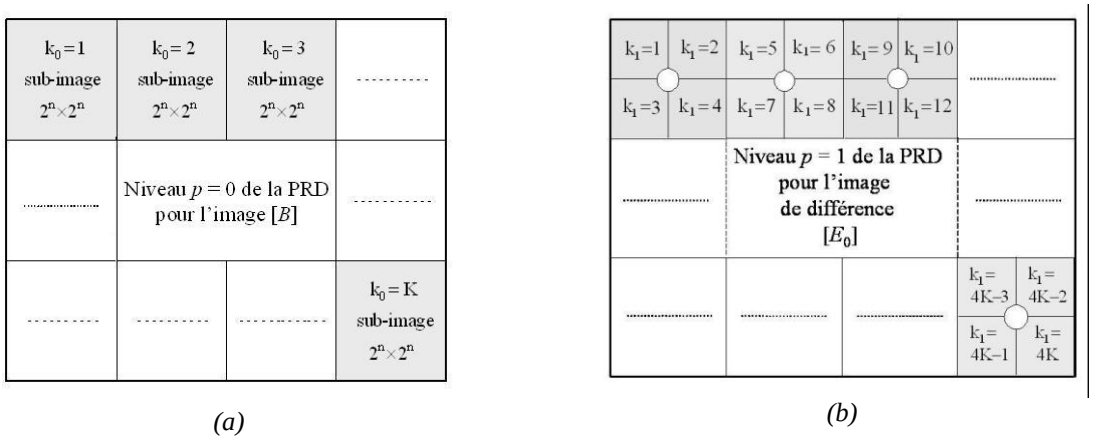
Ainsi, tous les niveaux de la pyramide sont composés seulement de coefficients calculés par la transformation choisie et retenus après le filtrage. L'ensemble des coefficients retenus pour chaque niveau de la pyramide peut être différent. La décomposition de l'image est arrêtée au niveau d'échelle requis dont le critère dépend de l'application, en général avant le niveau le plus fin de la pyramide.

La méthode PRD pour la décomposition des images est introduite dans [Kountchev 2002]. Cette approche a été utilisée avec succès pour la compression des images. En utilisant la PRD on a élaboré une méthode de recherche d'images en couleur par contenu [Kountchev&Manolova 2005] qui utilise la transformation de Karhunen-Loeve projetant les pixels couleur de trois à une dimension. Ici, nous utiliserons directement des images couleur transformées en niveaux de gris à travers la première dimension de l'espace (YCbCr).

3.2.1. Construction de la PRD d'une image

Chaque image en niveau de gris peut être représentée par la PRD en appliquant une transformation linéaire. On utilise des transformations orthogonales comme la transformation discrète de Fourier (FFT), de Cosinus (DCT), ou de Mellin-Fourier (MF) ou encore de Walsh-Hadamard (WH). Le choix de la transformation dépend de l'application choisie. Dans [Manolova et al. 2009] la transformation choisie est la transformation de Cosinus Discrète car est une transformation couramment utilisée dans la manipulation d'images. Dans [Manolova&Kountchev 2010], la transformation choisie est celle de Mellin-Fourier qui est un outil mathématique utile pour la reconnaissance d'image car son spectre est invariant par rapport de à rotation, la translation et aux changements de l'échelle. La transformation de Fourier elle-même (FT) est invariante par rapport de la translation, par le module et sa conversion en coordonnées log-polaires convertit des variations d'échelle et de rotation en des translations dans le domaine spectral.

Suivant l'algorithme de PRD proposé dans [Kountchev 2002] l'image d'entrée est divisée en blocs de taille $2^n \times 2^n$ où *n* est un nombre entier (fig. 3.2a.). L'algorithme peut être appliqué aussi sur l'image entière qui doit avoir aussi des dimensions $2^n \times 2^n$.



(a) *(b)*

Fig. 3.2 (a) Division de la matrice de l'image [B] en sous blocs de taille $2^n \times 2^n$ pour le niveau p = 0 ; (b) Division de l'image de différence [E_0] en sous blocs de taille $2^{n-1} \times 2^{n-1}$ pour le niveau p = 1.

Dans le cas où les dimensions de l'image ne sont pas en puissance de 2, une solution consiste à redimensionner l'image avec une interpolation des pixels afin d'obtenir les bonnes dimensions. Cette méthode peut provoquer des légères distorsions des formes générales de l'image mais généralement, elles sont négligeables. Une perturbation importante peut se produire seulement dans les cas où la largeur et l'hauteur de l'image diffèrent fortement. Une bonne solution de ce problème de redimensionner l'image dans le cadre de la PRD a été proposée dans [Aleksieva 2008].

Alors l'image peut être décomposée sans perte d'information de la façon suivante:

L'image originale [$B(2^n)$] est présentée comme une somme de composantes, représentant la décomposition de l'image dans le domaine spatial en correspondance par la relation:

$$\left[B\left(2^n\right)\right]=\left[\tilde{B}_0\left(2^n\right)\right]+\sum_{p=1}^{n-1}\left[\tilde{E}_{p-1}^{k_p}\left(2^n\right)\right]+\left[E_{n-1}\left(2^n\right)\right], \tag{3.1}$$

où p est le niveau de la pyramide de décomposition, $k^p = 1, 2, \ldots 4^p$ est le numéro de sous-image au niveau p de la pyramide. La composante $E_{n-1}\left(2^n\right)$ est un élément de reste. Chaque composante dans (3.1) est de taille $2^n \times 2^n$. Les éléments de la décomposition (3.1), $\tilde{B}_0\left(2^n\right)$ et $\tilde{E}_{p-1}^{k_p}\left(2^n\right)$ représentent respectivement le niveau 0 et p de la PRD, obtenus par un filtrage passe-bas après une transformation orthogonale appliquée respectivement sur l'image d'entrée [$B(2^n)$] et les sous-images de l'image de différence pour le niveau p ([$E_{p-1}\left(2^n\right)$]), définis comme suit:

$$\left[\tilde{B}_0\left(2^n\right)\right]=\left[T_0\left(2^n\right)\right]^{-1}\left[\tilde{S}_0\left(2^n\right)\right]\left[T_0\left(2^n\right)\right]^{-1}, \tag{3.2}$$

où [$T_0(2^n)$] est la matrice, de la transformation linéaire orthogonale 2D utilisée et [$T_0(2^n)$]$^{-1}$ est son inverse, [$\tilde{B}_0\left(2^n\right)$] est l'approximation de l'image originale. Et $\left[\tilde{S}_0\left(2^n\right)\right]$ est défini comme suit :

$$\left[\tilde{S}_0\left(2^n\right)\right]=m_0(u,v)S_0(u,v) \tag{3.3}$$

$$m_0(u,v)=\begin{cases}1 \text{ pour } (u,v)\in V_0\\ 0 \text{ ailleurs}\end{cases}, \tag{3.4}$$

$$[S_0\left(2^n\right)]=[T_0\left(2^n\right)][B_0\left(2^n\right)][T_0\left(2^n\right)] \tag{3.5}$$

où $m_0(u,v)$ est une masque binaire (filtrage), V_0 est le nombre choisi des coefficients à retenir et u, v = 0, à 2^n-1 sont des coefficients spectraux. Le choix de V_0 dépend de l'application et des résultats désirés. $S_0(u, v)$ sont des éléments de la matrice spectrale.

L'erreur d'approximation pour le niveau de la pyramide p = 0 est:

$$[E_0\left(2^n\right)]=[B_0\left(2^n\right)]-[\tilde{B}_0\left(2^n\right)]. \tag{3.6}$$

Cette matrice est ensuite divisé en 4 sous-matrices $[E_0^{k_1}(2^{n-1})]$, chaque de taille $2^{n-1} \times 2^{n-1}$ et le numéro de séquence $k_1 = 1, 2, 3, 4$:

$$[E_0(2^n)] = \begin{bmatrix} [E_0^1(2^{n-1})] & [E_0^2(2^{n-1})] \\ [E_0^3(2^{n-1})] & [E_0^4(2^{n-1})] \end{bmatrix}. \tag{3.7}$$

Les niveaux suivants de la PRD $p = 1, 2, .., n-1$ sont calculés de manière identique. La matrice de différences d'approximations pour le niveau p-1est la suivante:

$$[\tilde{E}_{p-1}(2^n)] = \begin{bmatrix} [\tilde{E}_{p-1}^1(2^{n-p})][\tilde{E}_{p-1}^2(2^{n-p})] & -- & [\tilde{E}_{p-1}^{2^p}(2^{n-p})] \\ [\tilde{E}_{p-1}^{2^p+1}(2^{n-p})][\tilde{E}_{p-1}^{2^p+2}(2^{n-p})] & -- & [\tilde{E}_{p-1}^{2^p+p}(2^{n-p})] \\ --- & --- & --- \\ [\tilde{E}_{p-1}^{4^p-2^p+1}(2^{n-p})][\tilde{E}_{p-1}^{4^p-2^p+2}(2^{n-p})] & -- & [\tilde{E}_{p-1}^{4^p+1}(2^{n-p})] \end{bmatrix}, \tag{3.8}$$

où

$$[\tilde{E}_{p-1}^{k_p}(2^{n-p})] = [T_p(2^{n-p})]^{-1}[\tilde{S}_p^{k_p}(2^{n-p})][T_p(2^{n-p})]^{-1} \quad \text{pour } k_p = 1,2,...,4^p. \tag{3.9}$$

Chaque matrice $[\tilde{E}_{p-1}(2^n)]$ est composée des sous-matrices $[\tilde{E}_{p-1}^{k_p}(2^{n-p})]$, obtenues en résultat de sa division « quad-tree » en 4^p blocks carrés. Alors les spectres de chaque sous-image sont :

$$[\tilde{S}_p^{k_p}(u,v)] = m_p(u,v)S_p^{k_p}(u,v) \tag{3.10}$$

$$m_p(u,v) = \begin{cases} 1 \text{ pour } (u,v) \in V_p \\ 0 \text{ ailleurs} \end{cases}, \tag{3.11}$$

où V_p est le nombre choisi des coefficients à retenir pour le niveau p. La matrice de différence pour le niveau p est :

$$[E_{p-1}(2^{n-p})] = \begin{cases} [B(2^n)] - [\tilde{B}_0(2^n)] \text{ pour } p = 1 \\ [E_{p-2}(2^{n-p})] - [\tilde{E}_{p-2}(2^{n-p})], \text{ pour } p = 2,3,...,n-1 \end{cases}. \tag{3.12}$$

Les coefficients spectraux retenus de chaque niveau pyramidal sont rangés dans des matrices suivant leur niveau *p*. Afin de représenter ses coefficients dans des vecteurs, on utilise le parcours récursif de Peano-Hilbert, illustré sur la figure 3.2.

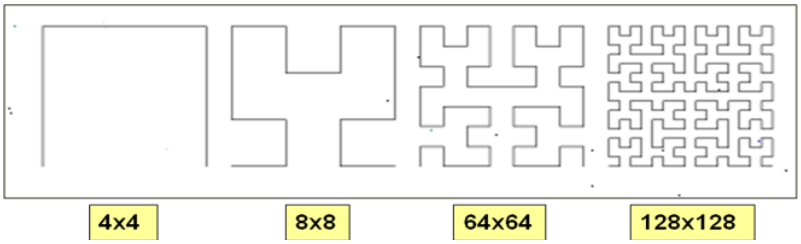
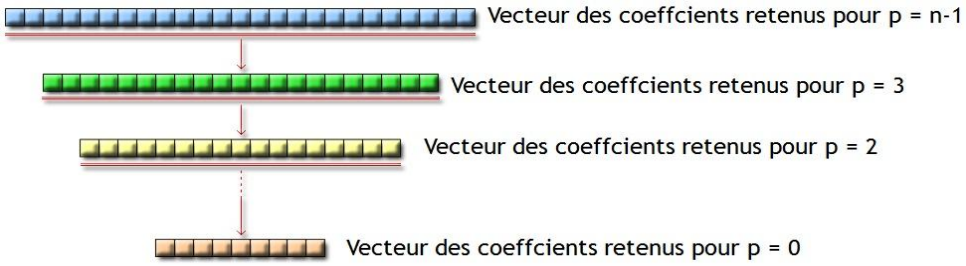
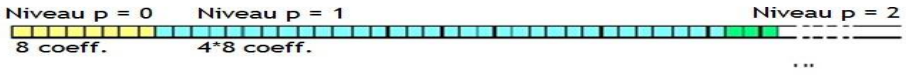


Fig. 3.2 Le parcours récursif de Peano-Hilbert pour des blocs de taille $2^n \times 2^n$ *avec n = 2, 3, 4, 5.*

Finalement chaque image de la base de données peut être représentée par un vecteur caractéristique soit pour chaque niveau de la pyramide soit pour tous les niveaux avec des coefficients retenus pour chaque couche de la pyramide (voir fig. 3.3a.b.).



(a)



(b)

Fig. 3.3 Vecteurs de représentation des images (a) Vecteur des coefficients retenus pour chaque niveau de la pyramide – 8 coefficients par bloc, (b) vecteur de représentation total de tous les niveaux de décomposition.

3.2.2. La transformation de Mellin-Fourier

Pour l'application de la PRD pour le projet FIVES [http://fives.kau.se/] où un problème de classement a été posé, on a décidé d'utiliser la transformation MF à cause des ces propriétés d'invariance à la rotation, la translation et le changement d'échelle. Un algorithme pour cette transformation par rapport à ce problème a été élaboré et sera présenté dans le paragraphe suivant.

La transformation de MF est réversible mais avec des pertes d'information car le passage des coordonnées cartésiennes vers coordonnées log-polaires (LP) n'est pas réversible. La transformation log-polaire est un algorithme important de la théorie de la vision humaine. La géométrie log-polaire des images a été motivée par sa ressemblance avec la structure de la rétine humaine et par ses qualités de compression de données [Schwartz 1994 ; Ferrari et al. 1995].

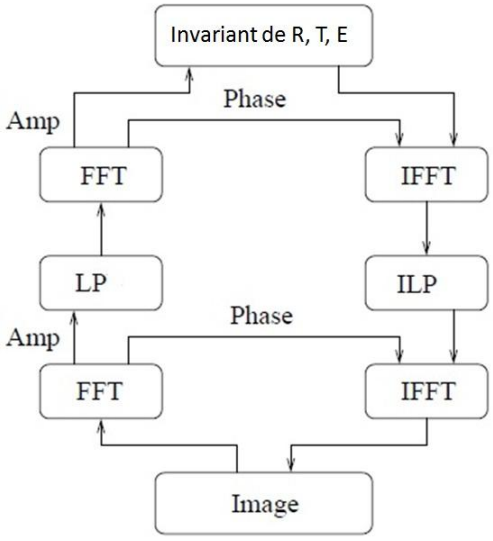


Fig. 3.4 Schéma block de la transformation de Mellin-Fourier directe et inverse. FFT – transformation rapide de Fourier, LP – transformation log-polaire, R – rotation, T – translation, E – échelle, IFFT – transformation inverse de Fourier, ILP – transformation inverse log-polaire.

Comme illustré sur la figure 3.4, l'algorithme de la transformation MF directe a trois étapes. Ces trois étapes seront décrites en détails dans le contexte de la décomposition PRD:

1. La transformation discrète de Fourier est appliquée sur l'image en niveaux de gris de l'entrée $B(m, n)$ de dimensions $N \times N$:

$$F(k,l) = \sum_m \sum_n B(m,n) e^{-\frac{2\pi}{N} j(mk+nl)} . \tag{3.13}$$

On peut représenter (3.13) par sa phase et son spectre:

$$\Theta_1(k,l) = \operatorname{arctg}\left(\frac{\operatorname{Im}(F(k,l))}{\operatorname{Re}(F(k,l))} \right), \tag{3.14}$$

$$M_1(k,l) = \sqrt{\operatorname{Re}(F(k,l))^2 + \operatorname{Im}(F(k,l))^2} . \tag{3.15}$$

Puisque l'image B est réelle, Re[F] est pair et Im[F] est impair symétriquement autour de la origine. En d'autres termes, $F(-k, -l) = F^*(k, l)$. Si l'image originale B est circulairement translatée en $\tilde{B}(m,n) = B(m+m_0, n+n_0)$, sa DFT sera donnée par $\tilde{F}(k,l) = F(k,l) e^{-\frac{2\pi}{N} j(m_0+n_0)}$. Par conséquent, le spectre de puissance de l'image translatée sera $\tilde{M}_1 = M_1$, alors que le spectre de la phase sera modifié. Afin de construire une représentation invariante à la translation, nous utiliserons le spectre de l'amplitude, et rejetons le spectre de phase. En vertu de la symétrie, même pour le spectre de l'amplitude, il suffit de conserver les coefficients $M_1(k, l)$ dans les deux

quadrants $\left\{(k,l), k \geq 0, l = -\frac{N}{2},...,\frac{N}{2}-1\right\}$, ce qui réduit les besoins en stockage de moitié. Il convient de noter que, même si cette représentation est invariante à la translation, elle implique une perte d'information et n'est donc pas inversible. Dans le contexte de l'algorithme de décomposition par PRD, le pas suivant est de centrer le spectre d'amplitude et d'appliquer un masque binaire carré au spectre de l'amplitude (filtrage par valeur de l'amplitude) et de retenir un petit nombre de coefficients significatifs :

$$M_1(k,l) = \begin{cases} M_1(k,l) \text{ si } (k,l) \in \text{masque binaire} \\ 0, \text{ ailleurs} \end{cases}. \tag{3.16}$$

La zone des coefficients retenus est un carré avec un côté $H \leq N$ qui contient le centre du spectre des amplitudes (H - nombre pair). Pour $H < N$ et $(k,l) = -(H/2),-(H/2)+1,...,0,...(H/2)-1$, ce carré contient les coefficients de basses fréquences seulement.

2. Afin de réaliser l'invariance à la rotation et à l'échelle aussi, nous effectuons d'avantage des opérations log-polaires et non inversibles sur le spectre d'amplitude M_1. C'est la transformation de coordonnées cartésiennes (u, v) en cordonnées log-polaires (ρ, ϑ) :

$$\rho = \log\sqrt{(k-k_0)^2 + (l+l_0)^2}\ , \tag{3.17}$$

$$\vartheta = a\tan\frac{l-l_0}{k-k_0}, \tag{3.18}$$

où (k_0, l_0) est à l'origine du nouveau système de coordonnées à l'égard du cartésien. Alors on obtient :

$$k = e^{\rho}\cos\vartheta + k_0, \tag{3.19}$$

$$l = e^{\rho}\sin\vartheta + l_0. \tag{3.20}$$

Bien qu'un tel changement de coordonnées soit bijectif dans le domaine continu, il n'est pas de même pour le cas discret, où les coordonnées du domaine transformé doivent prendre des valeurs d'une grille d'échantillonnage donnée.

Puisque u et v ont des valeurs discrètes, quelques-uns des coefficients de $M_1(\rho,\theta)$ sont manquants. À la fin de la transformation, les coefficients manquants $M_1(\rho_i,\theta_i)$ sont interpolés à l'aide de leurs plus proches voisins $M_1(k,l)$ dans le système de coordonnées rectangulaires (u, v) dans le sens horizontal ou vertical (interpolation d'ordre zéro). Le nombre de cercles discrets dans le système polaire d'un rayon ρ_i pour $i = 1$, à H, est égal au nombre des angles discrets θ_i. La taille (en coordonnées rectangulaires) du côté du carré H inscrit dans la matrice de la transformation polaire est telle que, pour assurer le transfert d'un maximum des coefficients sans

changement. Pour cela, la transformation polaire est modifiée conformément à la figure 3.5 et suivant [Kountchev et al. 2010]. Par exemple, pour une image, le radius du cercle circonscrit est:

$$r = \left(\sqrt{2}/2\right)H \tag{3.21}$$

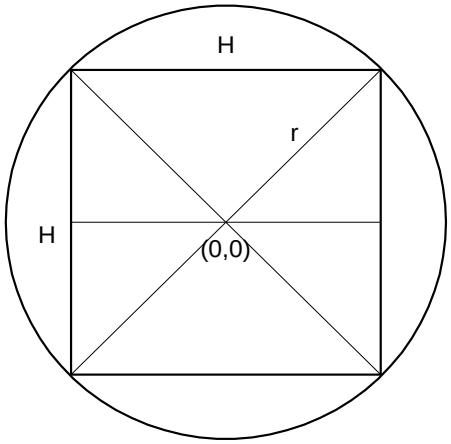


Fig. 3.5 Relation géométrique entre H et r.

Le moindre pas entre deux cercles concentriques est calculé comme suit :

$$\Delta\rho = r^{(1/H)} \tag{3.22}$$

Le résultat pour chaque cercle et chaque angle de la grille d'échantillonnage (exemple illustré sur la figure 3.6) est le suivant :

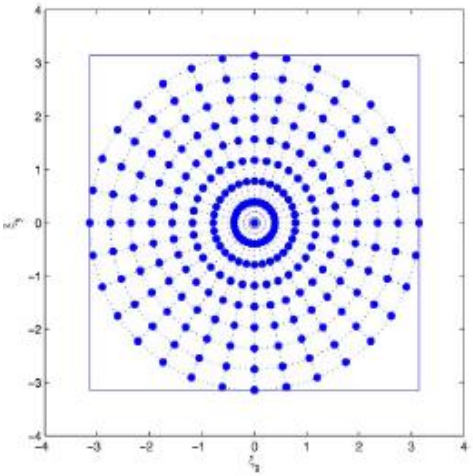


Fig. 3.6 Grille d'échantillonnage log-polaire avec 8 cercles concentriques et 16 rayons angulairement équidistants.

$$\rho_i = \left(\Delta\rho\right)^i = r^{i/H} \text{ pour } i = 1,2,..,H \tag{3.23}$$

$$\theta_i = \left(2\pi/H\right)i \text{ pour } i = \left(-H/2\right),...,0,...\left(-H/2\right)-1. \tag{3.24}$$

3. Le dernier pas de l'algorithme direct est la deuxième transformation discrète de Fourier de l'image log-polaire. On réécrit le spectre de l'amplitude en fonction de (ρ,ϑ):

$$F(\rho,\theta)=\sum_m\sum_n B(m,n)e^{-\frac{2\pi e^{\rho}}{N}j(m\cos\vartheta+n\sin\vartheta)}. \tag{3.25}$$

Et on applique la deuxième DFT :

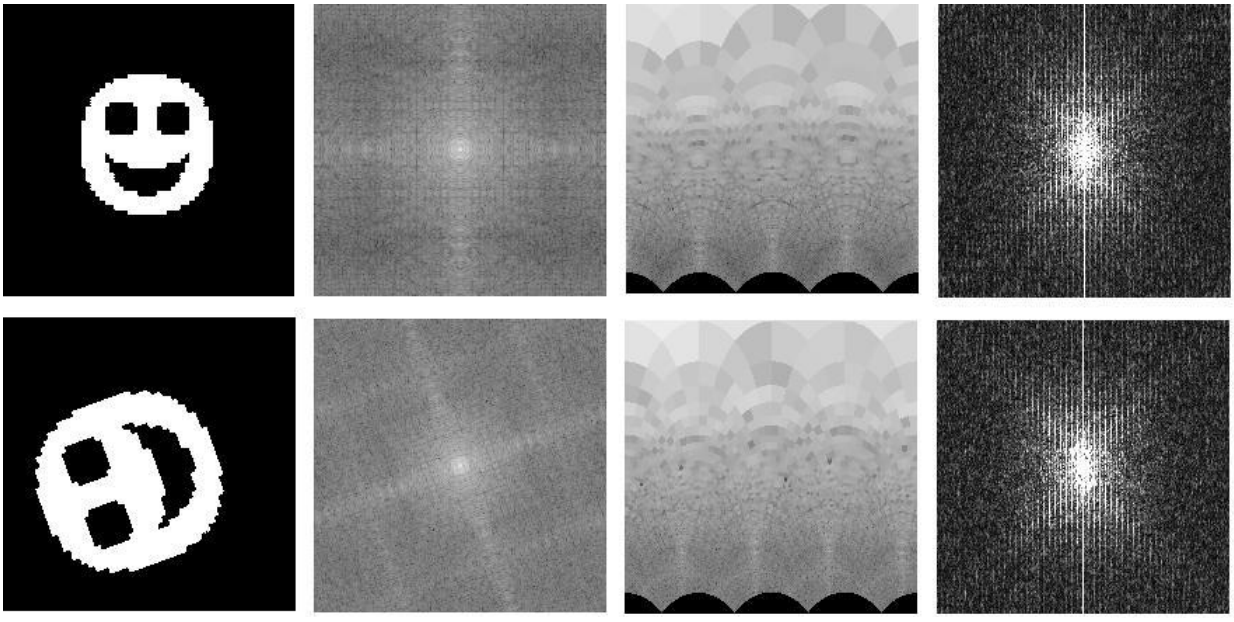
$$S(u,v)=\sum_\rho\sum_\vartheta F(\rho,\vartheta)e^{-\frac{2\pi}{N}j(\rho u+\vartheta v)}, \tag{3.26}$$

$$\Theta_2(u,v)=a\tan\left(\frac{\mathrm{Im}(S(u,v))}{\mathrm{Re}(S(u,v))}\right), \tag{3.27}$$

$$M_2(u,v)=\sqrt{\mathrm{Re}(S(u,v))^2+\mathrm{Im}(F(u,v))^2}\ . \tag{3.28}$$

Le deuxième spectre M_2 représente une description invariante à la translation, à la rotation et à l'échelle. Ce spectre est normalisé avant d'extraire les vecteurs de caractéristiques. On construit alors la représentation de l'image en choisissant un nombre limité de coefficients significatifs du spectre de l'amplitude M_2 par un nouveau masque binaire.

La figure 3.7, illustre l'invariance de la représentation proposée sur une image binaire.



(a) *(b)* *(c)* *(d)*

Fig. 3.7 (a) Image binaire d'origine de dimensions 209 × 209 et la même image après rotation de 110 degrés, une échelle de 1.3 et translation de l'axe x : -10 et de l'axe y : 10; (b) les spectres de l'amplitude équivalents à chaque image de la première DFT; (c) les transformations log-polaire équivalentes des spectres de l'amplitude; (d) les spectres de l'amplitude de la deuxième DFT de chaque image;

Les deux spectres d'amplitude centrés, représentés sur la fig. 3.7d, sont presque identiques. Après avoir construit le vecteur représentatif de l'image à partir du spectre d'amplitude M_2, on doit calculer la transformation inverse de Mellin-Fourier, calculer la première approximation $[\tilde{B}_0(2^n)]$ de l'image d'entrée et l'image de l'erreur (3.6) et continuer la décomposition pyramidale suivant les équations (3.7) à (3.12).

3.3. Applications pratiques de la caractérisation d'images en multi échelle par l'PRD

3.3.1. Recherche des images par le contenu

Comme déjà mentionné dans le paragraphe précédent une étude sur la recherche des images par le contenu avec la PRD a été menée dans [Kountchev&Manolova 2005 ; Aleksieva 2008 ; Manolova&Kountchev 2009 ; Manolova&Kountchev 2010]. Ces études sur l'application de PRD pour la recherche d'images, faites avec des différentes transformations et différentes formes de masques binaires ont permis de confirmer l'utilité de la PRD pour la description pertinente des images.

Le schéma bloc de l'algorithme général de recherche d'images par le contenu est illustré sur la figure 3.8.

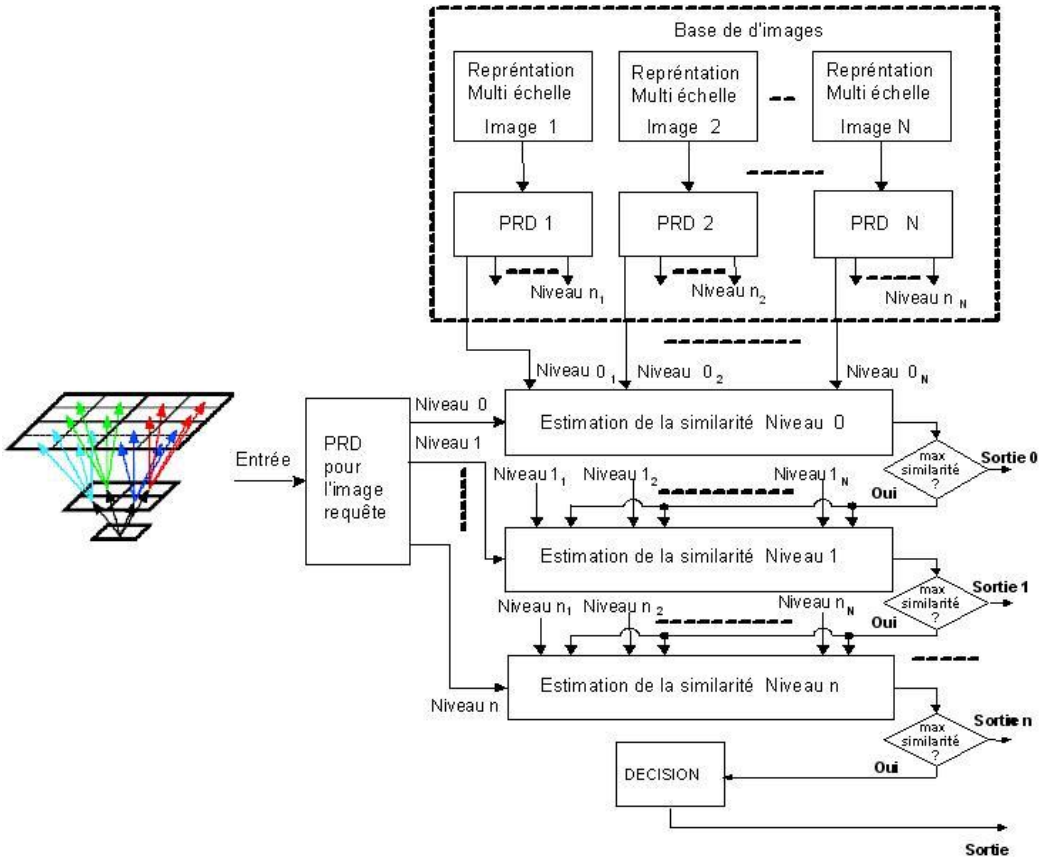


Fig. 3.8 Schéma bloc de l'algorithme de recherche de l'image par contenu dans une base de données des images en utilisant la représentation par la PRD avec n niveaux.

Cette méthode s'avérait utile dans les applications qui exigent des résultats rapides comme la reconnaissance des visages [Manolova&Kountchev 2010] et la reconnaissance d'objets [Kountchev&Manolova 2005 ; Aleksieva 2008 ; Manolova&Kountchev 2009]. L'utilisation du masque binaire réduit le nombre d'opérations mathématiques nécessaires et le temps de calcul est une fonction des dimensions du masque et du nombre de niveaux du PRD. Pour calculer la similitude, différents types de distances entre les images ont été utilisée en fonction du type de bases de données et la rapidité désirée de la recherche.

Cette application de la PRD serait la base de la construction de la matrice de dissimilitudes qui fait objet de cette thèse.

3.3.1. Construction de la matrice de dissimilitudes pour le classement

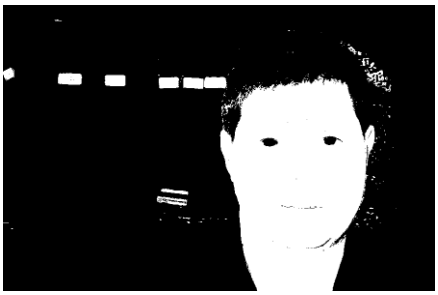
Pour l'application de classement d'une base d'images de visages qui sera présentée dans le chapitre 4, on dispose d'une base d'images de visages avec 450 images de 27 visages [http://www.vision.caltech.edu/html-files/archive.html].

Les photos des différentes personnes sont prises toujours en face mais les conditions d'éclairage et de fond sont différentes. Les visages sont ceux d'hommes et de femmes dont certaines images sont avec différentes expressions du visage. Le nombre d'images pour chaque visage n'est pas constant. Pour les résultats, présentés dans cette thèse, on a utilisé une partie de cette base pour le classement. La base de test utilisée est décrite avec détails dans le chapitre 4.

Afin d'isoler seulement le visage qui est notre région d'intérêt, on a implémenté un algorithme de segmentation de peau humaine qui est présenté en détails dans [Elgemmal&Muang 2009]. L'algorithme s'est avéré très performant et simple pour toutes sortes de couleurs de peau (blanche, noire et jaune) [Vezhnevets et al. 2003].

La figure 3.9 illustre le résultat de la segmentation de la peau, appliquée sur une des images de la base d'images.

(a)



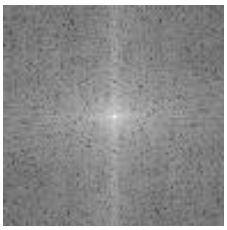
(b)

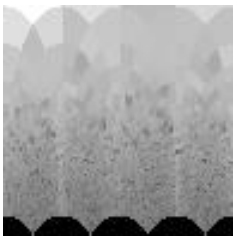
(c) *(d)*

Fig. 3.9 (a) Image d'origine de talle 896 x 592 ; (b) Image binaire résultante de la segmentation de la couleur de peau par seuillage dans le domaine RGB ; (c) Image binaire résultante de la filtration morphologique de l'image précédente pour l'élimination d'objets de petite taille ; (d) Cadrage du visage segmenté, taille 256 × 256 ;

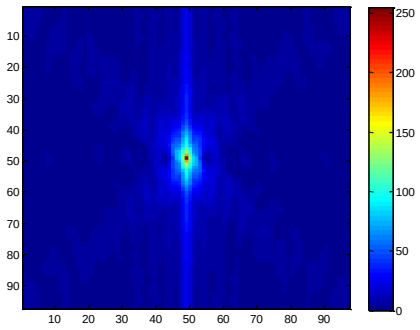
Suivant [Milanese et al. 1999] et [Derrode&Ghorbel 1999] qui aussi proposent d'utiliser la transformation de Mellin-Fourier comme moyen d'extraire une description de l'image, on a décidé d'utilisé cette transformation pour la construction de PRD. Derrode et ses collègues ont développé un système entier de recherche d'images par contenu. Les résultats de reconnaissance présentés dans [Ghorbel et al. 2006 ; Derrode et al. 2001 ; Derrode&Ghorbel 2004 ; Zana&Cesar 2006] encouragent le développement d'une méthode de classement, basée sur la description des images par la PRD avec la transformation de Mellin-Fourier.

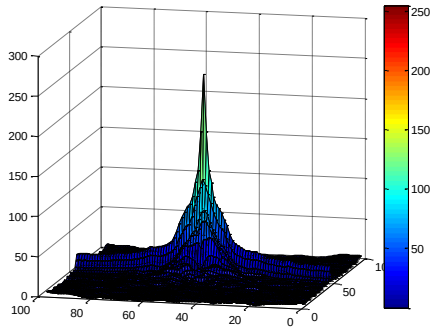
La figure 3.10 illustre la construction du premier niveau de la PRD de l'image en niveaux de gris le visage de la figure 3.9d.





(a) *(b)*





(c) *(d)*

(e) *(f)*

Fig. 3.10 (a) le spectre d'amplitude de la DFT de 3.11 d) centré après le masque binaire - coefficients retenus 96×96; (b) l'image de la transformation log-polaire avec l'interpolation au plus proche voisin; (c) spectre de l'amplitude de la deuxième DFT normalisé; (d) le spectre d'amplitude représenté en 3D (e) l'approximation de l'image d'entrée après la transformation inverse Mellin-Fourier et (f) image de la différence entre l'image de l'origine et l'approximation.

Le deuxième spectre d'amplitude (fig. 3.10c.d) est aussi symétrique par rapport au centre. Pour la construction du vecteur de caractéristiques de chaque visage qui serait invariant d'un certain degré à la translation, rotation et l'échelle, on utilise un nombre limité des coefficients du deuxième spectre d'amplitude parmi ceux avec des valeurs les plus significatives. La longueur choisie du vecteur caractéristique des coefficients a été choisie empiriquement à 22, sur la base de tests en classification. Une première étude dans un contexte de recherche d'image par le contenu a été déjà menée dans [Manolova&Kountchev 2010], uniquement sur le premier niveau (pour favoriser la rapidité), en utilisant une combinaison entre la distance euclidienne et la distance angulaire [Qianet al. 2004]. Pour le but d'une tâche de classement, présentée dans le chapitre 4, on a utilisé la distance de Minkowski fractionnaire. Cette distance a été choisie à cause de sa propriété de minimiser l'impact de grandes différences entre les valeurs des cordonnées des données à comparer. Cela permet de considérer comme différents les objets dont beaucoup de composantes montrent une différence modérée, et comme relativement semblables les objets dont peu de composantes sont très différentes qui correspond au cas présent. Au chapitre quatre, on montrera les résultats obtenus en classement avec les différents classifieurs étudiés durant cette thèse.

La difficulté pour la construction des autres niveaux de la PRD réside dans le fait que la transformation Mellin-Fourier est inversible. Un moyen de calculer l'inverse est proposé dans [Wanget al. 2006] mais c'est un processus lent et difficile à calculer car on doit sauvegarder l'information de la position de chaque coefficient du spectre d'amplitude de la première DFT et sa valeur après la transformation log-polaire. Afin de construire une application rapide, on a décidé de construire la représentation des images juste sur le premier niveau de la pyramide.

Cette méthode directe de caractérisation des images présente deux faiblesses. Tout d'abord les dimensions de l'espace de caractéristiques obtenu sont assez larges, on souffre de la « malédiction de dimensionnalité ». Deuxièmement, toute l'information spatiale est perdue lors

du passage dans l'espace fréquentiel par la transformation DFT et ce n'est pas possible d'isoler les coefficients qui décrivent des composantes spatialement distinctes dans l'image. Pour résoudre le problème de la dimensionnalité pour le classement on propose de calculer la distance entre les vecteurs caractéristique et appliquer un classifieur directement dans cet espace de dissimilitudes. Le problème devient alors de trouver une distance robuste qui tient compte de la deuxième faiblesse de la méthode mais ce serait assez complexe car la bande de fréquence et l'information de l'orientation sont fusionnées ensemble. La distance, utilisée pour cette application est la distance de Minkowski fractionnaire (1.4) choisie pour des raisons déjà mentionnées ci-dessus. C'est une distance semi-métrique.

3.4. Conclusion

La méthode de décomposition de l'image par la PRD s'est avérée intéressante dans le domaine de recherche d'image par contenu. Des tests ont été effectués sur des différentes bases de donnée ainsi qu'en modifiant l'algorithme par le choix de différentes transformations et formes de masque. Les résultats peuvent être consultés dans [Kountchev&Manolova 2005 ; Aleksieva 2008 ; Manolova&Kountchev 2009 ; Manolova&Kountchev 2010].

L'algorithme de la PRD trouve son application aussi dans la compression des images [Kountchev& Nakamatsu 2010] et dans le tatouage numérique [Kountchev et al. 2010].

La difficulté de la méthode de la caractérisation des images par la PRD avec la transformation Mellin-Fourier réside dans la transformation de Mellin-Fourier qui exige un temps beaucoup plus long que celui du calcul de la FFT, par exemple sous Matlab 7 pour une configuration avec Pentium M 2GHz avec 1.5 Go de RAM la FFT directe et inverse sur une image de dimensions 256 × 256 exige 0.06 secondes tandis que la MFT directe et inverse exige 1.7 secondes. Pour une grande base d'images de dimensions plus significatives ce fait peut devenir très contraignant. Alors on utilise juste le premier niveau de la décomposition pyramidale.

Le temps total d'extraction du visage de l'image de la base CALTECH de dimensions 896 × 592 et la construction de son vecteur caractéristique de longueur de 22 coefficients sous Matlab 7 pour une configuration avec Pentium M 2GHz avec 1.5 Go de RAM prend 3 secondes.

On a constaté que la représentation proposée résiste à des transformations plus complexes que la rotation, la translation et l'échelle, telles que les transformations affines induites par le mouvement autour de l'axe de la caméra par exemple, ainsi des petits changements du contraste dus à la normalisation des coefficients de la DFT. La représentation proposée capture les informations pertinentes de l'image. En comparaison avec les méthodes d'extraction des différentes caractéristiques correspondant à la couleur, la texture et des caractéristiques de forme, la méthode proposée emploie une approche holistique, et ne nécessite pas que l'utilisateur règle un poids approprié pour fusionner les traitements par caractéristiques.

En combinant l'algorithme de la PRD pour la construction du vecteur caractéristique des images et la méthode de classement par le « Coefficient de forme » sous ses différentes formes, présentées dans le chapitre précédant, on a établi une première version d'un système de classement des images en multi-classes qui inclut la construction de la description d'image, la

construction de la matrice de dissimilitudes et le classement basé sur cette matrice. Chaque étape de ce système peut être ajustée suivant les propriétés de la base d'images et l'application particulière.

Chapitre 4. Evaluation du comportement du classifieur « Coefficient de forme » sur des bases de données réelles

Dans ce chapitre on évaluera le comportement du classifieur « Coefficient de forme » sur des bases de données réelles, issues de différentes applications du monde réel. La précision de ce classifieur sous ses différents formes sera comparée aux celles des classifieurs, décrits dans le chapitre 1.

4.1. Introduction

Une collection de bases de données de dissimilitudes sera présentée et analysée en suivant une même méthodologie. Certaines des bases sont des bases publiques disponibles sur Internet. Les autres ont été soit, générées spécialement pour illustrer cette thèse, soit des résultats obtenus de recherches antérieurs.

Les résultats pour toutes les bases sont présentés de la même manière, en commençant par une brève description et des références de provenance d'origine, suivi par des caractéristiques des données et enfin en synthétisant les résultats de classement des différents classifieurs.

Les bases « Cat Cortex », « Protein », « Kimia », « Music », « USPS » et « UNIPEN » ont été prises directement de [http:// prtools.org/ disdatasets/] et [http:// lmb.informatik.uni-freiburg.de/people/haasdonk/datasets/distances.en.html]. Pour ces bases les résultats de classement avec les classifieurs de Pekalska et Haasdonk sont pris directement de leur recherche avec les références correspondantes. En effet, nous avons utilisé leur méthodologie de comparaison de performances. L'estimation du taux de reconnaissance est réalisée en « Leave One Out » pour ces bases. Ainsi ces classifieurs n'ont pas été re-implémentés et on dispose de leur meilleure estimation.

Pour toutes les expériences, deux environnements différents ont été utilisés. Pour les méthodes appelées « *Cs*-SVM », nous avons développé en langage C, en apportant des modifications au logiciel « SVM^{Light} » [http://svmlight.joachims.org], version 6.02, écrit en C par Thorsten Joachims. Pour toutes les autres méthodes, le code a été écrit en utilisant le Matlab® avec la bibliothèque « PRTools 4.0» [www.prtools.org], développé par le groupe « Pattern Recognition » de l'Université de Delft, et la bibliothèque « LS-SVMLab 1.5 », développée par l'équipe de J.A.K. Suykens de l'Université K.U.Leuven.

Afin de considérer de la même manière tous les classifieurs, nous avons pris en compte la répartition des données dans les classes c'est-à-dire les probabilités à priori d'appartenance d'une observation à classe, en introduisant la pondération des erreurs de classement [Weiss&Provost 2001]. Cela signifie qu'on donne plus d'importance d'une observation mal classée appartenant à une classe sous représentée qu'à une observation appartenant à une classe majoritaire.

Pour expliquer cette pondération, on donne un exemple de 3 classes : ω_1, ω_2 et ω_3.

On suppose que la classe ω_1 est majoritaire, puis la classe ω_2 et enfin la classe ω_3. Alors le nombre d'observations de chaque classe est noté n_1, n_2 et n_3, on a $n_1 > n_2 > n_3$. On appelle le coût d'erreur de type i, C_i, le coût de mal classer une observation de la classe i. Suivant notre exemple, on souhaite que le mauvais classement d'une observation de la classe ω_3 coûte plus cher que celui d'une observation de la classe ω_2 et que le coût de mauvais classement d'une observation de la classe ω_2 coûte plus cher que celui d'une observation de la classe ω_1.

On souhaite que :

$$n_1 \times C_1 = n_2 \times C_2 = n_3 \times C_3. \tag{4.1}$$

On pose :

$$C_1 = 1. \tag{4.2}$$

On obtient alors :

$$C_2 = \frac{n_1}{n_2} \text{ et } C_3 = \frac{n_1}{n_3}. \tag{4.3}$$

On obtient cette matrice des coûts :

$$\begin{pmatrix} 1 & C_1 & C_1 \\ C_2 & 1 & C_2 \\ C_3 & C_3 & 1 \end{pmatrix}. \tag{4.4}$$

La matrice de confusion a la forme suivante :

$$\begin{pmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \end{pmatrix}, \tag{4.5}$$

où n_{ij} est le nombre d'observations de la classe i, classées dans la classe j.

Au lieu de choisir les paramètres du classifieur qui optimisent le taux de reconnaissance obtenu de la matrice de confusion, on multiplie la matrice de confusion (4.5) terme à terme par la matrice de coût (4.4) [Koning 2001] :

$$\begin{pmatrix} 1\times n_{11} & C_1\times n_{12} & C_1\times n_{13} \\ C_2\times n_{21} & 1\times n_{22} & C_2\times n_{23} \\ C_3\times n_{31} & C_3\times n_{23} & 1\times n_{33} \end{pmatrix}. \tag{4.6}$$

On recalcule le nouveau taux d'erreur et on choisi les paramètres qui l'optimisent. Cette méthode de pondération de l'erreur sera utilisée dans les cas où une des classes sera sous représentée dans la base de données.

Concernant la caractérisation des bases de données, nous avons adopté le mode de présentation de [Pekalska&Duin 2009]. Ainsi, pour chaque matrice de dissimilitudes, on calcule un prolongement multidimensionnel, et on introduit aussi les quantités suivantes :

- Nombre *p* des valeurs propres positives et nombre *q* de valeurs propres négatives d'un total de *L* valeurs propres;
- Le rapport entre la valeur absolue de la valeur propre maximale négative et la valeur propre maximale positive (Negative Eigenratio):

$$\mathrm{NER} = \frac{\max_i |L_i < 0|}{\max_i (L_i > 0)}. \tag{4.7}$$

- Le rapport de la somme absolue de toutes les valeurs propres négatives et la somme de toutes les valeurs propres absolues (Negative Eigenfraction):

$$\mathrm{NEF} = \frac{\sum_{i:L_i<0} |L_i|}{\sum_i |L_i|}. \tag{4.8}$$

Ces deux rapports mesurent le comportement non euclidien des dissimilitudes. Les bases de données de dissimilitudes sont presque euclidiennes si leur NER est petit (≤ 0.1) et leur NEF aussi (≤ 0.03) [Pekalska&Duin 2009].

Le taux d'erreur pour les toutes les autres bases est calculé en utilisant la procédure « Leave One Out » (LOO) et par la procédure de demi-échantillonnage (« Half Sampling » HS). Pour ce faire, la base de données est divisée en deux au hasard en parties égales. Le premier ensemble est utilisé pour l'apprentissage du classifieur. Le deuxième ensemble est ensuite classé. La deuxième étape consiste à inverser le rôle des deux ensembles. Le taux d'erreur est la moyenne des deux erreurs de classement sur les deux ensembles de test. Cette procédure est répétée pour dix partitions aléatoires des deux ensembles.

Dans le cas des *Cs*-SVM et *Cs*-LSSVM, différentes valeurs des *C* ($C \in \{1.0e-3, \ldots, 1.0e+3\}$) ont été examinées et les meilleurs résultats sont sélectionnés. Le résultat de l'erreur de classement pour ces deux classifieurs affiché dans les tableaux est le meilleur obtenu avec deux ou trois contraintes pour le processus de l'optimisation.

4.2. Bases de données

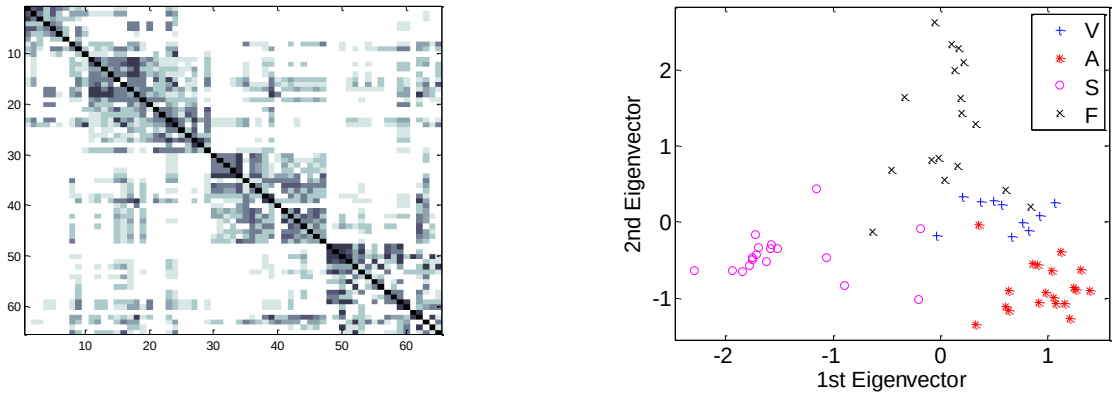
Base : Cat Cortex

Matrice de dissimilitudes décrivant les forces de connexions entre les 65 régions du cortex célébral du chat de quatre régions (classes) fonctionnelles : classe 1 (visuelle, V, 18 régions), classe 2 (auditive, A, 10 régions), classe 3 (somatosensoriel, S, 18 régions) et la classe 4 (frontolimbic, F, 19 régions). Les distances varient de 0 à 4 par pas de 0.5, elles sont les résultats d'un moyennage des distances initiales non symétriques.

Références :

- T. Graepel, R. Herbrich, P. Bollmann-Sdorra and K. Obermayer. Classification on pairwise proximity data. In Adv. Neural Info. Proc. Syst., volume 11, pp. 438-444, Cambridge, MA, 1999.MIT Press.
- J.W. Scannell, C. Blakemore and M.P. Young. Analysis of connectivity in the cat cerebral cortex. Journal of Neuroscience, 15(2):1463-1483, 1995.

65	Nombre d'objets
41, 23	Valeurs propres positives, négatives
0.208	NEF
0.272	NER
4	Classes avec taille [10 19 18 18]



(a) *(b)*

Fig. 4.1 (a) Matrice d'intensité des dissimilitudes Cat cortex ; (b) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 4 classes.

Tableau 4.1 : Erreur de classement en %, procédure LOO, pour la base totale

1NN dans l'espace de dissimilitudes	7.69
KNN dans l'espace de dissimilitudes	4.62 [4]

1NN dans l'espace pseudo-euclidien	12.3 [Duin&Pekalska 2009]
1NN dans l'espace euclidien *p*-dimensionnel, déterminé par les vecteurs propres positifs	6.2 [Duin&Pekalska 2009]
Cs, suivant 2.19	0

Tableau 4.2 : Erreur de classement en %, procédure LOO pour la base Cat Cortex, divisée en 4 sous matrices de manière « un cotre tous »

Base	Cat-cortex-V	Cat-cortex-A	Cat-cortex-S	Cat-cortex-F
1NN	4.62	3.08	4.62	3.08
KNN [K]	3.08 [3]	1.54 [3]	3.08 [2]	0 [8]
k^{nd}, suivant 1.69 avec $\beta = 2$ *	3.08	6.15	6.15	7.69
k^{pol}, suivant 1.69 avec $p = \{2,4,6,8\}$ *	1.54	3.08	3.08	6.15
k^{rfb}, suivant 1.69 *	0	4.62	3.08	4.62
k^{nd}, avec matrice de dissimilitudes régularisée *	3.08	4.62	3.08	4.62
k^{pol}, avec matrice de dissimilitudes régularisée *	3.08	4.62	3.08	4.62
k^{rfb}, avec matrice de dissimilitudes régularisée *	1.54	4.62	3.08	4.62
SVM lin *	4.62	1.54	3.08	1.54
Cs, suivant 2.24	1.54	1.54	3.08	1.54

* [Haasdonk&Bahlmann 2005]

Dans le cas de la base Cat Cortex à cause des dissimilitudes qui sont ordinales, la variance est presque nulle. Afin de pouvoir comparer le comportement du « Coefficient de forme » on utilise la règle linéaire 2.19. Le taux d'erreur est calculé avec pondération pour les résultats du tableau 4.2. Les classes de Cat Cortex sont bien regroupées parmi elles, comme on peut juger de la figure 4.1.b. et de l'erreur de classement de 1NN et KNN. La bonne performance du « Coefficient de forme » linéaire et des SVM linéaires indiquent que cette base de données est linéairement séparable.

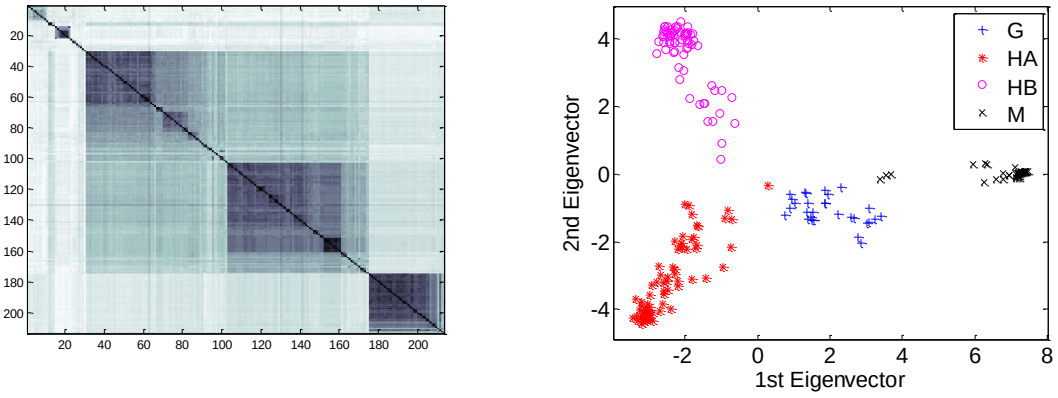
Base : Protein

Matrice de dissimilitudes de taille 213 × 213 comparant les séquences de protéines. Cette comparaison est basée sur le concept d'une distance évolutive. Il y a quatre classes: globine hétérogène (G), l'hémoglobine A (HA), l'hémoglobine B (HB) et de la myoglobine (M).

Références :

- T. Graepel, R. Herbrich, P. Bollmann-Sdorra, K. Obermayer, Classification on pairwise proximity data. In Advances in Neural Information System Processing vol. 11, pp. 438-444, 1999.
- T. Denoeux, T. and M.-H. Masson, EVCLUS: Evidential clustering of proximity data. IEEE Transations on Systems, Man and Cybernetics, vol. 34, pp. 95-109, 2004.

213	Nombre d'objets
205, 4	Valeurs propres positives, négatives
0.001	NEF
0.002	NER
4	Classes avec taille [30 72 72 39]



(a) *(b)*

Fig. 4.2 (a) Matrice d'intensité des dissimilitudes Protein ; (b) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 4 classes.

Tableau 4.3 : Erreur de classement en %, procédure LOO, pour la base totale

1NN dans l'espace de dissimilitudes	0
1NN dans l'espace pseudo-euclidien	1.9 [Duin&Pekalska 2009]
1NN dans l'espace euclidien *p*-dimensionnel, déterminé par les vecteurs propres positifs	1.9 [Duin&Pekalska 2009]
Cs géométrique sans paramètres	1.9
Cs géométrique avec $[\gamma_2, \delta_2]$, $[\gamma_3, \delta_3]$, $[\gamma_4, \delta_4]$	1.41, [1.4296, 0.4612], [1.3818, 0.5068], [0.5421, 0.4413]

Tableau 4.4 : Erreur de classement en %, procédure LOO pour la base Protein divisée en 4 sous matrices de manière « un cotre tous »

Base	Protein-HA	Protein-HB	Protein-M	Protein-G
1NN	0.89	1.77	0	0
KNN	0.89	1.77	0	0
k^{nd}, suivant 1.69 avec $\beta = 2$ *	0.89	3.54	0	0
k^{pol}, suivant 1.69 avec $p = \{2,4,6,8\}$ *	0.89	2.21	0	0.44
k^{rfb}, suivant 1.69 *	0.89	2.65	0	0
k^{nd}, avec matrice de dissimilitudes régularisée *	0.44	3.10	0	0
k^{pol}, avec matrice de dissimilitudes régularisée *	0.89	2.21	0	0.44
k^{rfb}, avec matrice de dissimilitudes régularisée *	0.89	2.65	0	0
SVM lin *	1.33	5.75	0	0
Cs-SVM	0	0.44	0	0
Cs-LSSVM	0	4.86	0	0

* [Haasdonk&Bahlmann 2005]

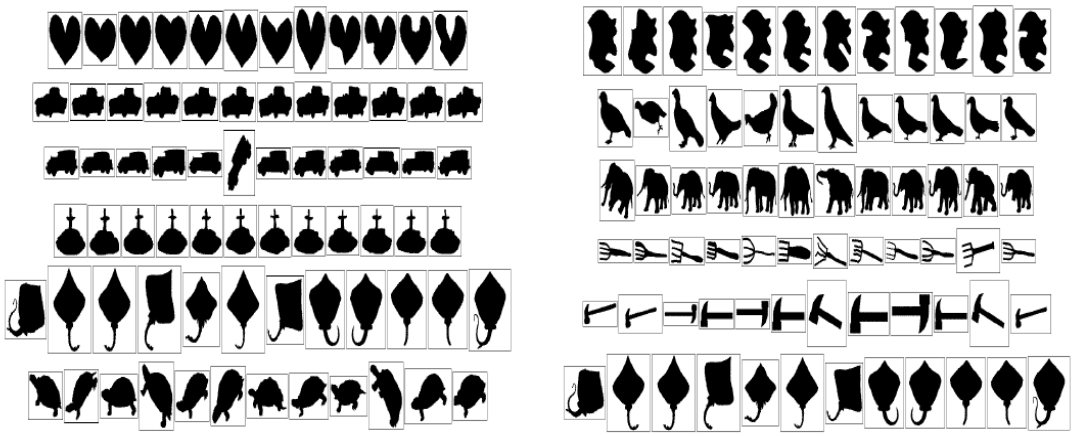
Les coefficients NEF et NER de la base Protein sont très petits ce qui veut dire que la dissimilitude choisie est presque euclidienne. La bonne performance du classifieur SVM linéaire de Haasdonk indique que cette base est linéairement séparable. Comme on peut voir du résultat du classement de *Cs* sans paramètres d'ajustement, il a très peu d'observations mal classées (4 dans ce cas) et dans ce cas-là la performance du *Cs* géométrique avec les tirages de Monte Carlo n'améliore pas le taux d'erreur à cause de la grande taille de l'espace, délimité par les quartiles des paramètres. Alors on a choisi une région autour du meilleur 6-uple des paramètres d'ajustement pour faire un nouveau tirage de points et on a réussi d'optimiser le taux d'erreur. La performance de *Cs*-SVM et *Cs*-LSSVM pour les bases Protein-HA et Protein-HB dépasse celle obtenue par Haasdonk.

Base : Kimia 1 et 2

Matrice de dissimilitudes calculée par la distance modifiée de Hausdorff entre des images binaires (figure 4.3) de 6 classes avec 12 images par classe.

Références :

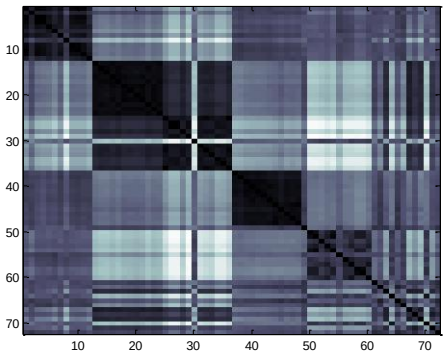
- T.B. Sebastian, P.N. Klein and B.B. Kimia. Recognition of Shapes by Editing Shock Graphs. In Proc. ICCV 2001, pp. 755-762, 2001.
- E. Pekalska, P. Paclik and R. Duin. A Generalized Kernel Approach to Dissimilarity Based Classification. Journal of Machine Learning Research, 2:175-211, 2001.



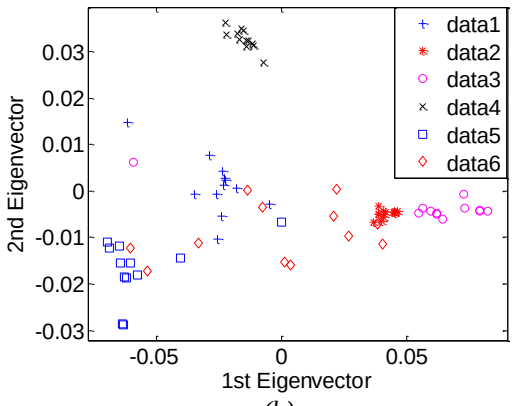
(a) *(b)*

Fig. 4.3 La base d'images des formes binaires Kimia (a) Kimia 1, (b) Kimia 2 [Pekalska et al. 2002]

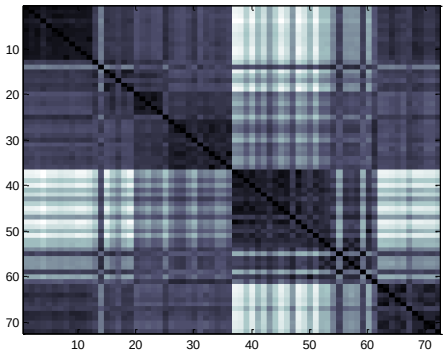
72 /72	Nombre d'objets /Kimia 1 et Kimia 2/
68, 4/69, 3	Valeurs propres positives, négatives
0.06/0.09	NEF
0.05/0.1	NER
6	Classes avec taille [12 12 12 12 12 12]



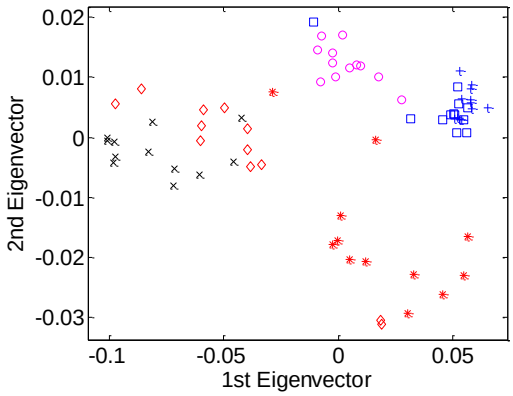
(a)



(b)



(c)



(d)

Fig. 4.4 (a), (c) Matrices d'intensité des dissimilitudes Kimia 1 et 2 ; (b), (d) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 6 classes de Kimia 1 et 2 respectivement.

Tableau 4.5 : Erreur de classement en %, procédure LOO, entre () est indiqué le nombre d'observations dans l'ensemble de représentation pour la méthode de Pekalska

Base	Kimia 1	Kimia 2
1NN	6.94	16.67
3NN	13.89	16.67
5NN	13.89	18.06
7NN	12.50	23.62
k^{nd}, suivant 1.69 avec $\beta = 2$ *	15.28	12.50
k^{pol}, suivant 1.69 avec $p = \{2,4,6,8\}$ *	11.11	9.72
k^{rfb}, suivant 1.69 *	4.17	9.72
k^{nd}, avec matrice de dissimilitudes régularisée *	4.17	8.33
k^{pol}, avec matrice de dissimilitudes régularisée *	4.17	8.33
k^{rfb}, avec matrice de dissimilitudes régularisée *	4.17	8.33
SVM lin *	8.33	8.33
SVM dans l'espace euclidien **	12.50	16.67
SVM dans l'espace pseudo euclidien **	11.11	9.72
FLD dans l'espace euclidien **	18.06	18.06
FLD dans l'espace pseudo euclidien **	13.89	8.33
RLD, R = ensemble d'apprentissage	8.33	6.94
FLD dans l'espace dissimilitudes avec R aléatoire **	12.50 (32)	6.94 (18)
Cs géométrique	5.56	12.05
Cs-SVM	6.94	16.67
Cs-LSSVM	8.33	13.89

*[Haasdonk&Bahlmann 2005]
**[Pekalska et al. 2002]

La base de données Kimia est utilisée dans [Pekalska&all 2001] afin de démontrer que la performance des 1NN et KNN pour une base de données non linéairement séparable peut être dépassée par d'autres types d'algorithmes de classement plus avancés, construits par plongement

dans l'espace euclidien ou pseudo euclidien ou dans l'espace des dissimilitudes. Pour interpréter les résultats, on doit noter que chaque observation mal classée ajoute au taux d'erreur 1.39%. Ce qui veut dire qu'un taux d'erreur de 4.17% représente 3 observations mal classées. On doit noter aussi que la distance utilisée pour cette base est sensible à la rotation alors il y a des points « outliers » qui représentent des images d'une classe qui diffèrent des autres de la classe par rotation (la classe 6 pour Kimia 1 de la fig. 4.4b est un bon exemple).

Le taux d'erreur de 1NN pour la base Kimia 1 indique qu'il y a 5 observations mal classées. Les classifieurs construits directement dans l'espace des dissimilitudes comme le classifieur linéaire de Fisher (FLD), le classifieur linéaire régularisé (RLD), et les classifieurs SVM et *Cs* sont plus performants que le 1NN. La performance de des classifieurs basés sur les SVM est très bonne ce qui est aussi expliqué par les petites valeurs des NEF et NER. La régularisation de la matrice de distances afin de la faire euclidienne est une méthode proposée par B. Haasdonk avec des résultats présentés dans [Haasdonk&Bahlmann 2005] avec les noyaux appropriés. Leur performance en taux d'erreur dépasse tous les autres classifieurs, plus particulièrement le noyau gaussien et le noyau polynomial sont très performants. Les *Cs*, *Cs*-SVM et *Cs*-LSSSVM s'avèrent aussi très performants, leurs taux d'erreur se retrouvent toujours parmi les meilleurs résultats des autres classifieurs proposés. On doit tenir compte que dans le cas de *Cs*-SVM et *Cs*-LSSSVM on ne fait aucune régularisation de la matrice de distances et aucun choix des prototypes n'est fait.

Pour la base Kimia 2, la performance de KNN se détériore rapidement avec l'augmentation du nombre des voisins. Dans ce cas aussi on peut voir que la performance des classifieurs dans l'espace des dissimilitudes est très bonne. Presque tous les classifieurs ont un meilleur succès que les 1NN et KNN.

Pour les deux groupes, la performance des classificateurs linéaires construits dans les espaces euclidiens ou pseudo euclidiens de Pekalska est plus faible que celles des classifieurs construits dans l'espace de dissimilitudes (quelques exceptions pour la base Kimia 2). Ce fait montre seulement que un classifieur linéaire peut ne pas être la meilleure solution pour ce problème, mais par exemple les noyaux non linéaires (gaussien et polynomial), proposés par Haasdonk donnent une meilleure performance.

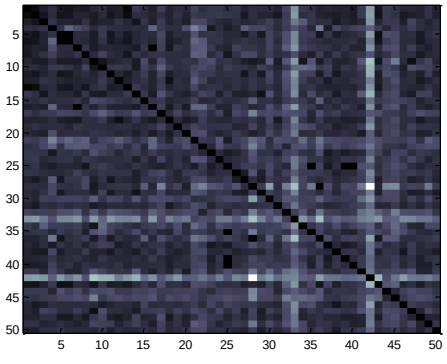
Base : Music EMD et PTD

Deux matrices de dissimilitudes entre les premières notes des compositions de musique. La première est calculée par la distance « Earth Mover's Distance » (EMD) et l'autre par « Proportional transportation distance » (PTD). La base de données totale contient 4 classes : 22 pièces de Georg Friedrich Händel, 28 pièces de Joseph Haydn, 27 pièces de Wolfgang Amadeus Mozart et 30 pièces de Gottfried Preyer. Les distances ont été normalisées par les soins de B. Haasdonk.

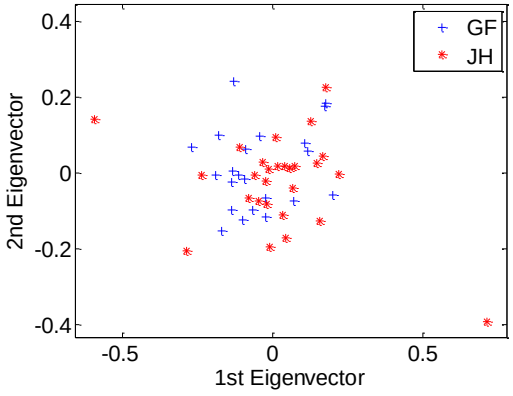
Référence :

- R. Typke, P. Giannopoulos, R.C. Veltkamp, F. Wiering and R. van Oostrum. Using transportation distances for measuring melodic similarity. In Proc. ISMIR 2003, pp. 107-114, 2003.

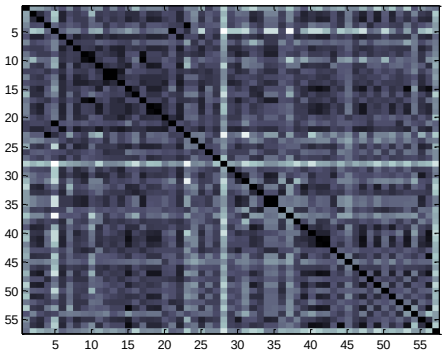
50 /57	Nombre d'objets /Musis-EMD1, PTD1 et Music-EMD2, PTD2/
31, 19/36, 21	Valeurs propres positives, négatives EMD1/EMD2
32, 18/35, 22	Valeurs propres positives, négatives PTD1/PTD2
0.28/0.28	NEF (EMD1/EMD2)
0.41/0.48	NER (EMD1/EMD2)
0.2/0.2	NEF (PTD1/PTD2)
0.31/0.23	NER (PTD1/PTD2)
4	Classes avec taille [22 28 27 30]



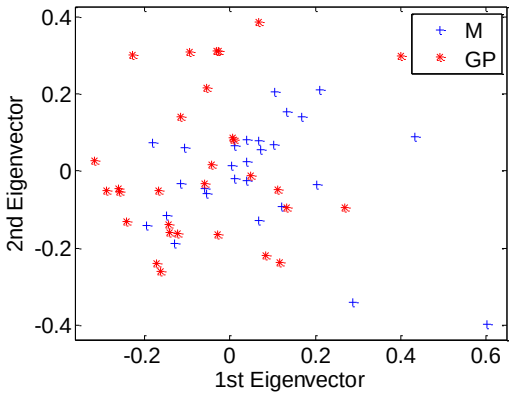
(a)



(b)

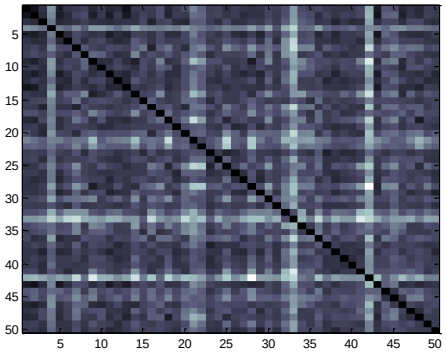


(c)

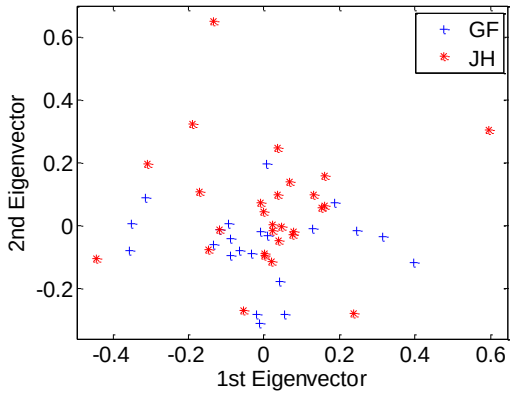


(d)

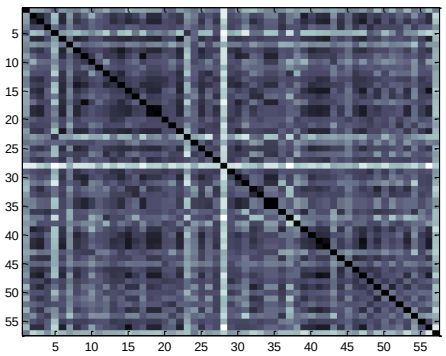
Fig. 4.5 (a), (c) Matrices d'intensité des dissimilitudes Music EMD1 et EMD2 ; (b), (d) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 2 classes par compositeur de Music EMD1 et EMD2 respectivement.



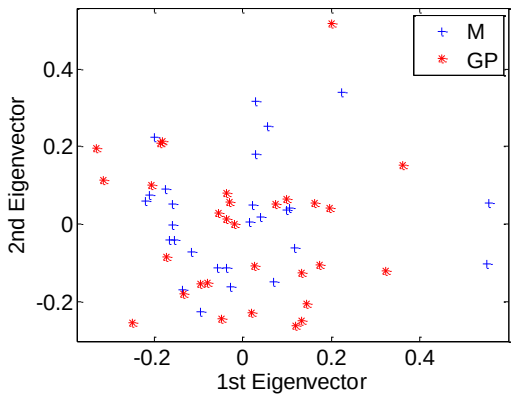
(a)



(b)



(c)



(d)

Fig. 4.6 (a), (c) Matrices d'intensité des dissimilitudes Music PTD1 et PTD2 ; (b), (d) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 2 classes par compositeur de Music PTD1 et PTD2 respectivement.

Tableau 4.6 : Erreur de classement en %, procédure LOO

Base	EMD1	EMD2	PTD1	PTD2
1NN	30	35.09	44	38.60
KNN [K]	26 [3]	29.82 [3]	36 [18]	38.60
k^{nd}, suivant 1.69 avec $\beta = 2$ *	40	42.11	34	31.58
k^{pol}, suivant 1.69 avec $p = \{2,4,6,8\}$ *	22	43.86	30	33.33
k^{rfb}, suivant 1.69 *	20	10.53	32	28.07
k_d^{nd}, avec matrice de dissimilitudes régularisée *	38	15.79	40	29.82
k^{pol}, avec matrice de dissimilitudes régularisée *	30	12.28	38	22.81
k^{rfb}, avec matrice de dissimilitudes régularisée *	30	10.53	32	17.54
SVM lin	44	21.05	40	20.82
Cs géométrique	34	36.84	36	35.09
Cs-SVM [vecteurs de support]	2 [12]	8.77 [21]	4 [12]	10.53 [17]
Cs-LSSVM	6	12.28	4	10.53

* [Haasdonk&Bahlmann 2005]

Les observations des résultats de Haasdonk montrent que le choix de la technique de régularisation de la matrice de dissimilitudes peut avoir un grand impact sur l'efficacité du classifieur. Suivant le choix du noyau, la structure non linéaire de l'espace de projection est complètement changée et un noyau peut être beaucoup plus approprié qu'un autre. Pour tous les ensembles MUSIC, la performance des deux modèles *Cs*-SVM et *Cs*-LSSVM dépasse celle de Haasdonk. Avec les modèles *Cs*-SVM et *Cs*-LSSVM, nous avons un seul type de recodage et le but de la stratégie d'apprentissage est de trouver le plan séparateur optimal par le meilleur choix des paramètres d'ajustement. Donc, si ce choix de recodage est approprié, l'efficacité du classifieur sera forte. C'est le cas pour les ensembles de MUSIC.

Des tests de normalité ont été aussi faits sur ces ensembles afin de vérifier si des données recodées suivent une loi normale car les résultats sur des distributions gaussiennes, présentées dans le chapitre 2, ont démontré une très bonne performance des classifieurs *Cs*-SVM et *Cs*-LSSVM vis-à-vis ce type de données. Pour ce test on a choisi celui de Lilliefors [Rakotomalala 2008] car le nombre d'échantillons est petit et parce que ce test est une variante du test de Kolmogorov-Smirnov où les paramètres de la loi (moyenne et variance) sont estimés à partir des données. La statistique du test est calculée de la même manière [Gosling 1995].

Les résultats de ce test et les droites de Henry sur les données recodées des quatre ensembles de MUSIC ne rejettent pas l'hypothèse nulle ce qui suppose la normalité de la population ou ce fait peut refléter aussi un manque de preuves solides contre l'hypothèse nulle en raison de la petite taille des échantillons. Ce résultat donne une explication probable de la meilleure performance des modèles *Cs*-SVM et *Cs*-LSSVM.

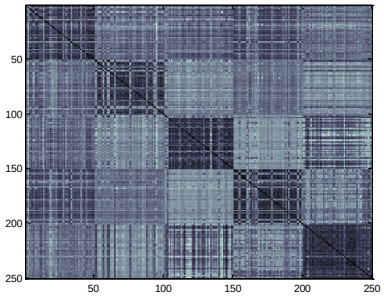
Base : UNIPEN

Les deux bases représentent une petite partie extraite de la grande base de données de caractères manuscrits du projet UNIPEN. Haasdonk a utilisé plus spécifiquement la partie 1c de la section Train-R01/V07 de la base avec des caractères minuscules. La dissimilitude utilisée est la distance tangentielle. Pour des raisons d'efficacité du classement multi-classes Haasdonk a tiré au hasard 50 échantillons de chacune des 5 classes de caractères de 'a' à 'e' dans la base entière pour chacune des deux matrices de dissimilitudes, utilisée dans cette étude.

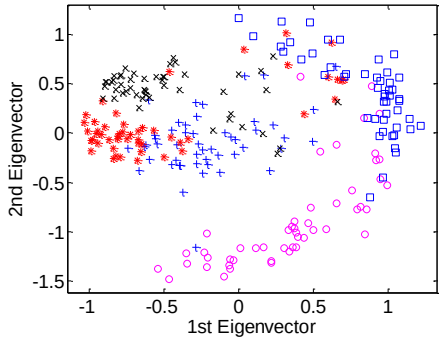
Références :

- B. Haasdonk and D. Keysers. Tangent Distance Kernels for Support Vector Machines. In Proc. of the 16th Int. Conf. on Pattern Recognition, vol. 2, pp. 864-868, IEEE, 2002.
- D. Keysers, W. Macherey, H. Ney, and J. Dahmen. Adaptation in Statistical Pattern Recognition Using Tangent Vectors. In IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26, Number 2, pp. 269-274, February 2004.

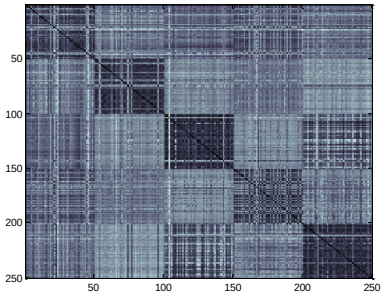
250 /250	Nombre d'objets /UNIPEN1 et UNIPEN2/
134, 116/135, 115	Valeurs propres positives, négatives UNIPEN1 et UNIPEN2
0.31/0.31	NEF (UNIPEN1/UNIPEN2)
0.2/0.2	NER (UNIPEN1/UNIPEN2)
5	Classes avec taille [50 objets par classe]



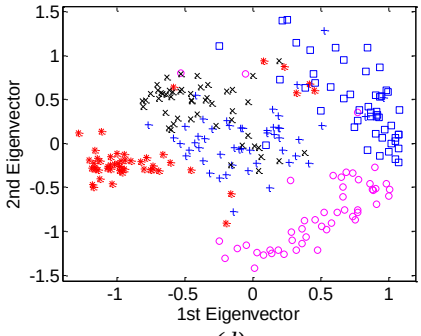
(a)



(b)



(c)



(d)

Fig. 4.7 (a), (c) Matrices d'intensité des dissimilitudes UNIPEN1 et UNIPEN2 ; (b), (d) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 5 classes de UNIPEN1 et UNIPEN2 respectivement.

Tableau 4.7 : Erreur de classement en %, procédure LOO

Base	UNIPEN1	UNIPEN2
1NN	6.80	8.40
KNN [K]	6 [4]	7.60 [3]
k^{nd}, suivant 1.69 avec $\beta = 2$ *	14.4	10.8
k^{pol}, suivant 1.69 avec $p = \{2,4,6,8\}$ *	6	7.60
k^{rfb}, suivant 1.69 *	5.20	6
k^{nd}, avec matrice de dissimilitudes régularisée *	8.40	7.60
k^{pol}, avec matrice de dissimilitudes régularisée *	5.20	6.40
k^{rfb}, avec matrice de dissimilitudes régularisée *	4.40	5.60
SVM lin	8	9.60
Cs géométrique	8.40	9.60
Cs-SVM	4.40	5.60
Cs-LSSVM	5.20	7.60

* [Haasdonk&Bahlmann 2005]

Pour les ensembles UNIPEN1 et UNIPEN2 on utilise la procédure de classement « un contre tous » pour les modèles *Cs*-SVM et *Cs*-LSSVM. Cette procédure est très performante [Rifkin&Klautau 2004] et [Duan&Keerthi 2005]. Le choix de recodage proposé est approprié pour cette base, l'efficacité du classement pour ces deux modèles est très bonne. On peut aussi utiliser le modèle *Cs* géométrique dont la performance reste dans l'intervalle des valeurs de Haasdonk car c'est une procédure qui gère toutes les classes à la fois au lieu de diviser la matrice chaque fois en deux parties.

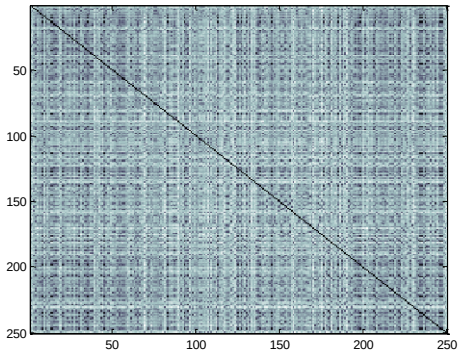
Base : USPS-TD

Les ensembles de données USPS-TD sont un extrait de la base USPS des chiffres manuscrits. La mesure de dissimilitude est la distance tangentielle. Afin d'obtenir un problème de classement binaire, Haasdonk fait une séparation des chiffres de 0-4 pour classe 1, 5-9 pour classe 2.

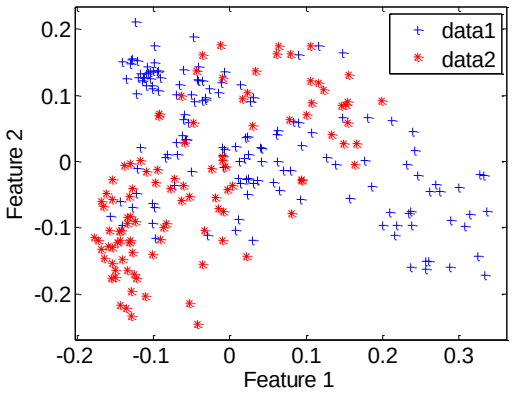
Références :

- B. Haasdonk and D. Keysers. Tangent Distance Kernels for Support Vector Machines. In Proc. of the 16th Int. Conf. on Pattern Recognition, vol. 2, pp. 864-868, IEEE, 2002.
- D. Keysers, W. Macherey, H. Ney, and J. Dahmen. Adaptation in Statistical Pattern Recognition Using Tangent Vectors. In IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26, Number 2, pp. 269-274, February 2004.

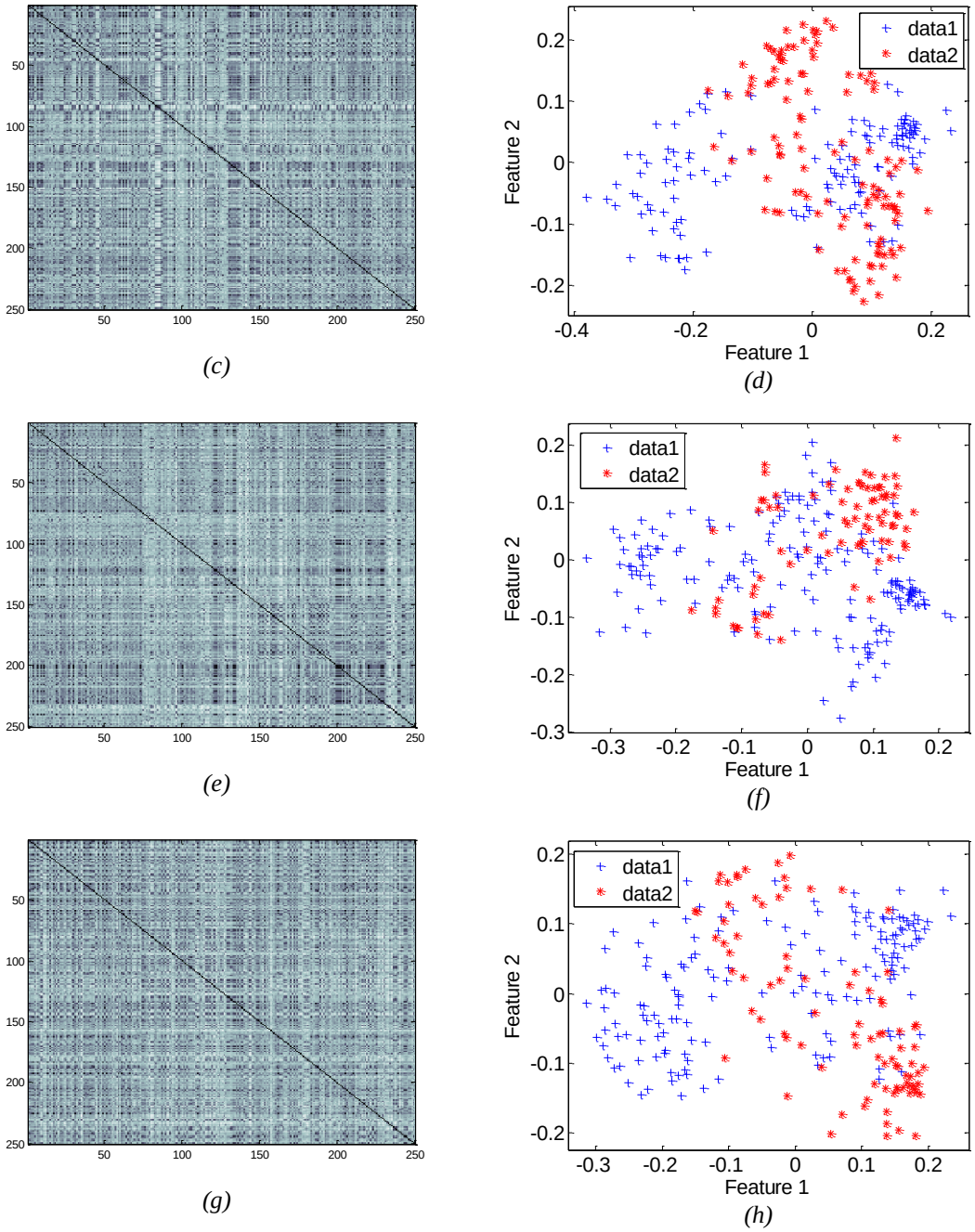
250	Nombre d'objets / USPS-TD1, USPS-TD2, USPS-TD3, USPS-TD4/
149, 101/150, 100	Valeurs propres positives, négatives USPS-TD1/USPS-TD2
151, 99/152, 98	Valeurs propres positives, négatives USPS-TD3/USPS-TD4
0.15/0.15	NEF (USPS-TD1/USPS-TD2)
0.15/0.15	NEF (USPS-TD3/USPS-TD4)
0.07/0.08	NER (USPS-TD1/USPS-TD2)
0.07/0.06	NER (USPS-TD3/USPS-TD4)
2	Classes avec taille USPS-TD1 [146 104], USPS-TD2 [133 117], USPS-TD3 [169 81], USPS-TD4 [160 90]



(a)



(b)



(c) *(d)* *(e)* *(f)* *(g)* *(h)*

Fig. 4.8 (a), (c), (e), (g) Matrices d'intensité des dissimilitudes USPS-TD1, TD2, TD3 et TD4 ; (b), (d), (f), (h) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 2 classes de USPS-TD1, TD2, TD3 et TD4 respectivement.

Tableau 4.8 : Erreur de classement en %, procédure LOO

Base	USPS-TD1	USPSTD2	USPS-TD3	USPS-TD4
1NN	4.40	5.20	5.20	2.80
KNN [K]	4.40	5.20	4.40 [3]	2.80
k^{nd}, suivant 1.69 avec β = 2 *	10.4	14.4	12.8	10.8

k^{pol}, suivant 1.69 avec p = {2,4,6,8} *	5.20	7.60	6.80	6.40
k^{rfb}, suivant 1.69 *	3.20	2.40	4	3.20
k^{nd}, avec matrice de dissimilitudes régularisée *	4	7.20	6.80	6.40
k^{pol}, avec matrice de dissimilitudes régularisée*	4	4	4.40	5.20
k^{rfb}, avec matrice de dissimilitudes régularisée *	3.20	2.40	4	3.20
SVM lin	6.80	6	6.80	7.20
Cs géométrique	6.40	5.60	4.80	5.60
Cs-SVM	3.20	2.80	1.20	1.20
Cs-LSSVM	2.40	2	0.8	1.20

* [Haasdonk&Bahlmann 2005]

Pour les ensembles USPS la performance des modèles *Cs*-SVM et *Cs*-LSSVM est meilleure que les SVM de Haasdonk et des KNN, ce qui indique que le recodage est approprié dans ce cas. Les faibles valeurs des coefficients NER indiquent un comportement proche de l'euclidien ce qui explique aussi les bons résultats des trois modèles proposés et du classifieur KNN.

Base : Scènes Jeffrey

La base d'images naturelles est composée de 473 images, classées en 4 classes : les scènes urbaines, les scènes d'intérieur, les paysages fermés (de type forêt, montagne…), et les paysages ouverts (de type désert et plage). Les données sont issues de travaux effectués au GIPSA sur la catégorisation d'images. Il a été montré que le spectre d'amplitude des images naturelles porte une information sur le contexte sémantique de l'image relativement à des catégories sémantiques de type :

- Scènes d'intérieur ;
- Scènes urbaines ;
- Plage, désert …;
- Montagnes, forêt …;

Ces catégories se justifient en analysant l'allure de décroissance de l'amplitude spectrale en fonction de la fréquence, en module et en orientation. En analysant le spectre de puissance des images naturelles, on remarque que l'on peut faire une classification suivant la forme des spectres en 4 catégories : (i) spectres à orientation horizontale dominante, (ii) spectre en "croix" (dominante verticale et horizontale), (iii) spectre plutôt isotrope, (iv) spectre à dominante verticale. Ces formes sont illustrées à la figure 4.9 avec une image de chaque catégorie sémantique associée. Ici pour chaque image, dans une basse résolution fréquentielle, on obtient par calcul d'orientation locale un histogramme représentatif de la présence de motifs orientés. L'image est donc simplement caractérisée par un histogramme d'orientation et deux images sont comparées entre elles par la divergence de Kullback – Leibler.

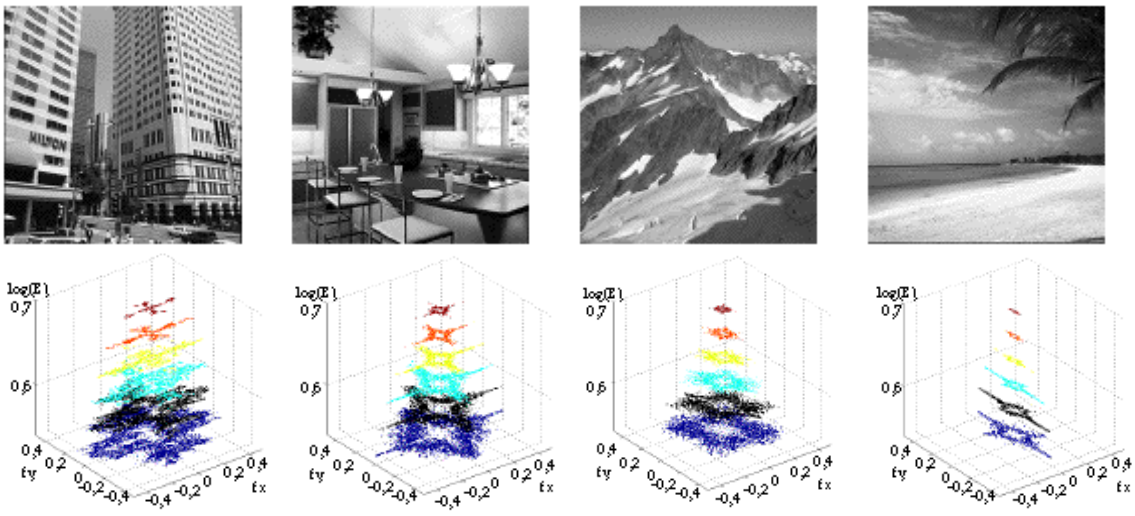
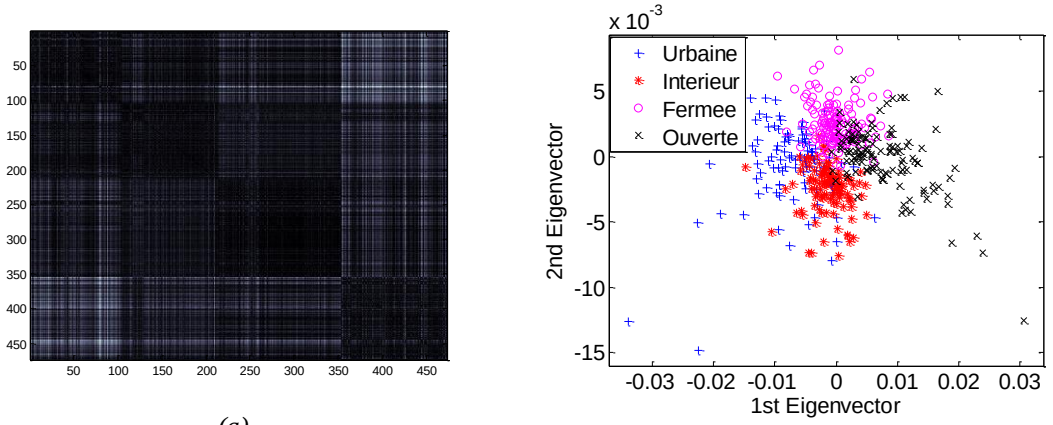


Fig. 4.9 Forme spectrale d'une scène urbaine (a), d'une scène intérieure (b), d'une scène fermée (c) et d'une scène ouverte (d) [Dugué&Oliva 2000].

Références :

- P. Ladret et A. Guerin- Dugue. Categorisation and retrieval of scene photographs from a JPEG compressed database, Pattern Analysis and Application 4 (2-3), pp. 185-199, 2001.
- A. Guérin-Dugué, A. Oliva. Classification of Scene Photographs from Local Orientations Features, Pattern Recognition Letters 21 (13-14), pp. 1135-1140, 2000.

473	Nombre d'objets
471, 2	Valeurs propres positives, négatives
0.24	NEF
0.26	NER
4	Classes avec taille [102 110 140 121]



(a) *(b)*

Fig. 4.10 (a) Matrice d'intensité des dissimilitudes Scènes Jeffrey ; (b) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 4 classes Urbaine, Intérieur, Fermée et Ouverte respectivement.

Tableau 4.9 : Erreur de classement en %, procédure LOO, pour la base totale, pour les modèles *Cs*-SVM et *Cs*-LSSVM est appliqué la méthode « un contre tous »

1NN/KNN	28.12/20.72 [18]
Cs géométrique [γ_2, δ_2], [γ_3, δ_3], [γ_4, δ_4]	20.51 [0.583, 0.977], [2.286, 0.851], [1.581, 0.972]
Cs-SVM	19.24
Cs-LSSVM	20.72

Tableau 4.10 : La moyenne de l'erreur de classement en %, procédure HS, pour la base totale. Les valeurs entre () affichent la variance de l'erreur

KNN	24.5 (3.4)
Cs géométrique	23.1 (1.9)

Tableau 4.11 : La moyenne de l'erreur en %, procédure HS, pour la base divisée en deux pour la procédure « Un cotre Tous » des SVM. Les valeurs entre () affichent la variance de l'erreur

Base	Urbaine	Intérieur	Ouverte	Fermée
Cs-SVM	10.64 (1.7)	13.05 (1.4)	9.73 (1.9)	11.84 (1.6)
Cs-LSSVM	11 (2.14)	16.78 (2)	11.42 (2.64)	12.56 (2.33)

En utilisant le classifieur *Cs* géométrique pour cette base d'images avec 4 classes on a 6 paramètres d'ajustement (γ_2; δ_2), (γ_3; δ_3), (γ_4; δ_4), la classe de référence est la première classe (scènes urbaines). La difficulté pour le modèle *Cs* géométrique dans ce cas est d'atteindre le bon intervalle de recherche pour les simulations de Monte Carlo de manière efficace. On a observé que les intervalles de recherche des paramètres γ_c sont assez larges. Afin d'accélérer la procédure

on a utilisé les modèles *Cs*(γ_c, 1) et *Cs*(1, δ_c) pour trouver des meilleurs intervalles de recherche des paramètres d'ajustement. En limitant de cette façon les intervalles de recherche on diminue considérablement le temps de calcul et on s'approche du temps de calcul du KNN.

Les modèles *Cs*-SVM et *Cs*-LSSVM en LOO s'avèrent très performants. En ce qui concerne le classement binaire « un contre tous », pour la classe urbaine on a un taux d'erreur *Cs*-SVM/*Cs*-LSSVM de 9.73/10.15%, respectivement pour la classe intérieure – 10.36/15.43%, pour la classe ouverte – 8.67/10.64% et pour la classe fermée – 10.36/11.84%. En effet, les résultats correspondent à la réalité: les scènes urbaines avec une orientation prédominante verticale sont facilement distinguées. Les scènes intérieures sont mélangées avec les scènes urbaines et plus difficilement distinguées. Les scènes ouvertes avec une orientation prédominante horizontale sont aussi bien classées mais parmi ces images on peut trouver des images qu'on peut difficilement séparer de la classe des scènes fermées. Ces résultats sont confirmés par la procédure en demi-échantillonnage. Le nombre d'itérations pour l'optimisation du modèle *Cs*-SVM exigé afin d'arriver à une solution optimale est autour de 7000 pour chaque cas.

Base : Chiffres manuscrits 3 et 8

Cette base d'images binaires a été aimablement mise en disposition par E. Pekalska. Elle contient 400 images binaires des chiffres manuscrits 3 et 8 (200 images par chiffre) et fait partie de la base NIST. Un exemple extrait de la base est illustré à la figure 4.11. Pour ces images de taille 128 × 128, on a construit la matrice de dissimilitudes en utilisant la distance modifiée de Hausdorff pour des contours binaires.

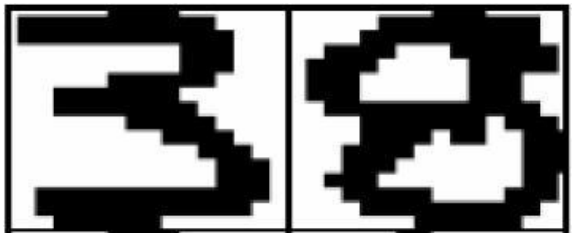
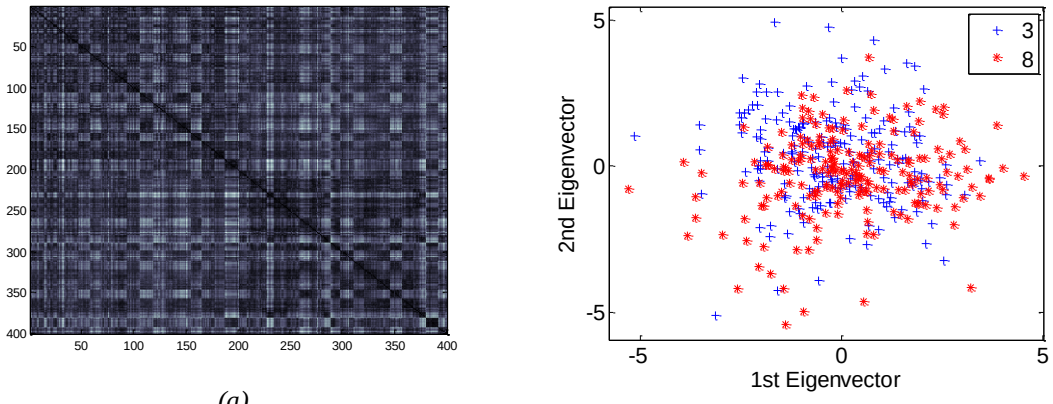


Fig. 4.11 Exemple des images binaires pour des chiffres manuscrits 3 et 8

Références :

- E. Pękalska, P. Paclik, and R.P.W. Duin. A Generalized Kernel Approach to Dissimilarity-based Classification, Journal of Machine Learning Research, Special Issue on Kernel Methods, vol. 2, no. 2, 175-211, 2002.
- C.L.Wilson and M.D. Garris. Handprinted character database 3. Technical report, National Institute of Standards and Technology, February 1992.

400	Nombre d'objets
171, 229	Valeurs propres positives, négatives
0.3	NEF
0.2	NER
2	Classes avec taille [200 200]



(a) *(b)*

Fig. 4.12 (a) Matrice d'intensité des dissimilitudes Chiffres Manuscrits 3 et 8; (b) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 2 classes 3 et 8 respectivement.

Tableau 4.12 : Erreur de classement en %, procédure LOO, pour la base totale

1NN/KNN [14]	7.75/7.0
Cs géométrique [γ_2, δ_2]	3.75 [1.01, 0.9915]
Cs-SVM	3
Cs-LSSVM	3

Tableau 4.13 : Erreur de classement en %, procédure HS, pour la base totale

1NN/KNN	11 (4)
Cs géométrique	5.5 (2.13)
Cs-SVM	3 (1.38)
Cs-LSSVM	3.75 (1.26)

Les modèles *Cs*(γ_c, δ_c,), *Cs*-SVM et *CS*-LSSVM surpassent le classifieur KNN. La différence est significative (test *t* de Student, $p > 0{,}001$). Une observation mal classée apporte 0.5 à l'erreur de classement alors l'erreur minimale des trois modèles du « Coefficient de forme » correspond à 6 observations mal classées. Les valeurs des paramètres d'ajustement pour le *Cs*(γ_c, δ_c,) sont proches de 1 pour les deux classes ce qui suivant les résultats, présentés dans le chapitre 2, indique que les deux classes ont une forme sphérique. Pour ce problème de classement les classifieurs 1NN et KNN sont trop simples pour arriver à une bonne performance.

Base : Signaux synthétiques de contrôle (SSC)

Cette base de données est composée de 6 classes avec 100 éléments par classe de signaux synthétiques de contrôle dont la dissimilitude est la distance DTW (exemple des signaux illustré à la fig. 4.13). L'algorithme DWT est choisi pour comparer ces séries chronologiques en raison que la distance euclidienne n'est pas en mesure d'atteindre une précision parfaite. En particulier, les couples de classes qui sont souvent confondus, sont Normal / Cyclique, Decreasing trend / Downward shift et Upward shift / Increasing trend.

600	Nombre d'objets
171, 229	Valeurs propres positives, négatives
0.3	NEF
0.2	NER
6	Classes avec taille [100 par classe]

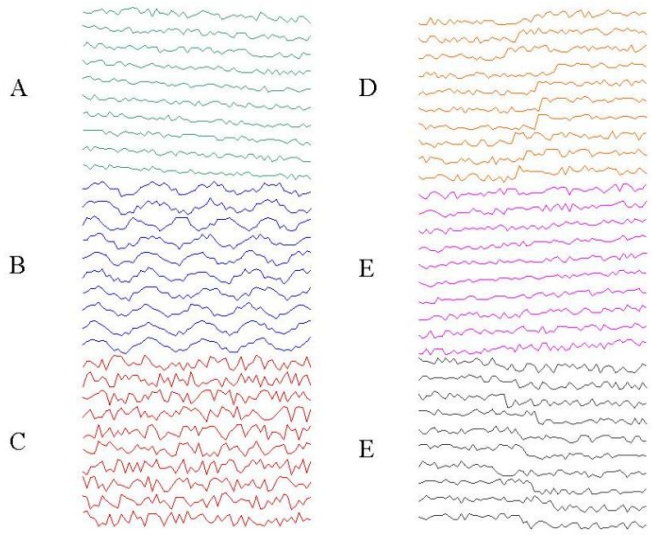
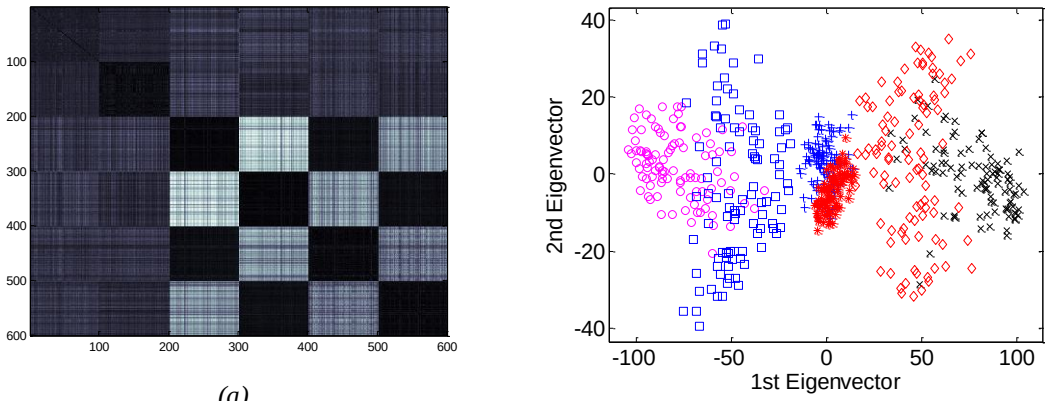


Fig. 4.13 Base de données des signaux synthétiques de contrôle avec 6 classes : A- Decreasing trend, B- Cyclic, C- Normal, D- Upward shift – D, F- Downward shift, E- Increasing trend.

Référence :

- D.T. Pham and A.B. Chan "Control Chart Pattern Recognition using a New Type of Self Organizing Neural Network" Proc. Instn, Mech, Engrs. Vol. 212, No 1, pp. 115-127, 1998.



(a) *(b)*

Fig. 4.14 (a) Matrice d'intensité des dissimilitudes des Signaux synthétiques de contrôle; (b) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 6 classes respectivement.

Tableau 4.14 : Erreur de classement en %, procédure LOO, pour la base totale

1NN/KNN [3]	2.33/2.0
Cs géometrique	1.67
Cs-SVM	1.33
Cs-LSSVM	2.83

Les expériences sur la base des signaux de contrôle synthétiques démontrent de très bons résultats du classement même si on a un problème de 6 classes où nous avons 5 couples des paramètres d'ajustement à optimiser dans un espace $\Re^{10}$ pour le modèle *Cs* géométrique. Dans ce problème de classement, on a utilisé aussi les modèles $Cs(\gamma_c, 1)$ et $Cs(1, \delta_c)$ pour trouver des meilleurs intervalles de recherche des paramètres d'ajustement. La dissimilitude de DWT utilisée pour ce problème s'avère très bien choisie. Chaque observation mal classée correspond à un taux d'erreur de 0.1663%. Dans le cas du modèle *Cs*-SVM alors on a juste 8 observations mal classées sur les 600 observations au total. Les expériences menées en « half sampling » ont montré que l'erreur de classement des trois modèles reste stable autour de l'erreur en LOO avec une variance faible. Les classes sont bien séparées et même si les coefficients NEF et NER sont élevés et on a beaucoup de valeurs propres négatives, les trois modèles du « Coefficient de forme » restent très performants.

Base : Signatures Génomiques

Cette base de données a été aimablement mise en disposition par S. Lespinat. Il s'agit d'un ensemble de données de 3 classes d'images des signatures génomiques de 2046 espèces, le nombre d'espèces est irrégulièrement réparti entre les trois classes. Le domaine de l'Archebacteria est le plus petit avec seulement 62 spécimens. Pour des raisons d'efficacité du classement dans cette étude cette classe ne sera pas prise en compte. Les autres deux classes sont respectivement Bacteria et Eycaryota. Un extrait de la base est illustré à la fig. 4.15. Pour les tests de classement, nous disposons de 4 différentes mesures de distance - distance euclidienne, la distance curviligne, distance euclidienne avec correction de l'équilibre et de la distance curviligne avec correction de l'équilibre. Le processus d'équilibrage est expliqué avec des détails dans le manuscrit de S. Lespinat.

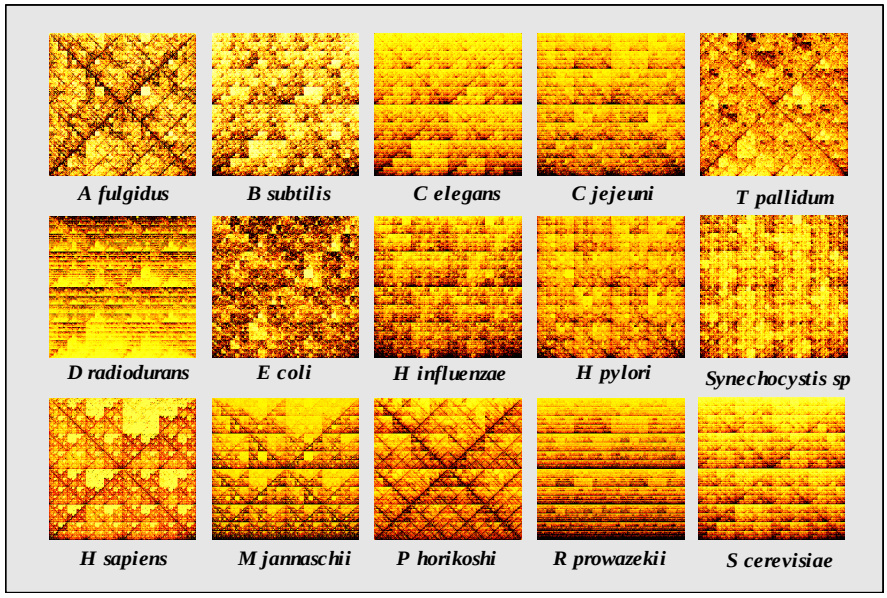
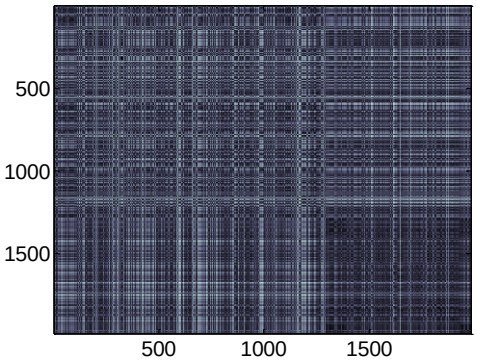


Fig. 4.15 Extrait de la base d'images des signatures génomiques [Lespinat 2006]

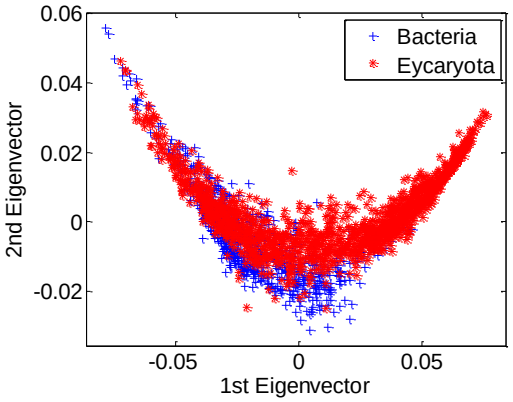
Référence :

- Sylvain Lespinat. Style du génome exploré par analyse textuelle de l'ADN, thesis 2006, Université Pierre et Marie Curie (Paris VI).

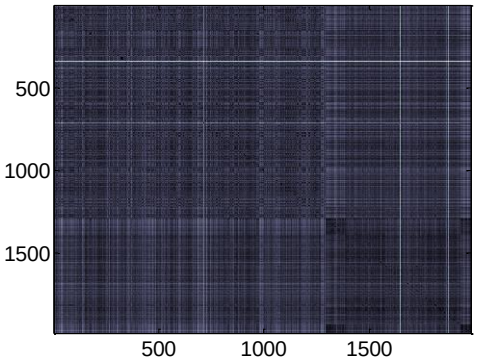
1984	Nombre d'objets
1043, 941/1106, 940	Valeurs propres positives, négatives DCurv/DCurv_ce
0.19/0.27	NEF DCurv/DCurv_ce
0.03/0.08	NER DCurv/DCurv_ce
2	Classes avec taille [697 1287]


500
1000
1500
500
1000
1500

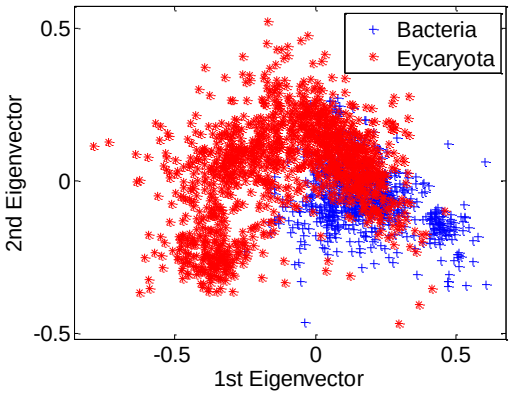
(a)


Bacteria
Eycaryota
0.06
0.04
0.02
0
-0.02
2nd Eigenvector
-0.05
0
0.05
1st Eigenvector

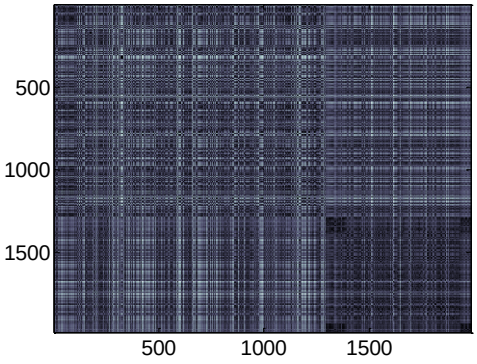
(b)


500
1000
1500
500
1000
1500

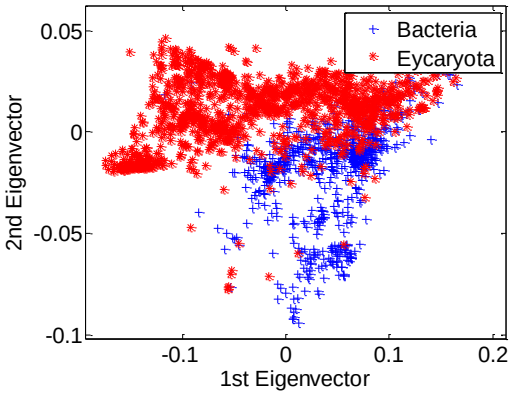
(c)


Bacteria
Eycaryota
0.5
0
-0.5
2nd Eigenvector
-0.5
0
0.5
1st Eigenvector

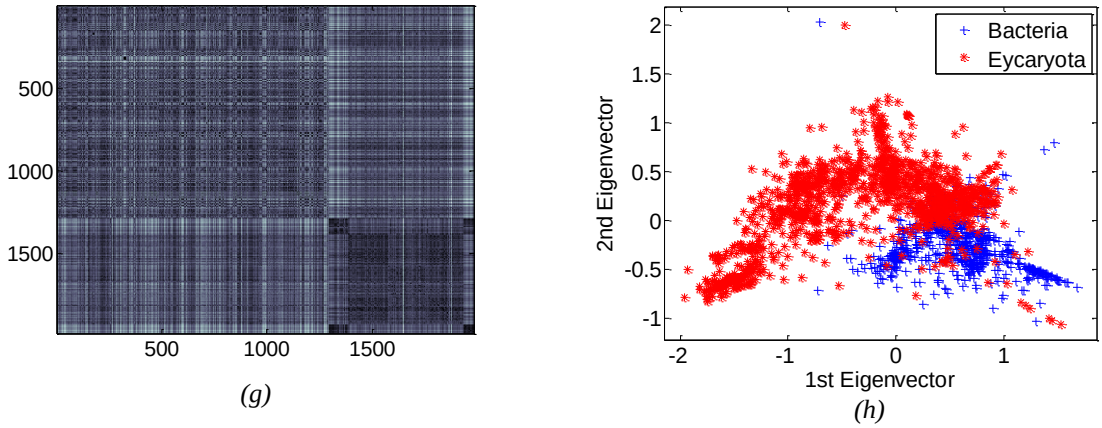
(d)


500
1000
1500
500
1000
1500

(e)


Bacteria
Eycaryota
0.05
0
-0.05
-0.1
2nd Eigenvector
-0.1
0
0.1
0.2
1st Eigenvector

(f)



(g) *(h)*

Fig. 4.16 (a), (c), (e), (g) Matrices d'intensité des dissimilitudes des Signatures Génomiques DEuc, DEuc_ce, DCurv, DCurv ce; (b), (d), (f), (h) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 2 classes respectivement de DEuc, DEuc_ce, DCurv, DCurv_ce.

Tableau 4.16 : Erreur de classement en %, procédure LOO, pour la base totale, en [] est indiqué le nombre des vecteurs de support

Base	DEuc	DEuc_ce	DCurv	DCurv_ce
1NN/KNN	10.84/10.84	6.25/5.54 [5]	7.06/5.6 [3]	5/4.18 [7]
Cs géometrique	15.02	12.86	11.30	9.40
Cs-SVM	12.99 [590]	5.23	7.41	4.43
Cs-LSSVM	15.47	6.46	7.81	6.19

Suivant [Lespinat 2006] les séquences d'ADN peuvent être considérées comme des textes écrits dans un alphabet de 4 lettres. Il s'est inspiré des techniques de l'analyse textuelle qui permettent de les caractériser à partir de fréquences d'apparition de courtes suites de caractères. L'ensemble des fréquences des mots d'une longueur donnée est appelé « signature génomique ». L'analyse des signatures génomiques se confronte rapidement à des limitations dues à la malédiction de la dimension (la signature génomique utilisée dans son étude a 256 dimensions). En utilisant différents types de distances Lespinat a pour but la meilleure représentation des espèces dans l'espace des caractéristiques. Il a bien constaté que la distance euclidienne ne permet pas une bonne séparation des classes (fig.4.16b), les deux classes se chevauchent. La meilleure séparation des deux classes est faite à l'aide des distances curvilignes ce qui aussi confirmé par la performance du KNN.

Pour la base de dissimilitudes euclidiennes le *Cs* géométrique avec $\gamma_1 = \gamma_2 = \delta_1 = \delta_2 = 1$ a un taux d'erreur très élevé avec plus de 1000 objets mal classés alors le nombre des paramètres candidats dépasse les 300 000. L'espace de recherche des paramètres est très grand mais on retrouve un finalement un taux d'erreur qui est proche de celui de KNN.

Le nombre d'itérations nécessaires pour l'optimisation des *Cs*-SVM dépasse les 65 000 et le nombre des vecteurs de support est aussi considérable (autour de 500). Dans les deux cas de distances avec correction de l'équilibre la performance du modèle *Cs*-SVM dépasse celle des 1NN. On constate que le taux de rappel est très bas pour les tests avec le modèle *Cs*-SVM.

Le choix de la mesure de dissimilitude influence les résultats du classement ce qui veut dire que le choix d'une dissimilitude appropriée à la base correspond au choix d'une caractéristique discriminante.

Base : CALTECH

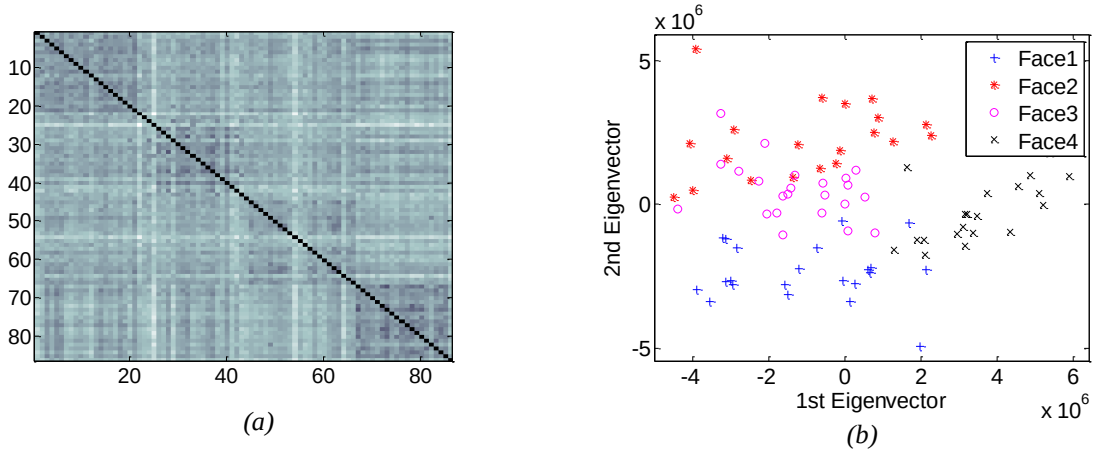
La base de données d'origine contient 450 images de visages en format JPEG avec 27 visages d'hommes et femmes avec un éclairage ou expressions ou milieux différents. De la base d'origine, on a choisi de présenter dans ce chapitre un extrait de cette base de 4 visages, choisis car ils sont présents dans la base avec le plus grand nombre d'images par visage. L'algorithme de description de chaque image est présenté dans le chapitre 3. Afin d'obtenir la distance entre les vecteurs caractéristiques, on a utilisé la distance de Minkowski fractionnaire avec p = 0.5 car il y a peu d'observations avec un grand nombre de caractéristiques. On a testé différentes valeurs de p ($p \in [0.1, \ldots, 0.9]$) et on a choisi celle qui donne les meilleurs résultats de classement avec le classifieur KNN en LOO.

Référence :

- http://www.vision.caltech.edu/Image_Datasets/faces

Fig. 4.17 Les quatre visages de la base CALTECH, utilisée pour ce problème de classement;

86	Nombre d'objets
81, 5	Valeurs propres positives, négatives
0.0017	NEF
0.0079	NER
4	Classes avec taille [22 21 23 20]



(a) *(b)*

Fig. 4.18 (a) Matrice d'intensité des dissimilitudes CALTECH; (b) Projection des dissimilitudes dans l'espace, déterminé par les deux premiers vecteurs propres, les différents types de points indiquent les 19 classes respectivement.

Tableau 4.18 : Erreur de classement en %, procédure LOO, pour la base totale

1NN/KNN [3]	10.47/4.65 [9]
Cs géometrique [γ_2, δ_2], [γ_3, δ_3], [γ_4, δ_4]	19.77 [1.347 0.9941], [1.1215 0.9901], [1.05 1.01]
Cs-SVM	0
Cs-LSSVM	0

Pour cette base la performance du classifieur KNN est très bonne et surpasse celle de 1NN et *Cs* géométrique. Même si les coefficients NER et NEF sont très proche de 0 ce qui veut dire un comportement de la distance proche de l'euclidien, le modèle *Cs* géométrique ne peut pas bien séparer le visage 3 et visage 4 du visage 1. Pour le visage 3 on a 7 images mal classées dans la classe du visage 1 et pour le visage 4 on a 6 images mal classées dans la même classe. Ce mélange des trois classes est illustré à la figure 4.17a.

Pour le modèle *Cs*-SVM, on constate que chaque visage est 100% séparé des autres trois visages. Le processus d'optimisation exige juste 2 ou 3 itérations afin de trouver un petit nombre de vecteurs de supports (3 ou 4 vecteurs suivant la base). Le même fait est observé pour le modèle *Cs*-LSSVM où aussi tous les visages sont classés correctement.

4.3. Conclusion

Pour des problèmes de classification supervisée, il est important de disposer de différentes techniques de classement adaptées aux structures de données de natures très diverses, comme dans le cas des données de dissimilitudes.

Les expérimentations numériques présentées dans ce chapitre comparent la performance des trois modèles du « Coefficient de forme » avec celle des classifieurs, basés sur de matrices de dissimilitudes de Pekalska et Haasdonk et le classifieur 1NN/KNN. Ces expériences ont montré des performances comparables aux classifieurs les plus performants de la littérature. Les modèles *Cs* géométrique *Cs*-SVM et *Cs*-LSSVM sont bien adaptés aux différentes mesures de dissimilitudes et ils sont plus réussis dans plus part des cas en comparaison avec le classifieur 1NN/KNN. L'exemple de comparaison avec 1NN est illustré à la fig. 4.19a.b.c où les axes de chaque nuage de points représentent les erreurs classement pour le classifieur 1NN et respectivement *Cs* géométrique *Cs*-SVM et *Cs*-LSSVM. Cette figure résume la comparaison entre ces différents classifieurs.

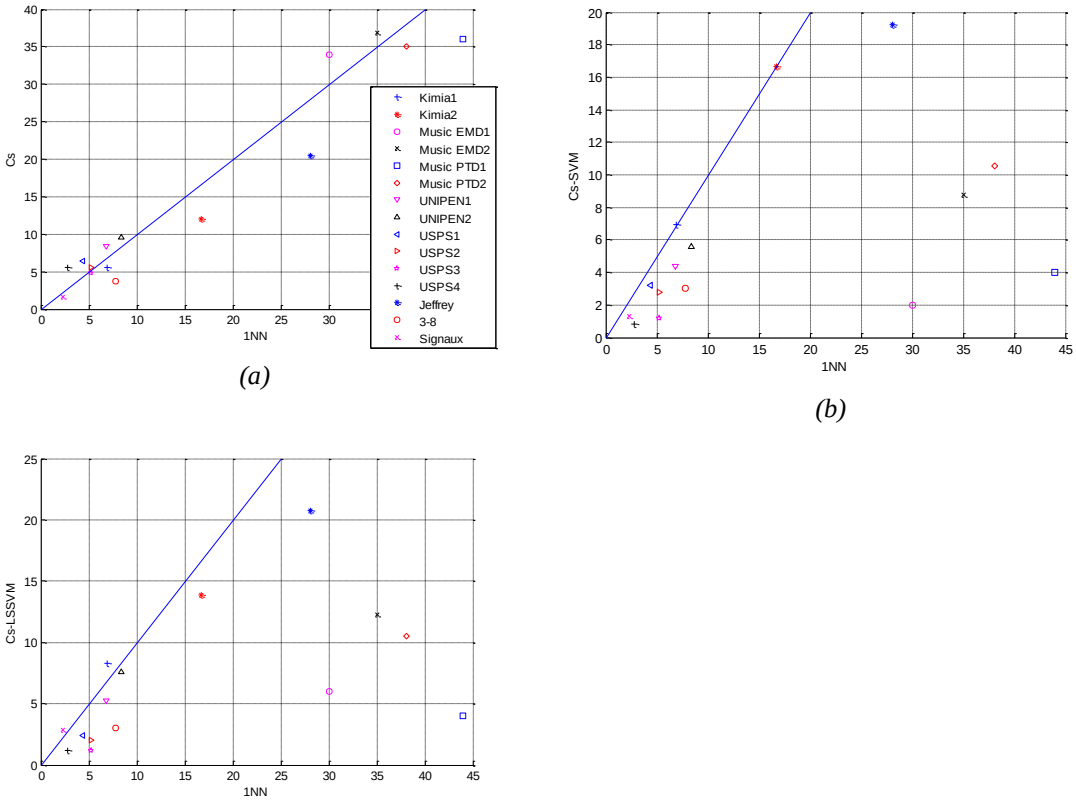


Fig. 4.19 Nuage des points représentants les taux d'erreur en LOO tirés des tableaux des différentes bases données de 1NN par rapport (a) Cs géométrique, (b)Cs-SVM et (c) Cs-LSSVM.

L'exemple de comparaison avec les SVM avec le noyau gaussien est illustré à la fig. 4.20a.b.c où les axes de chaque nuage de points représentent les erreurs classement pour ce classifieur et respectivement *Cs* géométrique *Cs*-SVM et *Cs*-LSSVM.

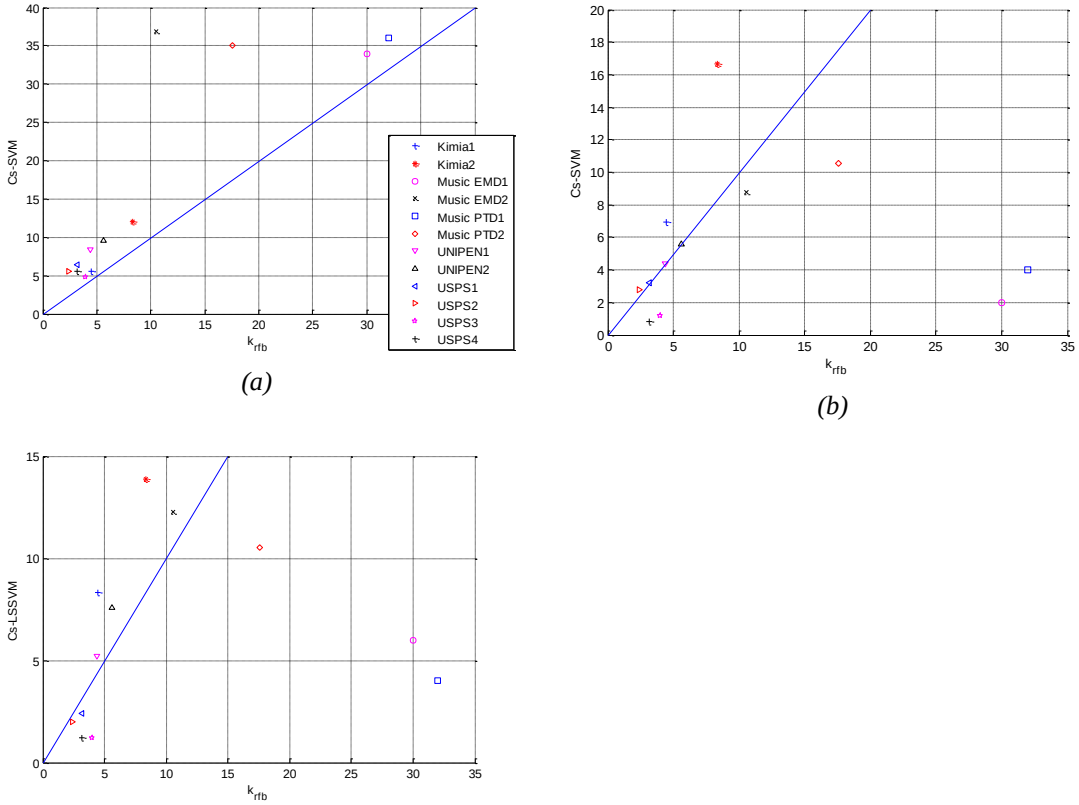


Fig. 4.20 Nuage des points représentants les taux d'erreur en LOO tirés des tableaux des différentes bases données de SVM k^{rfb} par rapport (a) Cs géométrique, (b)Cs-SVM et (c) Cs-LSSVM.

Enfin après les expériences menées sur des bases de données issues des applications du monde réel, on peut conclure que les caractéristiques des trois modèles proposés sont :

- leur simplicité (calcul de moyennes et de variances),
- leur implémentation efficace (implémentation récursive pour la validation croisée),
- leur paramétrage parcimonieux (2 paramètres d'apprentissage par classe),
- leur bonne adaptation aux différentes mesures de dissimilitudes,
- et leur robustesse face aux dissimilitudes manquantes (résultats dans le chapitre 2).

Le résultat de la comparaison avec les SVM montre une efficacité comparable et pour certaines bases, les modèles du « Coefficient de forme » sont parfois meilleurs que les autres classifieurs proposés. En résumé pour simplifier, nous avons d'une part des modèles comme ceux proposés par Haasdonk et Pekalska, plus compliqués et nécessitant des prétraitements et un choix de noyaux adéquats. Les performances obtenues sont très liées à ces choix. D'autre part, les modèles proposés dans cette thèse sont plus simples et nécessitent faible paramétrage. Leur performance est très robuste suivant les différents types de dissimilitudes.

Les résultats sur des ensembles réels montrent que le « Coefficient de forme » est un bon compromis vis-à-vis les classifieurs de Pekslaska et Haasdonk dans sa simplicité et dans sa rapidité. Il peut être utilisé avec différentes mesures de dissimilitudes.

Conclusion et Perspectives

Avec le développement des domaines tels que la Reconnaissance des formes et le « Data Mining », il est important de disposer de différentes techniques de classement adaptées aux structures de données de différente nature et origine. L'utilisation de la notion de proximité directement sur les observations, sans étape préalable d'extraction de caractéristiques est un concept assez récent dans le domaine de l'apprentissage statistique. Les mesures de proximité permettent de capturer les informations statistiques et structurelles des formes et par conséquent elles forment un pont naturel entre ces deux approches [Duin et al. 2004]. E. Pekalska et ses collaborateurs ont proposé une discussion très riche sur les représentations par dissimilitudes et les classifieurs appropriés.

Dans cette thèse, inspirée par les recherches de Pekalska, Duin et Haasdonk, on a défini un indice de proximité, nommé « Coefficient de forme » à partir de la considération sur la représentation de données gaussiennes dans un espace euclidien. Ce cadre d'étude a permis de concevoir un ensemble de règles de décision cohérentes - de la plus simple (linéaire) à la plus complète (quadratique). Avec seulement un ou deux paramètres par classe (selon le modèle de classement retenu), le modèle de description de classe proposé est compact et parcimonieux. Alors que le point de départ de la réflexion pouvait sembler restrictif de part les hypothèses de représentation euclidienne et de distribution gaussienne, les expériences sur des différents types de bases de données ont démontré une grande flexibilité et l'efficacité de l'indice proposé. Ces expériences ont montré des performances du « Coefficient de forme » comparables aux classifieurs les plus performants de la littérature. Les caractéristiques de ces règles de décision du « Coefficient de forme » sont les suivantes :

- parcimonie (uniquement fondées à partir des statistiques d'ordre un et deux des valeurs de dissimilitude,
- mise en œuvre facile (mise en œuvre récursive pour la validation croisée),
- flexibilité - deux paramètres d'ajustement pour apprendre la forme et les dimensions intrinsèques de chaque classe et la propriété la plus intéressante,
- comportement stable face à des matrices de dissimilitudes incomplètes et des dissimilitudes non métriques.

Nous nous sommes également intéressés à la description des images à partir d'une représentation multi échelle afin de tester cette décomposition pour la reconnaissance d'image par le contenu.

Finalement on peut résumer les résultats de cette thèse par ces différents points.

- A partir d'un raisonnement simple sur des distributions gaussiennes dans un espace euclidien, on a proposé un indice de proximité appelé « Coefficient de forme ». A partir

de là, plusieurs modèles des règles de décision sont proposés suivant le nombre des paramètres d'ajustement (chapitre 2).

- Pour réaliser l'apprentissage et trouver les paramètres optimisant la règle de décision, un recodage non linéaire des données de dissimilitudes a permis d'obtenir une règle de décision linéaire. L'optimisation réalisée utilise celle des classifieurs à vecteurs de support et des classifieurs à vecteurs de support aux moindres carrés. Pour ce faire, les équations pour l'optimisation des SVM et LSSVM ont été adaptées pour la limitation de l'espace des paramètres du plan optimal séparateur (chapitre 2).
- On a étudié expérimentalement l'influence des données manquantes sur le comportement du « Coefficient de forme » géométrique sous ses différentes formes vis-à-vis la règle des K plus proches voisins (chapitre 2) ;
- On a comparé expérimentalement le « Coefficient de forme » avec ses différentes variantes, *Cs* géométrique, *Cs*-SVM et *Cs*-LSSVM sur des bases de données artificielles des distributions bidimensionnelles gaussiennes et sur des bases de données issues des applications du monde réel (chapitre 2 et chapitre 4);
- Nous avons développé un algorithme de représentation d'images en se basant sur la méthodologie de « Pyramide Réduite Différentielle » utilisant la transformation de Mellin-Fourier ou la transformation en Cosinus Discrète (chapitre 3). Plusieurs stratégies de masquage des coefficients après la transformation ont été testées. Le choix parmi ces stratégies dépend du nombre de niveaux que l'on veut garder dans la pyramide. La méthode a été testée dans un système réduit de recherche d'image par contenu.

Perspectives

L'indice « Coefficient de forme » peut toujours être amélioré avec des propositions de nouvelles stratégies d'optimisation des paramètres. Une voie de développement futur sera d'étudier avec détails la proposition de Haasdonk et Pekalska du noyau de distance de Mahalanobis [Haasdonk&Pekalska 2008] et de comparer le comportement des deux classifieurs et de s'inspirer de ces nouvelles idées pour le développement du « Coefficient de forme » notamment avec les fonctions de « noyaux ».

L'algorithme de description des images en multi échelle offre beaucoup de possibilités d'optimisation et de travail. L'étape suivante est inspirée par [Derrode et al. 1999] et [Derrode&Ghorbel 2001] où on propose d'utiliser juste la combinaison des transformations de Fourier et log-polaire afin d'extraire les caractéristiques des images sans la deuxième transformation de Fourier.

Les bons résultats obtenus en identification des visages nous encouragent à poursuivre l'étude sur plus grandes bases de visages et naturellement d'essayer de se positionner parmi les autres études faites sur ce sujet.

Publications

[1]. R. Kountchev, A. Manolova. Fast algorithm for color space K-L image transform. Intern. Scientific Conf. on Information, Communication and Energy Systems and Technologies (ICEST'05), Nis, Serbia and Montenegro, Proc. Vol. 1, June 2005, pp. 322-325.

[2]. R. Kountchev, A. Manolova. Determination of Distances between Images Using Consecutive Approximations in the Linear Transforms Domain. Proc. of National Conf. with Foreign Participation (TELECOM'05), Varna, Bulgaria, October 6-7, 2005.

[3]. R. Kountchev, A. Manolova. General Approach for Fast Image Retrieval from Image Database using Consecutive Iterations in the Transform Domain. Proceedings of the Technical University - Sofia, vol. 55, 2005, pp. 167-176.

[4]. Manolova, A.Guerin-Dugue. Une nouvelle metrique pour l'analyse discriminante sur donnees de dissimilitude. 39ièmes Journées de Statistique, 11-15.06. 2007 Angers, France.

[5]. A. Manolova, G. Celeux, A.Guerin-Dugue. Classification of dissimilarity data with a new flexible Mahalanobis like metric, PAA Special Issue on Non-parametric Distance-based Classification Techniques and their Applications**,** Springer-link **,** 11(3-4): 337-351 (2008)

[6]. A. Manolova, R. Kountchev, I. Aleksieva, Improved system for content based image retrieval based on pyramid decomposition in the spectrum domain, Intern. Scientific Conf. on Information, Communication and Energy Systems and Technologies (ICEST'09), Veliko Tarnovo, Bulgaria, 25-27.06.2009

[7]. A. Manolova, A. Guerin-Dugue, Dissimilarity based metric for data classification based on Support Vector Classifiers, XVI Rencontres de la Societe Francophone de Classification, 2-4 September 2009, Grenoble, France.

[8]. A. Manolova, R. Kountchev, Face Identification with Modified K-NN classifier based on linear and angular distance, Intern. Scientific Conf. on Information, Communication and Energy Systems and Technologies (ICEST'10), Ohrid, Macedonia, 23-26.06.2010

Rapport de fin d'études

[9]. A. Manolova, *Analyse discriminante sur tableaux de dissimilitudes : Application à la catégorisation d'images*, rapport de DEA, 2004, CLIPS IMAG.

Références Bibliographiques:

[Abe 2005] S. Abe. *Support vector machines for pattern classification*, Springer-Verlag, 2005.

[Adelson 1991] E. H. Adelson, Layered Representations for Image Coding, MIT Media Laboratory Vision and Modeling Technical Report, 1991.

[Adelson et al. 1991] E. H. Adelson, E. P. Simoncelli, W. T. Freeman. *Pyramids and Multiscale Representations, Representations of Vision*, Cambridge University Press, pp. 3-16, 1991.

[Aleksieva 2008] Aleksieva. Multiresolution image analysis: Content Based Image retrieval, technical report, GIPSA-Lab, 2008.

[Alpaydin 2004] E. Alpaydin. *Introduction to machine learning*, MIT Press, Cambridge, 2004.

[Bensmail&Celeux 1996] H. Bensmail, G. Celeux. "Regularized Gaussian discriminant analysis through eigenvalue decomposition". Journal of the American Statistical Association, 91, 1743-48. (1996).

[Bensmail et al. 2006] C. Biernacki, G. Celeux, G. Govaert, G. and F. Langrognet. "Le logiciel MIXMOD d'analyse de mélange pour la classification et l'analyse discriminante" La Revue de Modulad, 35 , 25-44, (2006).

[Del Bimbo 1999] A. Del Bimbo. *Visual Information Retrieval*, Academic Press, San Francisco, CA, 1999.

[Bishop 2006] C. Bishop. *Pattern recognition and machine learning*, Springer-Verlag, 2006.

[Borg& Groenen 1997] I. Borg, P. Groenen. *Modern multidimensional scaling: theory and applications*. Springer, New York, 1997.

[Boser et al. 1992] B. E. Boser, I. Guyon, and V. Vapnik. A Training Algorithm for Optimal Margin Classifiers, Proceedings Fifth ACM Workshop on Computational Learning Theory, pp. 144–152, July 1992.

[Bunke et al. 2001] H. Bunke, S. Gunter, and X. Jiang. Towards bridging the gap between statistical and structural pattern recognition: Two new concepts in graph matching. In International Conference on Advances in Pattern Recognition: Springer LNCS2013, pages 1.11, 2001.

[Burges 1998] C.J.C. Burges. A tutorial on Support vector machines for pattern recognition, Data mining and knowledge discovery, 2, 121-167, 1998.

[Calciu&Benavent] M. Calciu, C. Benavent. L'analyse discriminante, note pédagogique, Université des Sciences et Technologies de Lille.

[Catwkell 2000] T. Cawkell. Image indexing and retrieval by content. Information Services and Use 20, no. 1 (2000): 49-58.

[Celeux&Govaert 1995] G. Celeux, G. Govaert. Parsimonious Gaussian models in cluster analysis. Pattern Recognition 28:781–793, 1995.

[Celeux 1990] G. Celeux. *Analyse Discriminante sur variables continues*. Collection Didactique **7**, INRIA (1990).

[Celeux&Turlot 1989] G.Celeux, J.C. Turlot "Estimation de la qualité d'une règle discriminante". La Revue de Modulad 4, 37-46, (1989).

[Cox&Cox 2001] T. F. Cox, M. A.A. Cox. *Multidimensional scaling*, CRC Press, 2001.

[Cristianini&Shawe-Taylor 2000] N. Cristianini, J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[Van Cuntsen 1994] B. Van Cutsen (edt). Classification and dissimilarity analysis. In: Lecture notes in statistics, vol 95. Springer, Heidelberg, 1994.

[Derrode et al. 1999] S. Derrode, M. Daoudi, F. Ghorbel. Invariant Content-Based Image Retrieval Using a Complete Set of Fourier-Mellin Descriptors, Proceedings of the 1999 IEEE International Conference on Multimedia Computing and Systems - Volume 02.

[Derrode&Ghorbel 2004] S. Derrode, F. Ghorbel: Shape analysis and symmetry detection in gray-level objects using the analytical Fourier-Mellin representation. Signal Processing 84(1): 25-39 (2004)

[Derrod&Ghorbel 2001] S. Derrode, F. Ghorbel. Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description, Computer Vision and Image Understanding, Vol. 83(1), pp. 57-78, 2001.

[Devroye et al. 1996] L. Devroye, L. Györfi, G. Lugosi. *A probabilistic theory of pattern recognition*, Springer, 1996.

[Deza&Deza 2009] M. M. Deza, E. Deza. *Encyclopedia of Distances*, Springer-Verlag, 2009.

[Duan& Keerthi 2005] Kai-Bo Duan, S. Keerthi. Which Is the Best Multiclass SVM Method? An Empirical Study. Proceedings of the Sixth International Workshop on Multiple Classifier Systems, (2005).

[Dubuisson&Jain 1994] M.P. Dubuisson and A.K. Jain. Modified hausdorff distance for object matching. In 12th International Conference on Pattern Recognition, volume 1, pages 566–568, 1994.

[Duch 2000] W. Duch. Similarity-based methods: a general framework for classification, approximation and association. Control Cybern 29(4):937–968, 2000.

[Duda et al. 2000] R. O. Duda, Peter E. Hart, David G. Stork. *Pattern classification*, Wiley-Interscience; 2 edition (October 2000).

[Duff et al. 1986] I.S. Duff, A.M. Erisman, J.K. Reid. *Direct methods for sparce matrices*, Clarendon press, 1986.

[Duin&Pekalska 2009] R.P.W. Duin and E. Pękalska, Datasets and tools for dissimilarity analysis in pattern recognition, Technical Report 2009_9, SIMBAD (EU,FP7,FET), 2009, 1-174.

[Duin et al. 2008] R.P.W. Duin, E.Pękalska, A.Harol, W.-J.Lee and H.Bunke, On Euclidean corrections for non-Euclidean dissimilarities, Joint IAPR International Workshops on Statistical and Structural Pattern Recognition, 551-561, 2008.

[Duin et al. 2004] R.P.W. Duin, E. Pękalska, P. Paclik and D.M.J. Tax. The dissimilarity representation, a basis for domain based pattern recognition?, invited talk (refereed), Representations in Pattern Recognition, IAPR Workshop, Cambridge, 43-56, 2004.

[Duin et al. 1998] R.P.W. Duin, D. de Ridder, and D.M.J. Tax. Featureless pattern classification. Kybernetika, 34(4):399.404, 1998.

[Edelman et al. 1998] S. Edelman, S. Cutzu, and S. Duvdevani-Bar. Representation is representation of similarities. Behavioral and Brain Sciences, 21:449.498, 1998.

[Elgemmal et al. 2009] A. Elgemmal, C. Muang, D. Hu. *Skin detection – a short tutorial*, in S.Z. Li and A. Jain [eds], Encyclopedia of Biometrics, Springer Verlag, 2009.

[Ferrari et al. 1995] F. Ferrari, J. Nielsen, P. Questa and G. Sandini. Space variant imaging, Sensor Review Vol. 15, issue N. 2, pages 17-20, 1995.

[Fuknaga 2001] K. Fukunaga, *Introduction to statistical pattern recognition*, second edition, Academic Press, 2001.

[Gevers 2001] T. Gevers, Principles of Visual Information Retrieval, M. Lew, Ed., ch. Color in image search engines. Springer-Verlag, Heidelberg, February 2001, 11–48.

[Ghorbel et al. 2006] F. Ghorbel, S. Derrode, R. Mezhoud, M. Tarak Bannour, S. Dhahbi: Image reconstruction from a complete set of similarity invariants extracted from complex moments. Pattern Recognition Letters 27(12): 1361-1369 (2006)

[Giannopoulos&Veltkamp 2002] P. Giannopoulos , R. C. Veltkamp, A Pseudo-Metric for Weighted Point Sets. In Heyden, A., Sparr, G., Nielsen, M.& Johansen, P. (Ed.), Proceedings of the 7th European Conference on Computer Vision (ECCV) (pp. 715–730). Copenhagen, Denmark: Springer-Verlag.

[Goldfarb et al. 2004] L. Goldfarb, O. Golubitsky, D. Korkin. What is a structural

representation?, Faculty of Computer Science, U.N.B., Technical Report TR00-137, December 2004.

[Goldberger et al. 2003] J. Goldberger, S. Gordon, H. Greenspan. An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures, Proceedings Ninth IEEE International Conference on Computer Vision, pp. 487 – 493, vol.1, 2003.

[Gosling 1995] J. Gosling. *Introductory statistics*, Glebe, N.S.W. : Pascal Press, 1995.

[Gower&Legendre 1986] J.C. Gower, Legendre. Properties of Euclidean and non-Euclidean distance matrices, Linear algebra and its applications, 67:81-97, Elsevier Science Publishing Inc. 1986.

[Grauman&Darell 2004] K. Grauman, T. Darrell. Fast Contour Matching Using Approximate Earth Mover's Distance, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington DC, June 2004.

[Guerin&Celeux 2001] A. Guérin-Dugué, G. Celeux. Discriminant analysis on dissimilarity data: a new fast Gaussian-like algorithm. In: AISTATS'20001, Fl, USA, pp 202–207, 2001.

[Gupta&Jain 1997] A. Gupta, R. Jain. *Visual information retrieval*, 1997.

[Haasdonk 2005a] B. Haasdonk, Transformation Knowledge in Pattern Analysis with Kernel Methods. PhD thesis, Computer Science Department, University of Freiburg, May 2005, Shaker-Verlag.

[Haasdonk 2005b] B. Haasdonk. Feature Space Interpretation of SVMs with Indefinite Kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(4):482-492, april 2005.

[Haasdonk& Bahlmann 2004] B. Haasdonk, C. Bahlmann. Learning with Distance Substitution Kernels. Pattern Recognition - Proc. of the 26th DAGM Symposium, Tübingen, Germany, August/September 2004, pp. 220-227. Springer Berlin, 2004.

[Haasdonk& Burkhardt 2007] B. Haasdonk, H. Burkhardt. Classification with Invariant Distance Substitution Kernels. Proc. of 31st. GfKl Conference, Data Analysis, Machine Learning, and Applications, University of Freiburg, 2007.

[Haasdonk& Burkhardt 2005] B. Haasdonk, H. Burkhardt. Invariant Kernels for Pattern Analysis and Machine Learning. Internal Report 3/05, IIF-LMB, Computer Science Department, University of Freiburg, 2005.

[Haasdonk&Keysers 2002] B. Haasdonk, D. Keysers. Tangent Distance Kernels for Support Vector Machines. In Proc. of the 16th Int. Conf. on Pattern Recognition, vol. 2, pp. 864-868, IEEE, 2002.

[Haasdonk&Pekalska 2008] B. Haasdonk, E.Pękalska. Classification with Kernel Mahalanobis Distance Classifiers, German Classification Society Annual Conference, 2008.

[Haasdonk&Pekalska 2009] B. Haasdonk, E.Pękalska. Indefinite Kernel Fisher Discriminant, oral presentation, International Conference on Pattern Recognition 2009.

[Haasdonk&Pekalska 2010] B. Haasdonk and E.Pękalska, Idefinite Kernel Discriminant, invited paper, International Conference on Computational Statistics, 2010.

[Hammer&Vilmann 2005] B. Hammer, T. Vilmann. Classification using non-standard metrics. In: Esann'2005, 27–29 April, Bruges, Belgium, pp 303–316, 2005.

[Han&Kamber 2006] J. Han, M. Kamber. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, March 2006.

[Hastie et al. 2009] T. Hastie, R. Tibshirani, J. Friedman. *The elements of statistical learning: Data mining, inference and prediction*, Springer-Verlag, 2009.

[Herbrich 2002] R. Herbrich. *Learning kernel classifiers: Theory and algorithms*, MIT Press, 2002.

[Huttenlocher et al. 1993] D. Huttenlocher, G. Klanderman and W. Rucklidge. Comparing Images Using the Hausdorff Distance, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, no. 9, pp. 850-863, 1993.

[Jacobs et al. 2000] D.W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with Non-Metric Distances: Image Retrieval and Class Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(6):583.600,

2000.

[Jiao et al. 2007] Licheng Jiao, Liefeng Bo, Ling Wang, Fast Sparse Approximation for Least Squares Support Vector Machine, IEEE transactions on neural networks, vol. 18, No. 3, May 2007

[Joachims 1999] T. Joachims. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press, 1999.

[Keshavarzi et al. 2009] M. Keshavarzi, M. A. Deghan, M. Mashinchi. Classification based on similarity and dissimilarity through equivalence classes, Appl. and Comput. Math., V.8, N.2, 2009, pp.203-215.

[Koening 2000] A. M. Koening. Analyse discriminante sur tableaux de dissimilarités. Université de Caen, INRIA, 2000-2001.

[Kountchev et al. 2002] R. Kountchev, V. Haese-Coat, J. Rosnin. Inverse Pyramidal Decomposition with multiple DCT, Signal processing: Image communication 17 (2002) 201-218 Elsevier 2002.

[Kountchev et al. 2005] R. Kountchev, M. G. Milanova, C. Ford, R. Kountcheva. Multi-layer Image Transmission with Inverse Pyramidal Decomposition. Computational Intelligence for Modelling and Prediction 2005: 179-196.

[Kountchev et al. 2010] R. Kountchev, V. Todorov and R. Kountcheva, Invariant object representation with modified Mellin-Fourier transform, 14th WSEAS International Conference on computers, pp. 232-236 Volume I, 2010.

[Kruskal& Wish 1978] J. B. Kruskal, M. Wish. *Multidimensional scaling*, Sage Publications, Inc. (January 1, 1978).

[Larose 2006] D.T. Larose. *Data mining: methods and models*, John Wiley and Sons, 2006.

[Mackay&Zinnes 1986] D. MacKay, Zinnes. *A Probabilistic Model for the Multidimensional Scaling of Proximity and Preference Data*, *Marketing Science*, 1986.

[Michie et al. 1994] D. Michie, D.J.Spiegelhalter, C.C. Taylor (editors). *Machine learning, Neural and statistical classification*, Ellis Horwood, 1994.

[Milanese&Cherbuliez 1999] R. Milanese, M. Cherbuliez. A Rotation, Translation, and Scale-Invariant Approach to Content-Based Image Retrieval, Journal of Visual Communication and Image Representation, Volume 10, Issue 2, June 1999, Pages 186-196.

[Moreno et al. 2003] P. Moreno, P. Ho, N. Vasconcelos. A Kullback–Leibler divergence based kernel for SVM classification in multimedia applications. In: Neural information processing system. Whistler, Canada, 2003.

[Muller 2007] M. Müller. *Information retrieval for music and motion*, Springer; 1 edition (November 14, 2007).

[Niels 2004] R. Niels. Dynamic Time Warping: An Intuitive Way of Handwriting Recognition?, Master's thesis, Radboud University Nijmegen, 2004.

[Nixon&Aguado 2002] M. S. Nixon, A. S. Aguado. *Feature extraction and image processing*, Newnes, Butterworth-Heinemann, 2002.

[Osuna et al. 1997] E. Osuna, R. Freund, F. Girosi. Training Support Vector Machines: Training and Applications. CVPR 1997: 130-136.

[Pekalska 2005] E.Pękalska. The Dissimilarity representations in pattern recognition. Concepts, theory and applications, ASCI Dissertation Series no. 109. Delft University of Technology, Delft, January 2005.

[Pekalska&Haasdonk 2009] E.Pękalska, B.Haasdonk. Kernel Discriminant Analysis for Positive Definite and Indefinite Kernels, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.31, no.6, 1017-1032, 2009.

[Pekalska&Duin 2008] E.Pękalska, R.P.W. Duin. Beyond traditional kernels: classification in two dissimilarity-based representation spaces, IEEE Transactions on Systems, Man and Cybernetics--Part C, vol. 38, no. 6, 729-744, 2008.

[Pekalska&Duin 2005] E. Pękalska, Robert P.W. Duin. *The dissimilarity representation for pattern recognition. Foundations and Applications*, World Scientific, Singapore, December 2005.

[Pekalska et al. 2006] E.Pękalska, R.P.W. Duin, P. Paclik. Prototype Selection for Dissimilarity-based Classifiers, Pattern Recognition, vol. 39, issue 2, 189-208, 2006.

[Pekalska&Duin 2002] E. Pękalska, R.P.W. Duin. Dissimilarity representations allow for building good classifiers, Pattern Recognition Letters, vol. 23, no. 8, 943-956, 2002.

[Pekalska et al. 2002] E. Pękalska, P. Paclik, and R.P.W. Duin. A Generalized Kernel Approach to Dissimilarity-based Classification, Journal of Machine Learning Research, Special Issue on Kernel Methods, vol. 2, no. 2, 175-211, 2002.

[Pekalska&Duin 2001] E. Pękalska, R.P.W. Duin. Automatic pattern recognition by similarity representations, Electronics Letters, vol. 37, no. 3, 159-160, 2001.

[Pekalska&Duin 2006] E.Pękalska, R.P.W. Duin. Dissimilarity-based classification with vectorial representations, oral presentation, International Conference on Pattern Recognition, vol. 3, 137-140, Hong Kong, 2006.

[Pekalska et al. 2005] E.Pękalska, A.Harol, C.Lai and R.P.W. Duin. Pairwise selection of features and prototypes, oral presentation, International Conference on Computer Recognition Systems, Rydzyna, Polanad, 271-278, 2005.

[Petres 2008] R.A. Peters II. On the Computation of the Discrete Log-Polar Transform, IEEE Trans IP (in review) 2008

[Piro et al. 2008] P. Piro, S. Anthoine, E. Debreuve, Michel Barlaud. Image retrieval via Kullback-Leibler divergence of patches of multiscale coefficients in the k-NN framework. In CBMI, London, UK, June 2008.

[Qian et al. 2004] G. Qian, S. Sural, Y. Gu, S. Pramanik. "Similarity between Euclidean and cosine angle distance for nearest neighbor queries." In Proc. of the 2004 ACM symposium on applied computing, pp. 1232-1237, 2004.

[Rakotomalala 2008] R. Rakotomalala. Tests de normalité: Techniques empiriques et tests statistiques, cours Université de Lyon, 2002.

[Rubner et al. 2000] Y. Rubner, C. Tomasi, L.J. Guibas. The earth mover's distance as a metric for image retrieval, International Journal of computer vision 40, 99-121, 2000.

[Kim&Oommen 2007] Sang-Woon Kim, B.J. Oommen, On using prototype reduction schemes to optimize dissimilarity-based classification, Elsevier Ltd. Pattern Recognition Journal vol. 40, 2946-2957, 2007.

[Rifkin&Klautau 2004] R. Rifkin, A. Klautau. In Defense of One-Vs-All Classification. Journal of Machine Learning Research, Volume 5, pp. 101-141, 2004.

[Schalkoff 1997] R. J. Schalkoff. *Arificial neural networks*, McGraw-Hill, 1997.

[Rscettini et al. 2001] R. Schettini, G. Ciocca, and S. Zuffi. A Survey of Methods for Color Image Indexing and Retrieval in Image Databases, ch. Color imaging science: Exploiting digital media, R. Luo and L. Mac Donald, Eds., JohnWiley & Sons, New York, 2001.

[Schlökopf 2000] B. Schlökopf. The kernel trick for distances. In: Neural information processing system, Vancouver, Canada, pp 301–307, 2000.

[Schlökopf et al. 2000] B. Schlökopf, A.J. Smola, R. Williamson, P.L. Bartlett. New support vector algorithms. Neural Computation, 12:1207–1245, 2000.

[Schwartz 1994] El. Schwartz. Computational studies of the spatial architecture of primate visual cortex: Columns, maps, and protomaps. In: Peters A, Rockland KS, editors. Primary Visual Cortex in Primates. Cerebral Cortex. Vol. 10. New York: Kluwer Academic/Plenum Publishers; 1994. pp. 359–411.

[Simard et al. 1996] P. Simard, Y. LeCun, J. Denker, B. Victorri, Transformation invariance in pattern recognition – tangent distance and tangent propagation, Neural Networks: Tricks of the Trade 1996: 239-27.

[Smeulders et al. 2000] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content based image retrieval at the end of the early years, IEEE Trans. on Patt. Anal. and Machine Intelligence, 22, 1349–1380, 12.2000.

[Suykens et al. 1999] J.A.K Suykens, L.Lukas, P.Van Dooren, B.De Moor, J.Vandewalle. Least squares support vector machine classifiers: a large scale algorithm, in Proc. of the European Conference on Circuit Theory and Design (ECCTD'99*)*, Stresa, Italy, Sep. 1999, pp. 839-842.

[Suykens&Vandewalle 2000] Suykens J.A.K., Vandewalle J., Recurrent least squares support vector machines, IEEE Transactions on Circuits and Systems-I, vol. 47, no. 7, Jul. 2000, pp. 1109-1114.

[Suykens et al. 2001] Suykens J.A.K., Vandewalle J., De Moor B., Optimal control by least squares support vector machines, Neural Networks, vol. 14, no. 1, Jan. 2001, pp. 23-35.

[Suykens et al. 2002] J. A.K. Suykens, T. Gestel, J. de Brabanter, B. de Moor, J. Vandewalle. *Least Squares Support Vector Machines*, World Scientific Publishing Co., 2002.

[Suykens et al. 2003] Suykens J.A.K., Horvath G., Basu S., Micchelli C., Vandewalle J., (eds.), *Advances in Learning Theory : Methods, Models and Applications*, vol. 190 of NATO-ASI Series III : Computer and Systems Sciences, IOS Press (Amsterdam, The Netherlands), (ISBN 1-58603-341-7), 2003.

[Taguchi et al. 2002] G. Taguchi, Jugulum Rajesh, Rajesh Jugulum. *The Mahalanobis Taguchi Strategy*, John Wiley & Sons Inc. 2002.

[Tewarson 1973] R. P. Tewarson. *Sparce matrices*, Academic Press, 1973.

[Theodoridis&Koutroumbas 2003] S. Theodoridis, K. Koutroumbas. *Pattern Recognition*, Elsevier, 2003.

[Typke et al. 2003] R. Typke, P. Giannopoulos, R.C. Veltkamp, F. Wiering and R. van Oostrum. Using transportation distances for measuring melodic similarity. In Proc. ISMIR 2003, pp. 107--114, 2003

[Typke 2011] R. Typke. Music Retrieval based on Melodic Similarity, 2011, Ph.D thesis.

[Van Gestel et al. 2004] T. Van Gestel, J. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and J. Vandewalle, Benchmarking least squares support vector machine classifiers, Machine Learning, vol. 54, no. 1, pp. 5–32, 2004.

[Vapnik 1995] V. Vapnik. *The Nature of Statisticsal Learning*. Springer, N.Y., 1995.

[Vezhnevets et al. 2003] V. Vezhnevets, V. Sazonov, A.Andreeva. A survey on pixel-based skin color detection techniques, Proc. Graphicon-2003.

[Wang et al. 2005] L. Wang, Yan Zhang, J. Feng, On Euclidean distance for images, Pattern Analysis and Machine Intelligence, IEEE, Aug. 2005, Volume:27 Issue: 8 , pages: 1334 - 1339

[Wang et al. 2006] Qi Wang, Yan-jun Li, Ke Zhang, Xian-ze Xiong. Inverse log-polar transformation algorithm based on sub-pixel interpolation, Optoelectronics Letters, Volume 2, Number 3 / May, 2006.

[Webb 2002] A. R. Webb, *Statistical pattern recognition*, 2002 John Wiley & Sons, Ltd.

[Weis&Provost 2001] G. Weiss, F. Provost, The effect of class distribution on classifier learning : on empirical study, Technical report, Department of computer science Rutgers University, 2001.

[Zana&Cesar 2006] Y. Zana, R M. Cesar. Face recognition based on polar frequency features. ACM Transactions on Applied Perception, 2006, 3(1): 62-82.

[Zinke&Mayer 2006] A. Zinke. D. Mayer, Iterative Multi Scale Dynamic Time Warping, Universität Bonn, Technical Report number CG-2006-1, Nov. 2006.

[Zoutendijk 1970] G. Zoutendijk. Nonlinear Programming Computational Methods. In: Integer and Nonlinear Programming, Abadie, J. (Ed.). North-Holland, Amsterdam, pp: 37-86, 1970.