Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITÉ MOULOUD MAMMERI - TIZI-OUZOU
Faculté des Sciences
Département de Mathématiques

T H È S E

pour obtenir le titre de

Doctorat

Spécialité : Mathématiques
Option : Recherche Opérationnelle

Présentée par

Ahmed Amirou

Optimisation des SVMs pour la discrimination de signaux

Dirigée par Mohamed Aidene et Djaffar Ould-Abdeslam

soutenue le 02 / 03 / 2015

devant le Jury :

Président :	Djennoune Said	Pr, Université de Tizi-Ouzou
Directeur de Thèse :	Aidene Mohamed	Pr, Université de Tizi-Ouzou
Co-Directeur de Thèse :	Ould-Abdeslam Djaffar	HDR, Université UHA, (France)
Examinateurs :	Benalla Hocine	Pr, Université de Constantine
	Kouadri abdelmalek	MCA, Université de Boumerdes
	Oukacha Brahim	MCA, Université de Tizi-Ouzou

Remerciements

~ Mes premiers remerciements sont adressés à mon directeur de thèse, Aidene Mohamed, professeur à l'Université de Tizi ouzou qui m'a ouvert les portes de la recherche scientifiques, d'abord pour acceder aux magister puis au doctorat. Je lui suis reconnaissant pour les qualités scientifiques et humaines et pour sa disponibilité. Il a su guider cette étude avec sagesse, tout en me laissant une grande liberté. Je le remercie sincèrement.

~ Je remercie mon co-directeur de thèse , Ould Abdeslam Djafar enseignant-chercheur au laboratoire de recherche Mips de l'université de Mulhouse, pour ses qualités scientifiques et humaines. C'est lui qui m'a donné les premières pistes d'initiation à la recherche. Je lui fais part de ma gratitude pour la très grande compétence avec laquelle il a suivi ce travail, mais également pour l'aide qu'il m'a apporté à chaque étape clé et sa disponibilité pendant mon séjour de 18 mois au sein de leurs laboratoire de recherche. Je le remercie sincerment pour sa disponibilité, sa patience et surtout sa gentillesse.

~ Mes remerciements s'adressent aux professuers responsables du laboratoire Mips Jean Phillips et Jean Merckle, pour m'avoir reçu pour une periode de 18 mois, en me fournissant les moyens matériels et humains pour finaliser cette thèse. Je remercie aussi l'équipe, pour leur disponibilité, leur patience et surtout leur gentillesse.

~ Je remercie Djenoune Said professeur au sein de l'Université de Tizi ouzou d'avoir accepté d'être le président du jury de soutenance de cette thèse.

~ Je remercie également Benalla Hocine Professeur à l'Université de Constantine, Kouadri abdelmalek MCA à l'Université de Boumerdes et Brahim Oukacha MCA à l'Université de Tizi ouzou d'avoir accepté d'examiner ma thèse.

~ A ma mère, mes soeurs et mes frères, je leur adresse mes plus chaleureux remerciements. Leur soutien moral tout au long de mes études m'a permis de tenir bon dans les moments difficiles. Qu'ils soient certains de toute ma reconnaissance et de tout mon amour.

~ Enfin, je tiens à remercier du fond du coeur ma petite famille. Vos nombreux sacrifices et votre support ont été pour moi une source de réconfort et d'encouragement.

~ Que tous ceux qui m'ont aidé de près ou de loin dans l'élaboration de ce travail trouvent ici l'expression de ma sincère gratitude.

Table des figures

2.1 *éboulis des valeurs propres* 11
2.2 *Projection de l'espace d'entrée sur 2D.* 11
2.3 *Illustration de l'inégalité (2.15). La courbe croissante, appelée confiance, correspond à la borne supérieure du terme de complexité. Les comportements du terme de complexité et de l'erreur empirique sont clairement opposés. On recherche donc le meilleur compromis entre complexité et erreur empirique.* 22
2.4 *La géométrie d'un hyperplan* 25

3.1 *Principe d'hyperplan séparateur, il en existe plusieurs. Celui qui correspond au minimum d'erreurs est l'hyperplan optimal.* 42
3.2 *L'hyperplan optimal H (en gras) avec la marge maximale M.* 43
3.3 *marge* 44
3.4 Exemple SVM Linéaire à marge dur avec support vecteurs 45
3.5 *hyperplan optimal, marge de séparation et vecteur de support* 46
3.6 *hyperplan sous optimal lié à la présence d'une donnée aberante* . . . 51
3.7 *Représentation du compromis entre la largeur de la marge souple et le coût d'une erreur.* 52
3.8 *a)Aucun hyperplan separateur b)hyperplan separateur dans un espace plus elevé* 55
3.9 Fonction de coût charnière. Les groupes de points sont indiqués par rapport à leur coût. I_0 correspond à un coût nul (points bien classés),I_w correspond à un coût nul aussi mais correspond au point de coupure et I_C correspond à un coût non nul (points du mauvais côté de leur marge). 70

4.1 Cette figure montre un couloir de rejet délimité par les deux lignes $+0.59277$ et -0.59277. La classe positive est représentée par des étoiles bleues alors que la classe négative est représentée par des ronds rouges. La ligne 0 correspond à $f(x) = 0$ ou $P(Y = 1 \mid X = x) = 0.5$. 76
4.2 Courbe de compromis erreur-rejet 80
4.3 Coûts prévus en fonction des probabilités *a poteriori*. Nous avons porté en gras la courbe minimale dans chaque intervalle de probabilités. 83
4.4 classification avec zones de rejet d'ambigüités (zone ambrée). 84
4.5 Coûts prévus en fonction des probabilités *a poteriori*. Nous avons porté en gras la courbe minimale dans chaque intervalle de probabilités pour un coût symétrique 85

4.6 *Fontion de perte pour les exemples positifs ($y_i = +1$), avec $P_- = 0.35$ et $P_+ = 0.6$ (continu : fonction construite, interrompu : fonction logistique)* 86

4.7 *Double hinge loss function for positive examples, with $P_- = 0.4$ and $P_+ = 0.7$ (solid : double hinge, red dashed : likelihood)* 87

4.8 Fonction de perte construite indiquant les differents groupes de points. 92

5.1 Génération de deux classes de données selon deux distributions gaussiennes de moyennes respectives m_1 et m_2 avec chevauchement afin de créer des exemples ambigûs. 97

5.2 Zone de rejet induite par les seuils définis en fonction des coûts d'exemples mal classés et celui des échantillons rejetés. Les exemples positifs sont représentés par des étoiles noires alors que les exemples négatifs sont représentés par des ronds rouges. Les lignes $+0.5$ et -0.5 correspondent respectivement à δ_o et $-\delta_o$ et la ligne 0 correspond à $f(x) = 0$ ou $P(Y = 1 \mid X = x) = 0.5$. 99

5.3 Zone de rejet induite par les seuils définis en fonction des coûts d'exemples mal classés et celui des échantillons rejetés. Les exemples positifs sont représentés par des étoiles noires alors que les exemples négatifs sont représentés par des ronds rouges. Les lignes $+0.32277$ et -0.8473 correspondent respectivement à δ_+ et δ_- et la ligne 0 correspond à $f(x) = 0$ ou $P(Y = 1 \mid X = x) = 0.5$ 100

5.4 Comparaison des taux de rejet en fonction du côut de rejet r obtenus respectivement avec le SVMR [Grandvalet 2009] (courbes noires) et avec la règle de décision [Bartlett 2008, Herbei 2006, Wegkamp 2010] (courbes rouges). Ces résultats sont obtenus en utilisant les données synthétiques. 101

5.5 Comparaison des taux de rejet en fonction du côut de rejet r obtenus respectivement avec le SVMR [Grandvalet 2009] (courbes noires) et avec la règle de décision [Bartlett 2008, Herbei 2006, Wegkamp 2010] (courbes rouges). Ces résultats sont obtenus en utilisant les données médicales. 102

5.6 (a) Compromis erreur-rejet obtenu avec les données synthétiques, (b) données ECG, (c)avec les données forest ; en utilisant la méthode [Grandvalet 2009] (courbes noires) et en utilisant la méthode [Bartlett 2008, Herbei 2006, Wegkamp 2010] avec $\delta_o = 0.5$ (courbes rouges). 104

5.7 Haut : taux d'erreurs en fonction du taux de rejet. Bas : Variation du taux d'erreurs en fonction de la variation du taux de rejet 105

5.8 Haut : coût de la classification en fonction du taux de rejet. Bas : Sensitivité (courbe noire) et la prédictivité positive (courbe rouge) en fonction du taux de rejet. 105
5.9 Comparaison de différentes energies obtenues avec différentes méthodes, energie de Shannon (trait continu) et enrgie quadratique(trait interrompu). 109
5.10 Localisation de singularité sur le plan temps-fréquence. 110
5.11 (a) Signal avec interruption instantanée ; (b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon. 111
5.12 (a) Signal avec interruption instantanée en présence d'harmoniques ; (b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon. 111
5.13 (a) Signal réel (données réelles bruitées) avec de multiples interruptions ; (b) Spectrogramme du signal ; (c) Detection des instants t_1, t_2 et t_3, t_4 sur l'enveloppe de Shannon. 112
5.14 (a) Signal avec affaissement de tension ;(b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon. . . . 113
5.15 (a) Signal avec harmoniques et affaissement de tension ;(b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon. 113
5.16 (a) Signal réel avec affaissement de tension ;(b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon. 114
5.17 (a) Signal avec houle de tension ;(b) Spectrogramme du signal ; (c) Détection des instants t_1 et t_2 sur l'enveloppe de Shannon. 114
5.18 (a) Signal avec harmoniques et houle de tension ;(b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon. 115
5.19 (Signal réel avec houle de tension ;(b) Spectrogramme du signal ; (c) Détection des instants t_1 et t_2 sur l'enveloppe de Shannon. 115
5.20 a) Signal avec interruption instantanée ; b) Coefficients d'ondelettes à l'échelle 1 " Morlet" 116

Liste des tableaux

2.1 Exemple de nuage de données 6
2.2 Moyenne et Ecart type 6
2.3 Observations centrées réduites 7
2.4 *Quelques noyaux usuels* 39

3.1 *Comparaison entre les formulations primale et duale* 49
3.2 *Relation entre les contraintes primales et duales.* 54
3.3 Ensemble de partition 62
3.4 Condition d'optimalité pour les points I_w, I_C et I_0 64

4.1 *Coût associé à chaque décision* 81
4.2 Ensemble d'apprentissage réparti en 5 sous-ensembles 91
4.3 Situation des constraintes pour les cinq types d'exemples 93

A.1 Répartition des exemples d'apprentissage 125

Table des matières

1 Introduction générale **13**
1.1 Préambule . . . 13
1.2 Cadre de recherche . . . 14
1.3 Motivation . . . 15
1.4 Contribution . . . 16
1.5 Organisation de la thèse . . . 16

2 Apprentissage artificiel et aide à la décision **1**
2.1 Introduction . . . 1
2.2 Caractérisation de données . . . 1
2.2.1 Différents types de données . . . 1
2.2.2 Types d'attributs . . . 2
2.2.3 Natures d'attributs . . . 2
2.2.4 Données bruitées . . . 2
2.3 Prétraitement des données . . . 2
2.3.1 Normalisation de données . . . 2
2.3.2 Analyse en Composantes Principales (ACP) . . . 3
2.3.3 Selection d'attributs . . . 12
2.4 Objectifs de l'apprentissage . . . 15
2.4.1 La classification . . . 15
2.4.2 La segmentation . . . 16
2.4.3 L'estimation . . . 16
2.4.4 La régression . . . 16
2.5 L'apprentissage, aspects théoriques . . . 17
2.5.1 Apprentissage statistique . . . 18
2.5.2 Hyperplans séparateurs . . . 22
2.5.3 Un peu de géométrie dans $\mathbb{R}^d$ . . . 24
2.5.4 Hyperplan caractérisé par un vecteur dans $\mathbb{R}^d + 1$ . . . 25
2.6 Classification et apprentissage automatique . . . 28
2.7 Différentes méthodes de classification . . . 30
2.7.1 Méthode des k-plus proches voisins . . . 30
2.7.2 Arbres de décision . . . 30
2.7.3 Approche Bayesienne . . . 31
2.7.4 Réseaux de neurones . . . 32
2.7.5 Séparateurs à Vaste Marge (SVMs) . . . 33
2.7.6 Fonctions noyaux . . . 36
2.8 Conclusion . . . 40

3 Séparateurs à Vaste Marge (SVMs) 41
3.1 Introduction . . . 41
3.2 Objectifs d'un SVM . . . 42
3.2.1 Hyperplan optimal . . . 42
3.2.2 Marge maximale . . . 43
3.3 Formulation du problème SVM . . . 43
3.3.1 calcul de la marge . . . 43
3.3.2 position des contraintes . . . 45
3.4 Cas de données linéairement séparables . . . 46
3.4.1 Marge dure . . . 46
3.4.2 Marge souple . . . 50
3.5 Cas de données non linéairement séparables . . . 53
3.6 Algorithmes de résolution . . . 56
3.6.1 Simple SVM . . . 57

4 Optimisation du coût de la classification 73
4.1 Introduction . . . 73
4.2 Etat de la question . . . 74
4.2.1 Les deux natures de rejet . . . 75
4.2.2 Définition du rejet d'ambiguïté . . . 75
4.2.3 Définition du rejet de distance . . . 76
4.2.4 Le compromis erreur-rejet (ER) . . . 79
4.2.5 coût du rejet . . . 80
4.3 Position du problème . . . 81
4.4 Règle de Bayes . . . 82
4.5 Classifieur SVM avec coût optimal . . . 84
4.5.1 Construction de la fonction de perte . . . 84
4.5.2 Apprentissage du classifieur . . . 88
4.5.3 Algorithme . . . 92
4.5.4 Complexité du problème . . . 93
4.6 Conclusion . . . 94

5 Résultats expérimentaux 95
5.1 Préambule . . . 95
5.2 Performances du classifieur . . . 95
5.2.1 Bases de données . . . 95
5.2.2 Protocole expérimental . . . 97
5.2.3 Mise en évidence de la zone de rejet . . . 98
5.2.4 Mise en évidence du coût de la classification . . . 98
5.3 Localisation de perturbations électriques . . . 107
5.3.1 Construction d'une bases de données . . . 107

5.3.2 Méthode proposée 108
5.4 Conclusion 116

6 Conclusion et perspectives 119

A Différentes méthodes d'apprentissage d'un SVM 123
A.0.1 Formulation primale du problème 123
A.0.2 Formulation duale du problème 123
A.0.3 L'Algorithme SMO 124
A.0.4 La méthode adaptée 130

B Bilan scientifique relatif à la thèse 135

Bibliographie 137

Chapitre 1

Introduction générale

Sommaire

1.1 Préambule 13
1.2 Cadre de recherche 14
1.3 Motivation 15
1.4 Contribution 16
1.5 Organisation de la thèse 16

1.1 Préambule

Depuis bientôt un demi-siècle, les chercheurs en intelligence artificielle travaillent à programmer des machines capables d'effectuer des tâches qui requièrent de l'intelligence. Nous citerons l'aide à la décision, la reconnaissance de formes, le contrôle de processus, la prédiction, l'exploration de grandes bases de données (on dit aussi la fouille de données), tant il est vrai que si nous croulons sous les informations, il nous manque souvent la connaissance. Chacune de ces tâches et bien d'autres ont stimulé l'inventivité des chercheurs et donné lieu à de nombreuses réalisations impressionnantes.

Cependant, programmer des machines capables de s'adapter à toutes les situations et éventuellement d'évoluer en fonction de nouvelles contraintes est difficile. L'enjeu est de contourner cette difficulté en dotant la machine de capacités d'apprentissage lui permettant de tirer parti de son expérience. C'est pourquoi des recherches sur le raisonnement automatique et sur l'apprentissage se sont développées.

L'objectif principal de l'apprentissage artificiel est de permettre, à partir d'un ensemble de données une meilleur généralisation pour produire un processus décisionnel performant.

Dans le cadre de l'apprentissage supervisé, le but est de produire une fonction de décision à partir d'un algorithme d'apprentissage et d'une base d'apprentissage de façon à ce que les prédictions réalisées sur de nouvelles données conduisent à un minimum d'erreurs.

Les domaines liés au traitement et à l'analyse de données s'inscrivent dans une problématique de fouille de données. L'exploitation des données nécessite généralement des contraintes de temps réelles. Une caractérisation optimale des ces données ainsi que certains prétraitements, notamment liés à la taille des données sont souvent nécessaires afin d'aboutir à des processus décisionnels de complexités raisonnables.

1.2 Cadre de recherche

Ce travail de thèse s'inscrit dans le domaine de la reconnaissance de formes. Il est articulé autour d'un classifieur SVM dont le fondement mathématique est modifié afin d'optimiser le coût de la classification. Cette contribution rentre dans le cadre d'un projet de recherche ayant abouti à plusieurs résultats déjà publiés dans des journaux. A ce projet de recherche, ont adéré des enseignants chercheurs de l'Université Mouloud Mammeri de de Tizi-Ouzou (UMMTO) et des enseignants de l'Université de Haute Alsace (UHA), Laboratoire MIPS. Ce laboratoire m'a accueilli selon le programme PNE afin de finaliser ma thèse.

Durant mon séjour en France, j'ai pu créer une passerelle entre mon travail et la thématique de recherche du laboratoire MIPS de l'UHA. J'ai alors travaillé sur la reconnaissance de formes sur des signaux électriques. Il s'agit de l'identification et de la classification non intrusive des équipements électriques polluants. Cela consiste en la caractérisation des signatures électriques liées au fonctionnement d'un équipement. Il peut s'agir de l'apparition d'harmoniques, d'affaissements, de houles ou encore d'interruptions dans le signal électrique. C'est ce qui nous a motivés à explorer un puissant outil de traitement du signal, appelé Transformée de Stockwell. Cette dernière, bien adaptée aux signaux non-stationnaires, opère dans le domaine temps-fréquence.

Mon séjour en France a ouvert des perspectives de travail pour nos équipes respectives. les deux équipes s'ouvrent ainsi aux signaux électriques d'origine biologique d'une part et à la pollution du réseau électrique d'autre part. Les outils mathématiques requis pour la caractérisation et la classification sont respectivement, la Transformée de Stockwell et les SVMs.

Un nouveau projet de recherche est né suite à nos échanges. C'est un travail financé par la région Alsace et quelques entreprises comme SOCOMEC. L'un des objectifs de ce projet est la conception d'un prototype, implémenté sur un FPGA, pour l'identification et la classification des signaux électriques et biologiques. Ce projet démarrera en janvier 2015.

1.3 Motivation

Parmi les différents types d'algorithmes d'apprentissage définis ces dernières années, on trouve les Séparateurs à Vaste Marge (SVMs : *Support Vector Machines*) dont le créateur principal est Vladimir Vapnik [Vapnik 1998]. Le nombre de publications concernant cet algorithme (et ces variantes) est très important, aussi bien sur le plan théorique que sur le plan pratique, marquant par là même l'intérêt qu'ils suscitent dans la communauté scientifique. La raison principale de cet intérêt est que Vladimir Vapnik a développé une théorie statistique de l'apprentissage avec des fondements solides qui est à l'origine des SVMs.

Les SVMs sont des classifieurs basés sur deux idées clés, qui permettent de traiter des problèmes de discrimination de données linéairement ou non linéairement séparables en formulant le problème de classification comme un problème d'optimisation quadratique.

- La première idée est la notion de marge maximale. Le SVM est un modèle discriminant qui tente de minimiser les erreurs d'apprentissage tout en maximisant la marge séparant les données des deux classes. La maximisation de la marge est une méthode de régularisation qui réduit la complexité du classifieur.
- La deuxième idée, l'astuce du noyau (*kernel truck*) [Cristianini 2000, Cortes 1982, Smola 1998, Wahba 1998], qui fait d'ailleurs la force des SVMs concerne les données non linéairement séparables. Ces dernières sont alors projetées dans un espace de plus grande dimension dans lequel opère plus facilement la séparation des classes car les données y deviennent linéairement séparable.

De nombreux travaux ont démontré la supériorité des SVMs aux méthodes discriminantes classiques telles que les réseaux de neurones. Les SVMs offrent les meilleures performances sur plusieurs bases de données standards [Cortes 1982, Boser 1992, Wahba 1998]. Leur robustesse vis à vis de la dimensionalité des données et leur pouvoir accru de généralisation, font que les SVMs sont nettement plus avantageux. Ceci explique le nombre et la qualité des résultats expérimentaux obtenus avec ces algorithmes.

Malgré la grande capacité de généralisation des SVMs, les erreurs de classification ne peuvent pas être complètement éliminées et peuvent alors engendrer des conséquences sévères ; notamment dans le cas de certaines applications. Souvent dans les problèmes de décision, les exemples ambigus portent préjudice. Pour réduire les erreurs, on peut construire des classificateurs qui s'abstiennent de classer ces exemples ambigus, c'est à dire, ceux dont la probabilité d'appartenance à une classe est autour de 0.5 [Bartlett 2008]. De tels classifieurs sont appelés : classifieurs avec option de rejet. Cette option a été introduite depuis les travaux de CK.Chow [Chow 1970] sur le compromis erreur-rejet. La performance de tels classifieurs n'est pas dictée uniquement par le taux d'erreurs mais aussi par le coût de la classification

où on introduit le coût des exemples mal classés et le coût des exemples rejetés.

1.4 Contribution

Cette thèse s'inscrit dans cette problématique. Notre première contribution consiste à modifier le problème d'optimisation du SVM standard afin d'optimiser le coût de la classification. Ceci, en définissant une troisième décision qui consiste à rejeter les exemples ambigus d'un côté et en assignant des coûts asymétriques pour les deux types d'erreurs. Ceci est très explicite sur les résultats que nous avons publiés dans [Zidelmal 2009, Zidelmal 2013, Zidelmal 2012]. Un autre travail de classification avec rejet a été mené sur différentes bases de données tout en faisant une étude comparative entre les SVMs dont l'option de rejet a été introduite en modifiant la fonction de perte *"Hinge loss"* à une fonction de substitution *" Double hinge loss"* [**Amirou** 2014b]. La règle de décision retenue et basée sur l'interprétation probabilistique des SVMs réunie les avantages de la fonction *" hinge loss "* pour assurer la consistance de la solution et les avantages de la fonction logistique *"likelihood loss"* pour assurer une meilleur estimation des probabilités *a posteriori*.

Une autre contribution dans le projet de recherche, consiste à localiser automatiquement des perturbations dans le réseau électrique afin d'identifier les éléments polluants. Un tel travail a été effectué autour de la transformée de Stockwell combinée à l'energie de Shannon. L'approche proposée a été testée et validée sur des données simulées et sur des données réelles générées au sein du laboratoire MIPS. Ceci a été évalué à travers un article présenté à ICON'14 [**Amirou** 2014a].

1.5 Organisation de la thèse

Après cette partie introductive, le manuel est organisé comme suit :

Le deuxième chapitre présente le contexte d'étude et quelques principes de la reconnaissance de formes et d'aide à la décision. Dans ce chapitre, nous parlons de la fouille de données de manière générale, de la caractérisations et du prétraitement des données puis nous introduisons le cadre général de l'apprentissage artificiel. Relativement à ce dernier point, nous présentons la notion de classification et d'hyperplans séparateurs. Les principales méthodes de classification sont alors passées en revue.

Le troisième chapitre expose le fondement mathématique des SVMs vus comme un problème d'optimisation quadratique sous contraintes. Ce problème peut

être simplememnt énoncé comme suit : Maximiser la marge entre les deux classes, tout en commettant un minimum d'erreurs. Les notions d'hyperplan optimal et l'astuce du noyau reglant le problème de données non linéairement séparables y sont introduites. Les principaux algorithmes d'apprentissage, autrement dit, de résolutions du problème d'optimisation sont présentés. L'accent est cependant mis sur l'algorithme SimpleSVM que nous retenons pour y introduire l'option de rejet.

Le quatrième chapitre est consacré à l'introduction du rejet d'ambiguités dans un classifieur SVM. Ce dernier étant basé sur la minimisation du risque structurel, la fonction de perte linéaire est remplacée par une fonction de substitution introduisant une troisième décision (le rejet) selon la règle de Chow.

Le cinquième chapitre est composé de deux volets.

1. Le premier montre l'impact de la règle de décision construite sur certains problèmes de classification binaire. Les tests ont été effectués sur des données médicales et industrielles où l'option de rejet s'avère d'une très grande importance. Pour une meilleur lisibilité de nos résultats, nous avons choisi leur représentation graphique afin de visualiser les couloirs de rejet, la comparaison des classifieurs ainsi que le taux de rejet optimal.
2. Le deuxième volet de résultats est consacré à la localisation automatique des perturbations dans le réseau électrique. Les perturbations sont causées par la prolifération des charges non linéaires, on distingue :
   – les interruptions de tension.
   – la présence d'harmoniques.
   – les houles de tension.
   – les affaissements de tension.

   Des tests ont été effectués sur des données industrielles et les résultats y sont ulistrés.

Enfin, nous concluons sur ce travail en l'ouvrant sur quelques perspectives de recherche.

Chapitre 2

Apprentissage artificiel et aide à la décision

2.1 Introduction

En une vingtaine d'années, l'apprentissage artificiel est devenu une branche majeure des mathématiques appliquées, à l'intersection des statistiques et de l'intelligence artificielle. Son objectif est de réaliser des modèles qui apprennent par des exemples : il s'appuie sur des données numériques (résultats de mesures ou de simulations). L'objectif des chercheurs en intelligence artificielle vise à programmer des machines capables d'effectuer des tâches qui requièrent de l'intelligence. Ses domaines d'applications sont multiples : fouille de données (FD), bio-informatique, génie des procédés, aide au diagnostic médical, télécommunications, interface cerveau-machines, et bien d'autres. La fouille de données est la traduction du "**Data Mining**", terme récent (1995). Elle permet l'extractions **Automatique** d'informations **Predictives** à partir de gros volumes de données. Elle représente un mélange d'outils provenant de la Statistique, de l'Intelligence Artificielle et de l'Informatique (reconnaissance de formes).

2.2 Caractérisation de données

2.2.1 Différents types de données

Une donnée est une entité caractérisant un objet. Elle est constituée d'attributs. Elle est aussi appelée exemple, échéntillon, point, et souvent vecteur. Un ensemble de données est souvent représenté par une matrice. Soit $\mathcal{X}$ un ensemble de données à n éléments caractérisé chacun par p caractéristiques (attributs).
Un attribut est un descripteur appelé aussi variable, champs, caratéristique ou observation.

2.2.2 Types d'attributs

Un attribut peut être de nature **qualitative** ou **quantitative**.

- Il est qualitatif si on ne peut pas en faire une moyenne, sa valeur est d'un type définit (ex : masculin, féminin, couleur,...).
- Il est quantitatif s'il peut prendre une valeur entière ou reélle. On peut leur appliquer les opérateurs arithmétiques habituels (ex : poids, la taille, les dimensions d'un objet...).

2.2.3 Natures d'attributs

La valeur d'un attribut peut être nominale ou catégoriel, elle appartient à un ensemble fini et non ordonné (ex : couleur). Elle peut être ordinale (température) ou absolue (nombre). Ces différentes natures entrainent que les opérations qu'on peut faire sur ces attributs ne sont pas les mêmes.

2.2.4 Données bruitées

Dans le cas où une donnée contient une où plusieurs attributs non valides, on dira que les données sont bruitées. (ex : âge=200).

2.3 Prétraitement des données

2.3.1 Normalisation de données

Nous vons mentionné, que les attributs (valeurs des variables du modèle) sont en général exprimés dans des unités différentes, et ont des ordres de grandeurs différents. Il est donc nécessaire de pré-traiter ces valeurs pour qu'elles aient la même influence sur la construction de modèles. Afin d'uniformiser l'importance de chaque entrée sur la détermination des paramètres d'un classifieur par exemple, les observations doivent être normalisées. Chaque attribut suivra alors une distribution gaussienne **centrées et réduites**. Soit un nuage de points à N échantillons décrits par P attributs. Pour obtenir un nuage de poitns centré réduit, chaque observation x_i^k va être remplacée par :

$$a_i^k = \frac{x_i^k - \overline{x^k}}{\sigma^k}$$

où $k = 1, ..., P$ et $i = 1, ..., N$ avec $\overline{x^k}$, la moyenne des observations x_i^k et σ^k, leur écart type. Ces deux paramètres sont donnés par :

$$\overline{x^k} = \frac{1}{N} \sum_{i=1}^{N} x_i^k \tag{2.1}$$

$$\sigma^k = \sqrt{\frac{1}{N} \sum_{i=1}^{N} [x_i^k - \overline{x^k}]^2} \tag{2.2}$$

- Cette opération est importante si les ordres de grandeur des variables sont très différents.
- si on ne réduit pas le nuage : une variable à forte variance va influer sur l'efficacité du modèle.

2.3.2 Analyse en Composantes Principales (ACP)

2.3.2.1 Elements de base

Variance, covariance d'une variable aléatoire :
La variance traduit la dispersion de la distribution de la variable aléatoire autour de sa valeur moyenne. Elle est définie comme suit :

$$\boxed{\mu_2 = E((X - E(X))^2) = \sigma^2 = variance\ de\ X = V(X) = E(X^2) - E(X)^2}$$

On dit aussi que la variance traduit la notion d'incertitude. Plus la variance est faible, moins le résultat de l'expérience aléatoire est incertain. A la limite, une variable aléatoire de variance nulle conduit à des expériences strictement identiques (i.e. le phénomène est complètement déterministe, il n'y a donc plus aucune raison de garder la notion de variable aléatoire). Dans le cas d'un signal aléatoire centré et reduit, la variance exprime aussi l'énergie de celui-ci. Etant un carré, la dimension de la variance n'est pas celle de la moyenne. C'est pourquoi on utilise plus souvent l'écart type, noté σ, qui est la racine de la variance. La variance a également des propriétés intéressantes vis à vis de la combinaison linéaire de variables aléatoires : Soient X_1 et X_2 deux variables aléatoires

$$\boxed{V(X_1 + X_2) = V(X_1) + V(X_2) + 2cov(X_1, X_2)}$$

où $cov(X, Y)$ est la covariance des variables aléatoires X et Y définie par :

$$\boxed{cov(X, Y) = \mu_{1,1} = E(XY) - E(X)E(Y) = E[(X - E(X))(Y - E(Y))]}$$

La covariance peut être vue comme le moment centré conjoint d'ordre 1 de deux variables aléatoires. Si les deux variables aléatoires sont indépendantes, alors leur covariance est nulle (mais la réciproque n'est pas vraie en général).

Coefficients de corrélation :

La relation entre deux variables aléatoires peut être quantifiée par la covariance comme vue précédemment. Cependant, à l'image de la moyenne et de la variance, la covariance est un moment, donc possède une dimension ce qui la rend plus difficile à interpréter. C'est pourquoi on utilise plus généralement le coefficient de corrélation, indicateur sans dimension. La corrélation entre deux variables aléatoires X et Y est définie comme suit :

$$\boxed{\rho(X,Y) = cor(X,Y) = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}} = \frac{cov(X,Y)}{\sigma_X \ \sigma_Y} = \frac{\mu_{1,1}}{\sqrt{\mu_2(X) \ \mu_2(Y)}}}$$

Le coefficient de corrélation mesure la qualité de la relation linéaire entre deux variables aléatoires X et Y (i.e. de la forme $Y = aX + b$). S'il existe une relation linéaire entre X et Y alors $\rho(X,Y) = \pm 1$. On a les propriétés suivantes :

- $\forall X, Y \ : \rho(X,Y) \in [-1,1]$
- Si X et Y sont indépendantes, alors $\rho(X,Y) = 0$ (la réciproque n'est pas vraie en général).
- On dit que deux variables sont corrélées positivement si leur coefficient de corrélation est positif, négativement sinon.

Valeurs propres et vecteurs propres d'une matrice

Soit $A \in \ \mathbb{R}^{n \times n}$ une matrice,

- λ est une valeur propre de A , si $\exists \ X \in \ \mathbb{R}^n, \ X \neq 0$ tel que $AX = \lambda X$
- Un tel vecteur X est un vecteur propre de A associé à la valeur propre λ.

Le système linéaire $AX = \lambda X$ est résoluble car il admet toujours au moins une solution $X = 0$. Pour savoir si λ est une valeur propre de A , il faut savoir si le système linéaire $AX = \lambda X$. (c'est à dire, $(A - \lambda \mathbb{I}_n)X = 0$ admet une solution autre que $X = 0$). Ceci est vérifié si $det(A - \lambda \mathbb{I}_n) = 0$. Le polynôme $P(\lambda) = det(A - \lambda \mathbb{I}_n)$ est appelé polynôme caractéristique de A.

2.3.2.2 Principe de l'ACP

Pour réduire le nombre de facteurs (composantes), l'ACP détermine un sous-espace vectoriel de dimension plus réduite que le nombre de facteurs (composantes initiales), dans lesquels la répartition des observations (points) est préservée au mieux. Le critère retenu sur la répartition est l'inertie totale du nuage de points. L'ACP se présente donc comme une méthode de projection linéaire qui maximise l'inertie du nuage des points [Jolliffe 1986].

2.3.2.3 Objectif de l'ACP

- Réduction de la dimension des vecteurs, donner leur approximation avec p variables ($p < P$). Autrement dit, on cherche à définir p nouvelles variables, combinaisons linéaires des P variables initiales qui feront perdre le moins d'informations possible. Ces nouvelles variables seront appelées **composantes principales**. Les axes qu'elles déterminent sont des axes principaux et les formes linéaires associées sont des facteurs principaux.
- Représentation graphique "optimale" des observations minimisant la déformation du nuage de points dans un sous-espace de dimension $p \ll P$.

2.3.2.4 Mise en oeuvre de d'ACP

L'analyse en composantes principales ne s'applique que sur des données quantitatives.

- L'idée fondamentale de l'ACP est la suivante : considérons un nuage de N points (individus) à P dimensions. On cherche à trouver une représentation bi-dimensionnelle par exemple, pour une meilleur lecture. Dans ce nouvel espace, la projection du nuage de points doit être la moins déformée possible, la plus fiable possible.
- Notons que comme toute méthode statistique, le nombre d'individus et le nombre d'attributs doivent être relativement importants pour que les conclusions de l'ACP aient un sens.

Pour illustration, nous considèrons les données de la table 2.1 qui est composé de 10 individus (N=10), décrits chacun par 4 observations ou attributs (poids, taille, âge, note), P=4. La moyenne et l'écart type de chaque obsevation sont donnés en table 2.2. Les observations centrées réduites du nuage de données considéré sont données par la table 2.3.

2.3.2.5 Inertie totale d'un nuage de points

L'inertie totale ou variance totale du nuage de points est donnée par :

$$\mathcal{I}_g = \frac{1}{N}[\sum_{i=1}^{N}(x_i^1 - \overline{x^1})^2 + .. + \sum_{i=1}^{N}(x_i^k - \overline{x^k})^2 + .. + \sum_{i=1}^{N}(x_i^P - \overline{x^P})^2] \tag{2.3}$$

$$\mathcal{I}_g = var(x^1) + var(x^2) + .. + var(x^k) + .. + var(x^P) \tag{2.4}$$

Le centre de gravité G d'un nuage de points est : $G(\overline{x^1}, .., \overline{x^k}, .., \overline{x^P})$ où $\{\overline{x^1}, .., \overline{x^k}, .., \overline{x^P}\}$ sont les moyennes respectives de chaque observation.

TAB. 2.1 – Exemple de nuage de données

Individu	Poids	Taille	Age	Note
1	45	1.5	13	14
2	50	1.6	13	16
3	50	1.65	13	16
4	60	1.75	15	9
5	60	1.7	14	10
6	60	1.7	14	7
7	70	1.6	14	8
8	65	1.6	13	13
9	60	1.55	15	17
10	60	1.7	14	11

TAB. 2.2 – Moyenne et Ecart type

Val.stat	poids	taille	âge	note
moyenne	58.5	1.64	13.8	12.0
écart type	7.84	0.08	0.79	3.50

Remarque 2.1

- *L'inertie totale d'un nuage de points est donnée par la somme des variances.*
- *L'inertie mesure la dispersion totale du nuage de points.*

Quand on projette un ensemble de points dans un espace de dimension inférieure à celle de l'espace initial, la distance entre les points ne peut que diminuer. Pour trouver le meilleur plan de projection :

- On cherche le plan dans lequel la distance entre les points projetés demeurent en moyenne, maximale. On recherche alors les axes portant le maximum d'inertie ce qui équivaut à la construction de nouvelles variables (auxquelles sont associés ces axes) de variance maximale.

TAB. 2.3 – Observations centrées réduites

Individu	Poids	Taille	Age	Note
1	-1.72	-1.72	-1.01	0.57
2	-1.09	-0.45	-1.01	0.14
3	-1.09	0.19	-1.01	0.86
4	0.19	1.47	1.52	-0.86
5	0.19	0.83	0.25	-0.57
6	0.19	0.83	0.25	-1.43
7	1.47	-0.45	0.25	-1.14
8	0.83	-0.45	-1.01	0.29
9	0.19	-1.09	1.52	1.43
10	0.83	0.83	0.25	-0.29

- En d'autres termes, on effectue un changement de repère dans $\mathbb{R}^P$ de façon à se placer dans un nouveau système de représentation où le premier axe apporte le plus possible de l'inertie totale du nuage, le deuxième axe le plus possible de l'inertie non prise en compte par le premier axe, et ainsi de suite. De tels axes sont qualifiés de principaux.
- On définit P droites orthogonales les unes aux autres qui permettent de définir un repère orthonormé. Ces P droites sont les P axes principaux.
- Les coordonnées d'un individu dans ce nouveau repère s'expriment par rapport à de nouveaux caractères appelés composantes principales. L'objectif est que pour chaque individu, seules quelques coordonnées aient une valeur importante et que la plupart soient presque nulles.

2.3.2.6 Dépendance entre attributs

Comme la matrice des données a été centrée et réduite , on va travailler à présent sur des variables a^k telles que :

$$\overline{a^k} = 0 \; et \;\; \sigma^k = 1$$

Construisons la matrice des covariances entres les vecteurs d'attributs.

$$V = \begin{bmatrix} var(a_1) & cov(a_1, a_2) & ..... & cov(a_1, a_p) \\ cov(a_2, a_1) & var(a_2) & ..... & cov(a_2, a_p) \\ cov(a_3, a_1) & cov(a_3, a_2) & ..... & cov(a_2, a_p) \\ cov(a_p, a_1) & cov(a_{p,} a_2) & ..... & var(a_p) \end{bmatrix} \tag{2.5}$$

Remarque 2.2 *:*

- *Dans le cas où les variables sont centrées réduites, la variance de chaque variable vaut 1.*
- *L'inertie totale est alors égale à P (nombre de variables).*
- *La trace de la matrice V représente la somme des éléments de la diagonale.*

$$\mathrm{Tr(V)} = \mathrm{P}$$

$$V = \begin{bmatrix} 1 & cov(a_1, a_2) & ..... & cov(a_1, a_p) \\ cov(a_2, a_1) & 1 & ..... & cov(a_2, a_p) \\ cov(a_3, a_1) & cov(a_3, a_2) & ..... & cov(a_2, a_p) \\ cov(a_p, a_1) & cov(a_{p,} a_2) & ..... & 1 \end{bmatrix} \tag{2.6}$$

Pour des données centrées et réduites, cette matrice est égale à la matrice de corrélation

$$R = \begin{bmatrix} 1 & R(a_1, a_2) & ..... & R(a_1, a_p) \\ R(a_2, a_1) & 1 & ..... & R(a_2, a_p) \\ R(a_3, a_1) & R(a_3, a_2) & ..... & R(a_2, a_p) \\ R(a_p, a_1) & R(a_p, a_2) & ..... & 1 \end{bmatrix} \tag{2.7}$$

Cette matrice est symétrique puisque : $R(a_k, a_j) = R(a_j, a_k)$, la fonction de corrélation est symétrique. L'examen de la matrice de corrélation est intéressant : elle permet de repérer immédiatement les caractères fortement corrélés et les caractères qui ne sont pas du tout corrélés. Dans le cas des données centrées réduites de la table 2.2, la matrice de corrélation est :

$$R = \begin{pmatrix} & Poids & Taille & Age & Note \\ Poids & 1 & 0.367 & 0.485 & -0.568 \\ Taille & 0.367 & 1 & 0.396 & -0.629 \\ Age & 0.485 & 0.396 & 1 & -0.322 \\ Note & -0.568 & -0.629 & -0.322 & 1 \end{pmatrix} \tag{2.8}$$

2.3.2.7 Analyse de la matrice de corrélation

- Le calcul des valeurs propres et des vecteurs propres de R fournissent toute l'information recherchée.
- La matrice R étant par sa nature, symétrique et définie positive, ses valeurs propres sont positives et réelles et ses vecteurs propres sont réels.

dans le cas de la matrice (2.8), Les valeurs propres et les vecteurs propres qui leurs sont associés sont :

$$\begin{array}{ll} \lambda_1 = 2.391 \rightarrow & \overrightarrow{V_1} = (0.5080;\ 0.5038;\ 0.4453;\ 0,5383) \\ \lambda_2 = 0.750 \rightarrow & \overrightarrow{V_2} = (0.3065;\ 0.4641;\ 0.7058;\ 0.4381) \\ \lambda_3 = 0.584 \rightarrow & \overrightarrow{V_3} = (0.6593;\ 0.5253;\ 0.4712;\ 0.2593) \\ \lambda_4 = 0.274 \rightarrow & \overrightarrow{V_4} = (0.4619;\ 0.5042;\ 0.2855;\ 0.6715) \end{array} \tag{2.9}$$

2.3.2.8 Les axes principaux

On appelle axes principaux d'inertie, les axes dont la direction est donnée par les vecteurs propres de R. Il y'en a P.

- Si quelques valeurs propres ont des valeurs bien plus importantes que les autres, cela signifie que l'essentiel des informations est donné par les axes principaux correspondants.
- Le premier axe est celui associé à la plus grande valeur propre . On le note u^1.
- Le deuxième axe est celui associé à la deuxième valeur propre . On le note u^2 etc...
- la situation idéale est bien entendu, lorsque 1 ou 2 valeurs propres sont très importantes par rapport aux autres.

Notons c^k, $(k = 1, .., P)$, les composantes principales, c_i^k dénotant le k^{eme} caractère principal de l'individu i.
On a naturellement une relation reliant les attributs originaux aux attributs synthétiques. Chaque composante c_i^k est donnée par le produit scalaire entre le vecteur ligne a_i et le vecteur propre V_k avec $k = 1, ..., P$.

$$c_i^k = a_i^1 V_k^1 + a_i^2 V_k^2 + a_i^3 V_k^3 + .. + a_i^P V_k^P$$

Chaque individu i sera alors représenté par des attributs artificiels comme suit :

$$Individu\ i \begin{pmatrix} c_1^1 & c_1^2 & ... & c_1^P \\ c_2^1 & c_2^2 & ... & c_2^P \\ ... & ... & ... & .... \\ c_i^1 & c_i^2 & ... & c_I^P \\ ... & ... & ... & .... \\ c_N^1 & c_N^2 & ... & c_N^P \end{pmatrix} \tag{2.10}$$

2.3.2.9 Les composantes principales

Choix des axes principaux : La meilleure représentation des données au moyen de $p < P$ caractères s'obtient en ne prenant en compte que les p premières composantes principales.

Remarque 2.3

- *Ces vecteurs sont orthogonaux deux à deux* $< \overrightarrow{V_j}, \overrightarrow{V_k} >= 0$. *Ils constituent alors un espace orthonormé.*
- *Les 4 vecteurs* $\overrightarrow{V_k}$ *sont unitaires,* $|\overrightarrow{V_k}| = 1$.
- *L'ordre de grandeur des valeurs propres les unes par rapport aux autres indiquent leurs importances.*
- *L'importance d'une valeur propres est donnée par son inertie :* $\mathcal{I}_k = \frac{\lambda_k}{P}$.
- *Si* $\lambda_1 \gg \lambda_2 > \lambda_3 > ..$, *cela signifie que l'essentiel de l'information sera donné par l'axe défini par le vecteur associé à* λ_1
- λ_k *représente la variance le long de l'axe principal porté par le vecteur* $\overrightarrow{V_k}$.

La qualité des estimations auxquelles conduit l'ACP dépend, de façon évidente, du choix de p, c'est-à-dire du nombre de composantes retenues pour reconstituer les données, ou encore de la dimension du sous-espace de représentation. De nombreux critères de choix de p ont été proposés dans la littérature. **L'éboulis** est le graphique figure **??** présentant la décroissance des valeurs propres. Le principe consiste à rechercher, s'il existe, un **coude** (changement de signe dans la suite des différences d'ordre 2) dans le graphe et de ne conserver que les valeurs propres jusqu'à ce coude. En considérant dans l'exemple précédent, les deux premières valeurs propres λ_1 et λ_2, les individu i seront représenté comme suit :

$$Individu\ i \begin{pmatrix} c_1^1 & c_1^2 \\ c_2^1 & c_2^2 \\ ... & ... \\ c_i^1 & c_i^2 \\ ... & ... \\ c_N^1 & c_N^2 \end{pmatrix} \tag{2.11}$$

Les 10 individus peuvent maintenant être projetés sur un espace à deux dimension, $p = 2$ (voir figure 2.2)

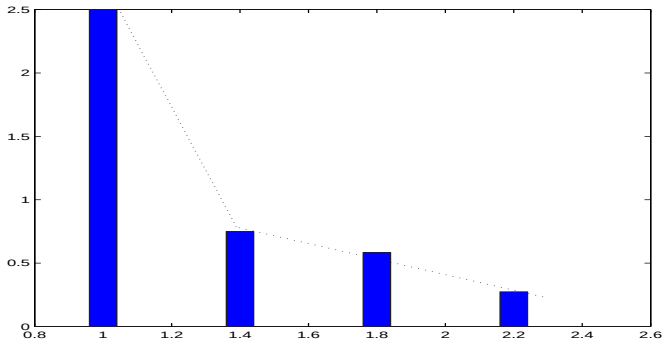


FIG. 2.1 – *éboulis des valeurs propres*

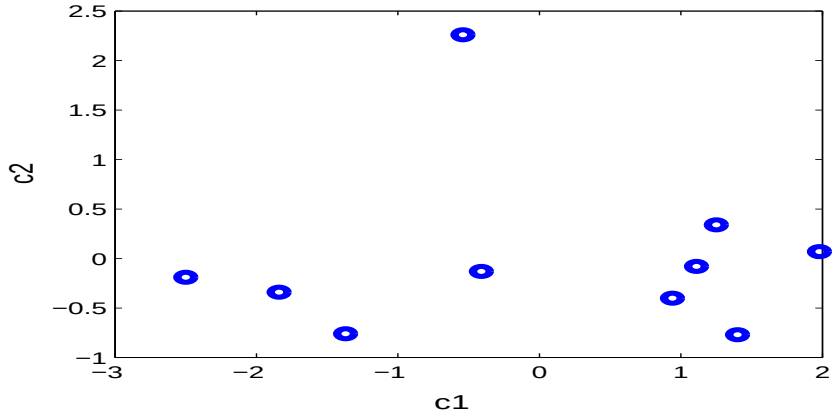


FIG. 2.2 – *Projection de l'espace d'entrée sur 2D.*

2.3.3 Selection d'attributs

Dans cette section, nous présentons quelques méthodes de sélection de caractéristiques ou d'attributs. Le but reste toujours la réduction du nombre de caractéristiques ou d'attributs.

2.3.3.1 *Sequential Forward Selection* (SFS)

La sélection séquentielle croissante (SFS) est la première méthode proposée pour la sélection de caractéristiques. Cette méthode a été proposée en 1963 par Marill et Green [Marill 1963]. Une approche heuristique de recherche est utilisée dans cette méthode, en commençant par un ensemble vide de caractéristiques. A chaque itération, la meilleure caractéristique parmi celles qui restent sera sélectionnée, supprimée de l'ensemble de départ et ajoutée au sous-ensemble des caractéristiques sélectionnées Algorithme 1. Le processus de sélection continue jusqu'à un critère d'arrêt.

2.3.3.2 *Sequential Backward Selection* (SBS)

La sélection séquentielle arrière SBS méthode similaire au SFS à été proposée en 1971, par Whitney [Whitney 1971]. A la différence de la méthode SFS, cette méthode commence par l'ensemble de toutes les caractéristiques et à chaque itération, la caractéristique la plus mauvaise sera supprimée Algorithme 2. Bien que les deux méthodes SFS et SBS semblent similaires, Ahan et Bankert [Aha 1995] ont montré que la méthode SBS est plus performante parce qu'elle prend en considération l'interaction d'une caractéristique avec un ensemble de caractéristiques plus large, contrairement au SFS qui ne prend en considération que l'interaction de cette caractéristique avec le sous-ensemble déjà sélectionné. Par ailleurs, l'évaluation des sous ensembles de grande taille avec la méthode SBS pose un problème au niveau du temps de calcul.

2.3.3.3 GSFS et GSBS

En 1978, des généralisations des méthodes SBS et SFS appelées GSFS et GSBS, sont proposées par Kittler [Kittler 1978]. Dans ces méthodes, l'auteur propose, au lieu d'inclure (ou exclure) une caractéristique à chaque itération, d'inclure (ou exclure) un sous ensemble de caractéristiques. Ces algorithmes ont montré une meilleure performance par rapport aux méthodes initiales, mais ils conservent toujours les mêmes problèmes que les méthodes de base.

Algorithme 1 Algoritme SFS

Entrée :

$F = \{f_1, f_2, .., f_N\}$

M = taille de l'ensemble final

Sorties :

$E = \{f_{s1}, f_{s2}, .., f_{sM}\}$

E = ∞

Pour $i = 1$ a M $Faire$

Pour $j = 1$ a $|F|$ $Faire$

$Evaluer$ f_j $\cup$ E

Fin Pour

$f_{max} =$ $meilleure$ f_j

$E =$ $E \cup f_{max}$, $F =$ $F|f_{max}$

Fin Pour

Retourner E

2.3.3.4 SFFS et SFBS

Deux autres méthodes de la famille (FS, BS) qui limitent les inconvénients des méthodes décrites ci-dessous, appelées SFFS (*Sequential Floating Forward Selection*) et SFBS (*Sequential Floating Backward Selection*) sont proposées en 1994 par Pudil [Pudil 1994]. Ces méthodes consistent à utiliser une fois l'algorithme SFS de manière à ajouter une variables, puis à utiliser r fois l'algorithme SBS afin d'en supprimer r. Ces étapes sont alors réitérées jusqu'à l'obtention du critère d'arrêt. La dimension du sous-ensemble à chaque étape sera alors dépendante des valeurs de l et r. Les valeurs optimales de ces paramètres ne pouvant pas être déterminées théoriquement, les auteurs proposent de les laisser flottantes au cours du processus de sélectionné de se rapprocher au maximum de la solution optimale.

Algorithme 2 Algoritme SBS

Entrée :

$F = \{f_1, f_2, .., f_N\}$

M = taille de l'ensemble final

Sorties :

$E = \{f_{s1}, f_{s2}, .., f_{sM}\}$

E = F

Pour $i = 1$ a $M - N$ $Faire$

Pour $j = 1$ a $|E|$ $Faire$

$Evaluer$ $E|f_j$

Fin Pour

$f_{min} =$ la $plus$ $mauvaise$ f_j

$E =$ $E|f_{min}$, $F =$ $F|f_{min}$

Fin Pour

Retourner E

2.3.3.5 Méthode de *Branch and Bound*

Ce type de méthode est lié à la modélisation du problème de recherche du meilleur sousensemble sous forme de graphe. Alors les algorithmes développés sur les graphes sont applicables, par exemple la methode "*Branch and Bound*". Cette méthode consiste à énumérer un ensemble de solutions d'une manière intelligente en ce sens que, en utilisant certaines propriétés du problème en question, cette technique arrive à éliminer des solutions partielles qui ne mènent pas à la solution que l'on recherche. Pour ce faire, cette méthode se dote d'une fonction qui permet de mettre une borne sur certaines solutions pour soit les exclure, soit les maintenir comme des solutions potentielles. Bien entendu, la performance de cette méthode dépend de la qualité de cette fonction d'évaluation partielle. Cette technique a été appliquée pour résoudre des problèmes de sélection de caractéristiques en 1977 par [Narendra 1977]. Son principe est de construire un arbre de recherche où la racine représente l'ensemble des caractéristiques et les autres noeuds représentent des sous-ensembles de caractéristiques. En parcourant l'arbre de la racine jusqu'aux

feuilles, l'algorithme enlève successivement la plus mauvaise caractéristique du sous ensemble courant (noeud courant) qui ne satisfait pas le critère de sélection. Une fois que la valeur attribuée à un noeud est plus petite qu'un seuil (*bound*), les sous-arbres de ce noeud sont supprimés. Cette technique garantit de trouver un sous-ensemble optimal de caractéristiques à condition d'utiliser une fonction d'évaluation monotone. L'inconvénient de cette méthode est son temps de calcul qui croût vite avec l'augmentation du nombre de caractéristiques et qui devient impraticable à partir d'un certain nombre (30 caractéristiques). Une amélioration de cette méthode en utilisant d'autres techniques de recherche dans l'arbre afin d'accélérer le processus de sélection a été proposée dans [Chen 2003, Somol 2004].

2.3.3.6 Les algorithmes génétiques

Les algorithmes génétiques ont été utilisés dans le domaine de la sélection de caractéristiques afin d'accélérer la recherche et d'éviter les optima locaux. De nombreuses études rapportées dans la littérature ont montré que les méthodes qui utilisent les AGs comme technique de recherche ont donné de meilleurs résultats que les résultats obtenus par les autres méthodes de sélection [Jain 1997, Kuncheva 1999, Ishibuchi 2000].

2.4 Objectifs de l'apprentissage

Pour mettre en oeuvre un projet d'apprentissage et évaluer ces performances, plusieurs étapes sont nécessaire :

- Comprendre et analyser les objectifs de l'application.
- Créer une base de données pour la mise au point de l'application.
- Prétraitement et nettoyage des données.
- Analyse statistique des données (réduction de la dimension, projection, etc...).
- Identifier le type de problèmes (discrimination, *clustering*, etc...) et choisir un algorithme.
- Evaluer les performances de l'algorithme.

Les objectifs poursuivis dans ce domaine sont divers :

2.4.1 La classification

Dans le cas d'un problème de classification, on cherche à prédire la classe à laquelle appatient un échantillon de données :

- Par construction d'un modèle arborescent (arbres de décision).
- Par approche probabiliste (Classifieurs Bayesiens).
- Par construction d'un modèle non interprétable par un humain et basé sur l'apprentissage automatique (Réseaux de neurones, Support Vector Machines).
- Par approximation et interpolation (Les K Plus Proches Voisins KPPv).

2.4.2 La segmentation

L'objectif de la segmentation est le suivant : on dispose de données **non étiquetées**. On souhaite les regrouper en nous basant uniquement sur leurs ressemblances. La segmentation consiste à partitionner par exemple une image ou un signal en partitions homogèmes. La tâche consiste alors à repérer des groupes de points qui se ressemblent. Certains algorithmes construisent une représentation géométrique interprétable par un humain. Souvent des algorithmes d'apprentissage non supervisé sont utilisés.

2.4.3 L'estimation

On peut disposer d'un ensemble de points et d'autres points dont les attributs ne sont pas tous évalués. On doit alors déterminer la valeur de ces attributs manquants. C'est alors un problème d'estimation. Notons que dans tous les problèmes réels, la notion de corrélation est pesante : l'extraction d'informations repose sur la recherche de corrélations entre des données. Ces corrélations peuvent être linéaires ou non linéaires.

2.4.4 La régression

Nous avons vu dans qu'il est possible d'utiliser l'approche par fonction noyau pour résoudre des tâches de régression non linéaire, c'est-à-dire de prédiction d'une valeur réelle à partir d'une entrée prise dans un espace, éventuellement multidimensionnel, typiquement $\mathbb{IR}^d$. Le problème inductif revient à chercher une fonction $h(x) = y \in \mathbb{R}$ telle que, pour tous les points d'apprentissage $(x_i, u_i)_{1 \leq i \leq m}, h(x_i)$ soit le plus " proche " possible de u_i, tout en se plaçant dans un espace d'hypothèses contrôlé pour éviter le risque de sur-apprentissage. Une analyse de la convergence du risque empirique sur le risque réel, mesuré en terme d'écart quadratique, ainsi que la recherche d'une solution parcimonieuse en terme d'exemples support, a conduit Vapnik [Vapnik 1998] à proposer d'utiliser la fonction de perte suivante :

$$|y - h(x)|_\varepsilon = max \ \ \{0, \ \ |y - h(x)| \ - \ \varepsilon\}$$

Alors, afin d'estimer la régression linéaire :

$$h(x) = \langle w.x \rangle + w_0$$

avec une précision de ε, on minimise :

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} |u_i - f(x_i)|_\varepsilon$$

Comme dans le cas de la classification, on peut généraliser à la régression non linéaire en passant par un espace de redescription des entrées grâce à l'utilisation de fonctions noyau. L'introduction de multiplicateurs de Lagrange conduit au problème d'optimisation dans lequel $C > 0$ et choisie. [Christopher 1998]

2.5 L'apprentissage, aspects théoriques

L'apprentissage statistique de données utilise des vecteurs x de dimension d. Dans le cas de l'apprentissage supervisé, chaque individu x est accompagné de son étiquette y. Cette étiquette permet de caractériser l'individu (son groupe d'appartenance, sa valeur, sa structure,...). Le but de l'apprentissage est de permettre à une machine de retrouver l'étiquette pour un élément donné.

Definition 2.5.1 ***La base d'apprentissage :*** *$(\mathcal{X}_n^d, \mathcal{Y}_n)$ regroupe l'ensemble des exemples connus générés par la loi sous-jacente inconnue $P(x, y) \in \mathcal{P}$. La taille de cette base est notée n et chaque exemple (ou individu) est noté x_i, avec $i \in [1, n]$.*

Les exemples appartiennent à un espace de dimension d (le plus souvent $\mathbb{R}^d$). Les étiquettes associées à chaque exemple sont la plupart du temps représentées par des scalaires. Les cas les plus fréquents sont $y_i \in \mathbb{R}$ pour la régression et $y_i \in \{-1, 1\}$ pour la discrimination.

Definition 2.5.2 ***La base de test :*** *$(\mathcal{X}_t^d, \mathcal{Y}_t)$ représente un ensemble quelconque d'exemples générés par la même loi $P(x, y) \in \mathcal{P}$.*

La théorie statistique de l'apprentissage, (voir Vapnik pour plus de détails voir [Bernhard 2002, Smola 1998, Vapnik 1982, Vapnik 1998]) utilise la notion de capacité pour exprimer cette quantité.

Definition 2.5.3 *La capacité est définie comme étant le cardinal de l'espace des hypothèses. Dans le cas où cet espace est de dimension infinie, la capacité est définie par la VC-dimension.*

Definition 2.5.4 *La dimension de Vapnik-Chervonenkis où VC-dimension d'une famille de fonctions F est le nombre maximum d'exemples d'un ensemble de données D qui peuvent être séparables par la famille de fonctions F (pour plus de détails consulter [Vapnik 1982, Vapnik 1998]).*

La VC-dimension Selon la théorie statistique de l'apprentissage, la dimension de Vapnik-Chervonenkis (La VC-dimension) donne une notion de capacité d'un ensemble de fonctions. Elle est définie en indiquant combien de points peuvent être séparés de toutes les manières possibles par les fonctions de cet ensemble. A partir de cette VC-dimension, la théorie de Vapnik-Chervonenkis fourni des bornes sur le risque.

Theorem 2.5.1 *Soit $\mathcal{F}$ l'ensemble des hyperplans dans*

$$\mathbb{R}^p, \mathcal{F} = x \mapsto sign(< w.x > +b), w \in \mathbb{R}^p, b \in \mathbb{R}$$

La dimension de Vapnik-Chervonenkis [Vapnik 1998] pour $\mathcal{F}$ est egale a $p+1$.

2.5.1 Apprentissage statistique

Effectuer une classification consiste à déterminer une règle de décision capable, à partir d'observations externes, d'assigner un objet à une classe parmi plusieurs. Le cas le plus simple consiste à discriminer deux classes. D'une manière plus formelle, la classification bi-classe revient à estimer une fonction $f : \mathcal{X} \longmapsto \{+1, -1\}$ à partir d'un ensemble d'apprentissage constitué de couples (x_i, y_i). On notera ici une hypothèse fondamentale pour toute la théorie statistique de l'apprentissage, à savoir que les exemples sont tirés indépendamment les uns des autres, selon une même distribution de probabilités $P(x, y)$ inconnue, tels que :

$$(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}; \quad i = 1, \ldots, N; \quad \mathcal{Y} = \{+1, -1\}$$

De sorte à ce que f classifie correctement des exemples inconnus (x_t, y_t). Par exemple, nous pouvons assigner x_t à la classe $(+1)$ si $f(x_t) \geq 0$ et à la classe (-1) sinon. Les exemples inconnus sont supposés suivre la même distribution de probabilité $P(x, y)$ que ceux de l'ensemble d'apprentissage. La meilleure fonction f est celle obtenue en minimisant le risque :

$$R(f) = \int L[f(x), y] dP(x, y) \tag{2.12}$$

Où L désigne une fonction de coût comme :

$$L[f(x), y] = [f(x) - y]^2 \tag{2.13}$$

Le risque (2.12) ne peut pas être directement minimisé dans la mesure où la distribution de probabilité sous-jacente $P(x, y)$ est inconnue. Aussi, il faut chercher une fonction de décision proche de l'optimale, à partir de l'information dont on dispose, c'est à dire l'ensemble d'apprentissage et la classe de fonctions F à laquelle la solution f appartient. Pour ce faire, on approxime le minimum du risque théorique par le minimum du risque empirique. Le risque empirique est l'une des quantités que l'on cherche à minimiser lors de l'apprentissage. Ce risque est la moyenne des coûts mesurés pour chaque exemple d'apprentissage. Il prend alors la forme :

$$R_{emp} = \frac{1}{N} \sum_{i=1}^{N} [f(x_i) - y_i]^2 \tag{2.14}$$

où N est le nombre de vecteurs d'apprentissage. Il est possible de donner des conditions au classifieur pour qu'asymptotiquement (*quand* $N \rightarrow \infty$), le risque empirique (2.14) converge vers le risque (2.12).

Theorem 2.5.2 *Pour que le principe de Minimisation du Risque Empirique (MRE) soit consistant, indépendamment de la distribution de probabilité des exemples, il suffit que la famille de fonctions que le système d'apprentissage est capable d'implémenter ait une VC-dimension finie .*

Ce théorème donne seulement une condition sufisante concernent la consistance de l'approche MRE. La théorie développée par Vapnik et Chervonenkis fournie également des bornes sur la vitesse de convergence du processus d'apprentissage. Ainsi, ils ont démontré que pour $N > h$ où h est la VC-dimension de la famille F et N est la taille de l'ensemble d'exemples, l'inégalité suivante (bornant le risque)donnée par :

$$R(f) \leq R_{emp}(f) + \sqrt{\frac{h(ln(\frac{2N}{h}) + 1) - ln(\frac{\delta}{4})}{N}} \tag{2.15}$$

sera vérifiée avec une probabilité au moins égale à $(1 - \delta)$ et $R_{emp}(f)$, le risque empirique défini par (2.14).
Cette borne n'est qu'un exemple. Des formulations du même type ont été démontrées pour d'autres fonctions de perte et d'autres mesures de complexité[Scholkopf 1999].

Cette inégalité donne une borne pour le risque à partir du risque empirique dépendant de la taille de l'échantillon d'apprentissage et de propriétés intrinsèques au classificateur (la VC-dimension), mais pas de la distribution des observations d'apprentissage, ce qui permet d'assurer une bonne généralisation du classificateur sans avoir besoin de connaître a priori la distribution des observations. Le but recherché ici est de minimiser l'erreur de généralisation $R(f)$

2.5.1.1 La généralisation

Les SVMs présentent également de bonnes performances en terme de généralisation [Burges 1998]. La généralisation est la faculté d'un classificateur à prédire correctement les classes de nouvelles observations et non pas seulement les classes des observations d'apprentissage. Cette inégalité donne une borne pour le risque à partir du risque empirique dépendant de la taille de l'échantillon d'apprentissage et de propriétés intrinsèques au classificateur (la VC-dimension), mais pas de la distribution des observations d'apprentissage, ce qui permet d'assurer une bonne généralisation du classificateur sans avoir besoin de connaître a priori la distribution des observations. La borne sur l'erreur de généralisation dépend de deux termes :

- $R_{emp}(f)$ Le risque empirique dit aussi biais.
- $\sqrt{\frac{h(ln(\frac{2N}{h})+1)-ln(\frac{\delta}{4})}{N}}$ La largeur de l'intervalle de confiance dite aussi variance.

Le biais caractérise l'écart des estimations aux exemples d'apprentissage, et la variance exprime la sensibilité du modèle aux données utilisées pour son apprentissage.

L'objectif est donc de trouver un compromis entre qualité de l'apprentissage et capacité de généralisation.

- L'intervalle de confiance est d'autant plus large que la dim VC de l'espace F est élevée.
- D'autre part, le risque empirique est d'autant plus élevé que la dim VC de l'espace F est réduite.

Capacité de généralisation : Un algorithme est efficace s'il est capable de donner des résultats similaires (en termes d'erreur) à la fois sur la base d'apprentissage et sur la base de test. C'est ce que l'on appelle la capacité à généraliser.

2.5.1.2 Principe de minimisation du risque structurel

En général, ce problème est difficile, mais dans le cadre de la théorie de Vapnik et Chervonenkis la recherche de la bonne classe de fonctions F peut être basée sur la minimisation de la borne donnée par (2.15). Cette idée est à la base

du principe inductif de minimisation du risque. Ce principe minimise la borne du risque structure en agissant simultanément sur le risque empirique et la largeur de l'intervalle de confiance en utilisant la dimension de Vapnik-chevonenkis comme variable de contrôle. Pour avoir une idée de comment nous pouvons contrôler le risque lorsque la dimension VC de F varie, munissons-nous maintenant de N espaces de fonctions emboîtés F_i, Considérons une famille imbriquée de classes de fonctions, $F_1 \subset F_2 ... \subset F_N$ dont les capacites h_i sont ordonnées et finies, avec une $VCdimension$ non décroissante $h_1 \leq h_2 \leq ... \leq h_N$ et $f_1, ..., f_k$, les fonctions minimisant le risque empirique dans chacune de ces classes.

La Minimisation du Risque Structurel (MRS)

Elle consiste à choisir la classe F_i (et la fonction f_i) de sorte à ce qu'une borne supérieure de l'erreur de généralisation puisse être minimisée.

Une manière de contrôler la complexité d'une classe de fonctions est donnée par la théorie de Vapnik-Chervonenkis (VC) et le principe de MRS. Ici, le concept de compléxité de la fonction de décision f s'exprime par la dimension VC (notée h) de la classe de fonctions F à laquelle appartient f. Grossièrement, la dimension VC mesure combien d'échantillons de l'ensemble d'apprentissage peuvent être séparés par toutes les classifications possibles issues des fonctions de la classe. Elle mesure la capacité du classifieur.

En appelant $R_{emp}(f_i, S_l)$ le minimum du risque empirique sur F_i, on peut deduire de (2.15). Cette dernière borne $R(f_i, S_l)$ est appelée risque structurel et dépend de la classe F_i. Suivant ce principe, il est possible, par rapport à un tirage de S_l, de définir les performances d'un modèle par rapport à un autre en comparant leurs bornes. L'inégalité (2.15) fait apparaître deux cas extrêmes :

- une très petite classe de fonctions (par exemple F_1) fait décroître rapidement le terme de complexité (celui sous la racine carrée), mais le risque empirique demeure grand.
- une très grande classe de fonctions (par exemple F_k) implique un risque empirique petit, mais le terme de compléxité explose.

Comme on peut obtenir aisement $R_{emp}(f_N, S_l) \leq ... \leq R_{emp}(f_1, S_l)$

La meilleure classe de fonctions est généralement intermédiaire entre la plus petite et la plus grande, puisque l'on cherche une fonction qui explique au mieux les données tout en préservant un faible risque empirique (voir figure 2.3).

En définitif, le principe de minimisation du risque structurel consiste en la recherche d'une fonction dont le risque empirique s'approche du risque, par optimisation d'un critère qui intègre également la capacité de l'ensemble des fonctions candidates, définies au sein d'une séquence emboîtée de classes. Toutefois, nous pouvons dire qu'un risque structurel faible induit une faible erreur de généralisation malgré que les problèmes réels auxquels sont confrontées les machines ne constituent pas

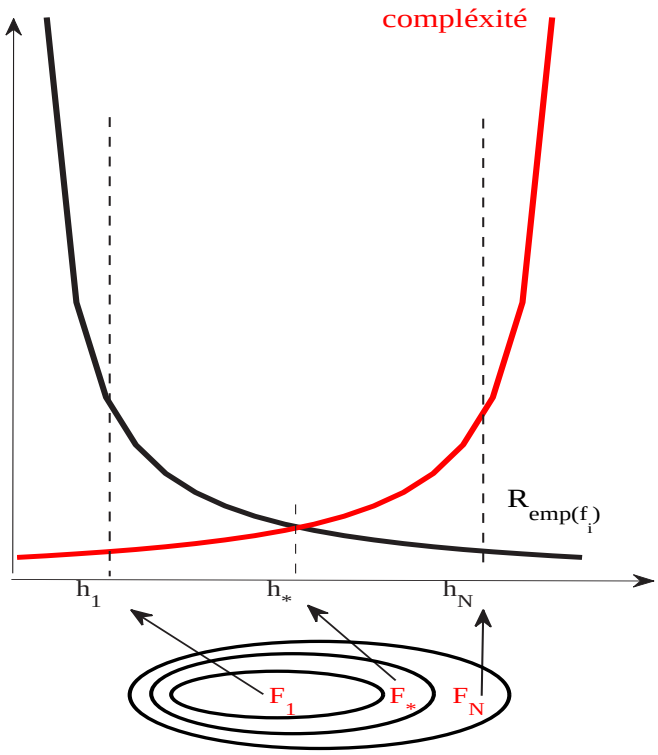


FIG. 2.3 – *Illustration de l'inégalité (2.15). La courbe croissante, appelée confiance, correspond à la borne supérieure du terme de complexité. Les comportements du terme de complexité et de l'erreur empirique sont clairement opposés. On recherche donc le meilleur compromis entre complexité et erreur empirique.*

l'ensemble de tous les problèmes pouvant exister. Néanmoins, il est remarquable de constater qu'un tel principe est valable dans la pratique.

2.5.2 Hyperplans séparateurs

Quand on est dans un espace de représentation euclidien, on peut librement faire des hypothèses sur la géométrie des classes ou sur celles de leurs surfaces séparatrices ; ceci permet de mettre au point des techniques d'apprentissage non statistiquement fondées a priori, mais peut être plus faciles à appliquer. La plus simple d'entre elles est de supposer que deux classes peuvent être séparées par une certaine surface, définie par une équation ; les paramètres qui régissent cette équation sont alors les variables à apprendre. Cette hypothèse n'est pas au fond

tellement plus forte que de supposer que les classes sont de nature gaussienne, elle est souvent bien sûr fausse, mais peut mener à une probabilité d'erreur raisonnable dans un bon nombre de problèmes. Le nombre de paramètres à calculer est minimal si l'on suppose cette surface linéaire ; aussi est-ce l'hypothèse qui prévaut dans la plupart des cas, d'autant qu'elle permet de mener des calculs faciles et de visualiser précisément le résultat obtenu.

Dans $\mathbb{R}^d$, une surface linéaire est un hyperplan $\mathcal{A}$, définie l'équation :

$$a_0 + a^T x = 0 \tag{2.16}$$

avec a vecteur de dimension d et a_0 scalaire. Si deux classes $w_1 et w_2$ sont séparables par $\mathcal{A}$, tous les points de la première classe sont par exemple tels que :

$$x \in w_1 \quad \Rightarrow \quad a_0 + a^T x \geq 0 \tag{2.17}$$

et ceux de la seconde vérifient alors :

$$x \in w_2 \quad \Rightarrow \quad a_0 + a^T x \leq 0 \tag{2.18}$$

Cas particulier Dans un espace de dimension $d = 1$, une séparation linéaire se réduit à la comparaison à un seuil. Prenons ce cas particulier pour donner un exemple où un problème de discrimination à deux classes ne peut pas en pratique être complètement résolu par une séparatrice linéaire.

Exemple Cherchons à classer les hommes et les femmes sur la seule mesure de leur taille. Bien que les hommes soient en général plus grands que les femmes, aucun échantillon d'apprentissage représentatif ne permettra de fixer une valeur pertinente pour le seuil, étant donné qu'il existera évidemment toujours certains hommes plus petits que certaines femmes. Dans ce cas, c'est l'espace de représentation qui est trop pauvre : d'autres critères doivent être mis en oeuvre pour décider sans erreur du sexe d'un individu. Il n'est en général pas possible d'en trouver un. On se contentera donc de chercher un hyperplan discriminant qui en sera une approximation, au sens d'un critère à fixer.

Le problème de l'apprentissage de surfaces linéaires n'est autre que la recherche des paramètres a *et* a_0 séparant le mieux possible les points d'apprentissage de w_1 et de w_2 dans l'espace $\mathbb{R}^d$ et pourvues de la meilleure faculté de généralisation possible.

Definition 2.5.5 ***(Hyperplan séparateur)*** *On appelle hyperplan séparateur ou séparatrice linéaire un hyperplan qui sépare parfaitement les deux classes, c'est-à-dire qui vérifie les équations (2.17) et (2.18) ; en particulier, il sépare parfaitement leurs points d'apprentissage.*

Calcul du biais b :

Le parametre de biais b permet de definir des surfaces de separation ne passant pas par l'origine. Son calcul exploite les vecteurs de support respectant l'inegalite $0 \leq \alpha < C$ dont les ξ_i correspondants sont nuls. L'égalité suivante est alors vérifiée :

$$y_i(b + \sum_{j=1}^{l} y_i \alpha_j k(x_i, x_j)) \tag{2.19}$$

2.5.3 Un peu de géométrie dans $\mathbb{R}^d$

Un hyperplan dans $\mathbb{R}^d$, où les coordonnées sont notés $x_1, ..., x_d$ est définie par $d+1$ paramètres $a_0, a_1, ..., a_d$. Il répond à l'équation :

$$a_0 + a_1 x_1 + ... + a_d x_d = 0 \tag{2.20}$$

En notant vectoriellement $\mathbf{a}^T = (a_1, ..., a_d)$ *et* $\mathbf{x}^T = (x_1, ..., x_d)$, l'équation 2.20 s'écrit de manière équivalente :

$$a_0 + \mathbf{a}^T \mathbf{x} = 0.$$

Un hyperplan d'équation $a_0 + \mathbf{a}^T \mathbf{x} = 0.$ à pour vecteur normal $\mathbf{a}$. Sa distance algébrique à l'origine vaut

$$\frac{a_0}{\|\mathbf{a}\|} \quad avec \quad \|\mathbf{a}\|^2 = \sum_{i=1}^{d} a_i^2$$

La figure (2.4) montre un hyperplan à deux dimensions qui est la droite d'équation $g(x) = a_0 + a_1 x_1 + a_2 x_2 = 0$, ainsi que son vecteur normal $\mathbf{a}$. Cette figure (2.4) donne aussi la distance de deux points particuliers à cet hyperplan : l'origine, à la distance $\frac{a_0}{\|\mathbf{a}\|}$, et un point y de l'autre côté de la droite, à la distance $\frac{g(\mathbf{y})}{\|\mathbf{a}\|}$ Les signes de ces deux distances algébriques sont opposés, puisque les deux points ne sont pas du même côté de l'hyperplan. Le signe de la distance de l'origine à l'hyperplan est le même que celui de a_0. Pour caractériser l'hyperplan d'équation $a_0 + \mathbf{a}^T \mathbf{x} = 0$, on peut rassembler tous ses paramètres dans un vecteur $\mathbf{A}$ de dimension $d+1$ construit comme :

$$\mathbf{A}^T = (a_0, \mathbf{a}^T) = (a_0, a_1, ..., a_d)$$

Par extension, on note $\mathbf{A}$ l'hyperplan lui-même. Notons qu'en réalité il suffit de d paramètres pour définir un hyperplan : on peut sans perdre de généralité imposer par exemple $a_0 = 1$.
Posons

$$g(x) = a_0 + \mathbf{a}^T \mathbf{x}$$

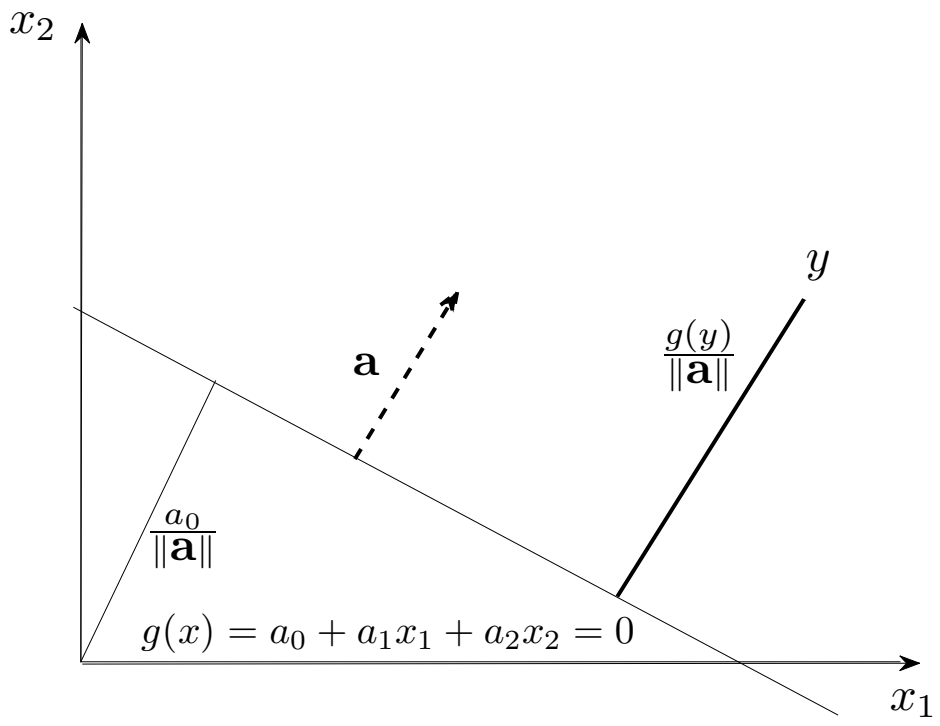


FIG. 2.4 – *La géométrie d'un hyperplan*

et notons $\Delta(\mathbf{y}, \mathbf{A})$ la distance entre l'hyperplan $\mathbf{A}$ d'équation $g(x) = 0$ et un point $\mathbf{y}$ *de* $\mathbb{R}^d$. Elle est définie comme la distance entre $\mathbf{y}$ et le point de $\mathbf{A}$ le plus proche de y. Ce point n'est autre que la projection orthogonale de $\mathbf{y}$ sur $\mathbf{A}$.
La valeur $\Delta(y, \mathbf{A})$ est, comme on l'a vu, une distance algébrique donnée par la formule :

$$\Delta(y, \mathbf{A}) = \frac{g(y)}{\|\mathbf{a}\|}$$

2.5.4 Hyperplan caractérisé par un vecteur dans $\mathbb{R}^d + 1$

2.5.4.1 Une solution globale

Supposons pour le moment que les deux classes w_1 *et* w_2 soient parfaitement séparables par un hyperplan. Il existe donc un vecteur $\mathbf{a}$ caractérisant cet hyperplan séparateur tel que :

- $\mathbf{a}^T x$ positif pour $x \in w_1$

• $\mathbf{a}^T x$ négatif pour $x \in w_2$

Notons qu'il est facile de se ramener à ce cas quand c'est le contraire qui se produit, en inversant simplement le signe de toutes les coordonnées de $\mathbf{a}$. La formalisation de la séparation linéaire pour un problème à deux classes se fait alors en plongeant le problème dans l'espace de dimension $d+1$.

Le vecteur $\mathbf{a}$ et le scalaire a_0 sont transformés en un vecteur $\mathbf{A}$ en ajoutant a_0 à $\mathbf{a}$ comme une coordonnée de rang 0, ce qui se note :$\mathbf{A}^T = (a_0, \mathbf{a}^T)$. On impose une valeur non nulle à a_0, par exemple $a_0 = 1$.

Soit $\mathcal{S}$ l'ensemble des données d'apprentissage : il est composé de m exemples $(\mathbf{x}, w)$ où $\mathbf{x}$ est un vecteur de $\mathbb{R}^d$ et $w \in w_1, w_2$. Au besoin, on notera x_j le j^{me} point de $\mathbf{A}$. On transforme ces exemples en vecteurs X de $\mathbb{R}^d + 1$ par :

$$\mathbf{x} \in w_1 \Rightarrow \mathrm{X}^T = (1, \mathbf{x}^T) \tag{2.21}$$

$$\mathbf{x} \in w_2 \Rightarrow \mathrm{X}^T = (-1, -\mathbf{x}^T) \tag{2.22}$$

Par conséquent, quelle que soit la classe de $\mathbf{x}$, on a désormais : $\mathbf{A}^T \mathrm{X} = 0$ (rappelons que l'hyperplan caractérisé par le vecteur $\mathbf{A}$ est pour le moment supposé séparateur : tous les points sont bien classés). Si on range maintenant les m vecteurs X comme les colonnes d'une matrice M, il est alors facile de voir le problème de séparation linéaire comme la recherche d'un vecteur $\mathbf{A}$ dans $\mathbb{R}^d + 1$

$$\mathbf{A}^T M = \mathbf{B}^T \tag{2.23}$$

où $\mathbf{B}$ est un vecteur positif de $\mathbb{R}^m$ (dont toutes les coordonnées sont positives). Ceci revient à dire que l'on recherche un hyperplan dans l'espace de représentation augmenté d'une dimension, avec les contraintes qu'il passe par l'origine et que $\mathbf{B}$ reste positif. Cette manipulation mathématique n'a d'autre but que de rendre plus aisée la résolution du problème. Cette équation ne peut pas se résoudre dans le cas général, puisque M n'est pas inversible (c'est une matrice à $d+1$ lignes et m colonnes) et que $\mathbf{B}$ est inconnu. De plus, il faut maintenant traiter le cas général où les classes et les données d'apprentissage ne sont pas séparables. On se ramène donc à la recherche d'une séparatrice linéaire en cherchant un hyperplan $\mathbf{A}$ le meilleur possible pour distinguer les deux classes. Par exemple, celui qui minimise le critère :

$$J(\mathbf{A}, \mathbf{B}) = \frac{1}{2} \|\mathbf{A}^T M - \mathbf{B}^T\|^2 \tag{2.24}$$

Pourquoi choisir cette valeur ? Si les deux classes sont séparables, il existera un couple $(\mathbf{A}, \mathbf{B})$ avec $\mathbf{B}$ positif pour lequel ce critère sera exactement nul. En imposant dans le cas général la contrainte $\mathbf{B}$ positif ou nul, on peut donc comprendre intuitivement qu'une bonne solution approchée pour $\mathbf{A}$ rendra $J(\mathbf{A}, \mathbf{B})$ assez petit. Donnons une justification plus profonde à cet argument. $J(\mathbf{A}, \mathbf{B})$ peut s'écrire :

$$J(\mathbf{A}, \mathbf{B}) = \frac{1}{2} \sum_{j=1}^{m} [\mathbf{A}^T \mathrm{X}_j - b_j]^2 \tag{2.25}$$

X_i étant la projection dans l'espace de dimension $d+1$ du j^{me} point d'apprentissage x_j . On sait que la distance $\Delta(x_j, \mathbf{A})$ du point x_j à l'hyperplan caractéerisé par le vecteur A a pour valeur :

$$\Delta(x_j, \mathbf{A}) = \frac{a^T x_j + a_0}{\|a\|} = \frac{\mathbf{A}^T \mathrm{X}_j}{\|a\|}$$

Son carré peut donc s'écrire :

$$[\Delta(x_j, \mathbf{A})]^2 = \frac{(\mathbf{A}^T \mathrm{X}_j)^2}{\|a\|^2}$$

On en déduit :

$$J(\mathbf{A}, \mathbf{B}) = \frac{1}{2\|a\|^2} \sum_{j=1}^{m} [\Delta(x_j, \mathbf{A}) - b_i \|a\|]^2$$

Pour les points de A bien classés par l'hyperplan $\mathbf{A}$, le terme

$$\sum_{j=1}^{m} [\Delta(x_j, \mathbf{A}) - b_i \|a\|]^2$$

peut être annulé en choisissant $b_j = \frac{[\Delta(x_j, \mathbf{A})}{\|a\|}$ qui est positif. Pour les points mal classés, à défaut de pouvoir prendre b_j négatif, le mieux que l'on puisse faire est d'imposer b_j nul. Minimiser $J(\mathbf{A}, \mathbf{B})$ sous la contrainte $\mathbf{B}$ positif ou nul revient donc d'une certaine manière à se placer dans le cadre du critère des moindres carrés : on cherche l'hyperplan $\mathbf{A}$ qui minimise la somme des distances entre $\mathbf{A}$ et les points de l'ensemble d'apprentissage que $\mathbf{A}$ classe mal. On a, si $\mathbf{A}$ et $\mathbf{B}$ minimisent effectivement $J(\mathbf{A}, \mathbf{B})$:

$$J(\mathbf{A}, \mathbf{B}) = \sum_{x \in \mathcal{S} \ mal \ classe' \ par \ \mathbf{A}} [\Delta(x_j, \mathbf{A}) - b_i]^2$$

Par conséquent, sous l'hypothèse que les distances entre $\mathbf{A}$ et les points mal classées par $\mathbf{A}$ sont réparties selon une distribution gaussienne, et que ceux-ci sont les seuls à compter dans le positionnement de $\mathbf{A}$, la minimisation de $J(\mathbf{A}, \mathbf{B})$ est la recherche du meilleur hyperplan a posteriori (au sens bayésien), c'est-à-dire le plus probable connaissant les données.

2.6 Classification et apprentissage automatique

Identifier les classes auxquelles appartiennent des objets à partir de leur description.

- $\mathcal{X}$ une population
- $\mathcal{D}$ l'ensemble des descriptions
- $\mathcal{C}$ l'ensemble des classes

On recherche une fonction f qui associe à un individu de $\mathcal{X}$ à partir de sa description dans $\mathcal{D}$, une classe de $\mathcal{C}$. L'apprentissage automatique (*Machine learning*) est une discipline basée sur l'analyse et l'implémentation de méthodes qui permettent à une machine d'effectuer des tâches qui requierent de l'intelligence et ainsi de remplir des tâches impossible de remplir par des algorithmes classiques. On cite parmi ces tâches :

- l'aide à la décision
- la reconnaissance de formes
- la prédiction ...

L'apprentissage automatique est un domaine dont l'intérêt majeur est le développement des algorithmes permettant à une machine d'apprendre a partir d'un ensemble de données. La motivation originale de ce domaine était de mettre en oeuvre des systèmes artificiels intelligents. Les algorithmes issus de ce domaine sont utilisés par plusieurs autres domaines, tels que la vision par ordinateur, la reconnaissance de forme, la recherche d'information, la bioinformatique, la fouille de données et beaucoup d'autres. Un algorithme d'apprentissage peut être défini comme suit : C'est un algorithme qui prend en entrée un ensemble de données qui contient les informations nécessaires pour caractériser un problème donné et qui retourne un modèle f qui représente des concepts caractérisant ces données. Nous appelons un ensemble d'apprentissage ou ensemble d'entraînement et chacun de ses éléments, représentant un exemple d'apprentissage. Il existe plusieurs types d'apprentissage automatique qui se distinguent essentiellement par leur objectif. Les deux types les plus connus sont : l'apprentissage supervisé et l'apprentissage non supervisé. Si l'on dispose d'un ensemble de points étiquetés, nous parlerons de classification supervisée. Dans le cas contraire, nous efféctuons une classification non supervisée. L'objectif, dans le cas d'apprentissage supervisé, est déterminé explicitement par la prédiction d'une cible. Dans ce cas, pour chaque exemple d'apprentissage x_i de $\mathcal{X}$, une cible (étiquette) y_i est associée. La tâche d'un algorithme d'apprentissage est alors d'entraîner un modèle qui puisse prédire, pour une entrée x quelconque, la valeur de la cible y. La nature de l'ensemble de cibles (noté Y) dépendra du type de problèmes à résoudre. Deux types de problèmes fréquents sont les problèmes de classification et de régression. Pour un problème de classification, Y correspond

à un ensemble fini de classes aux quelles peuvent appartenir les différents x_i. Par contre pour celui de la régression, Y correspond à un ensemble de valeurs continues. Dans le cadre de cette thèse, nous parlerons surtout de l'apprentissage supervisé pour des problèfmes de classification. Dans la suite, nous présentons les principaux algorithmes de classification supervisée proposés dans la littérature. Il ne s'agit pas de faire une présentation exhaustive de toutes les méthodes. Les algorithmes d'apprentissage peuvent être catégorisés selon le mode d'apprentissage qu'ils emploient.

Aprentissage supervisé :

A partir d'un ensemble d'observations $x_1, .., x_n \in \mathcal{X}^p$ et de mesures (labels, étiquettes) $y_i \in \mathcal{Y}$, on cherche à estimer la dépendence entre l'ensemble $\mathcal{X}$ et $\mathcal{Y}$.

- On parle d'apprentissage supervisé car les y_i permettent de guider le processus d'estimation.

Exemple de donnée : Estimer les liens entre la morphologie des battements cardiaques (forme, largeur,..) et leur nature.

- x_i est un battement cardiaque (vecteur caractérisant le battement)
- y_i une cathégorie ($y_i = 1$, battement systolique,$y_i = -1$, battement ectopique).

Exemples d'algorithmes :

- Méthode des K plus proches voisins
- Arbres de décision
- Approche Bayesienne
- Réseaux de neurones
- Séparateurs à Vastes Marges (SVMs)

Aprentissage non supervisé :

Comme seules les observations $x_1, .., x_n \in \mathbb{R}^p$ sont disponibles, l'objectif est de decrire comment les données sont organisées et d'en extraire des sous-ensemble homogènes. L'algorithme doit découvrir par lui-même la structure des données. (exemples : classer les données en 2 ou plus de classes où chaque classe contiendra les exemples similaires). exemple d'algorithmes : Le *clustering* est un algorithme d'apprentissage non supervisé.

Aprentissage semi supervisé :

Parmi les observations $x_1, .., x_n \in \mathbb{R}^p$, seulement un petit nombre d'entre elles ont un label y_i. L'objectif est le même que pour l'apprentissage supervisé.

2.7 Différentes méthodes de classification

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains traits descriptifs. Elles s'appliquent à un grand nombre d'activit és humaines et conviennent en particulier au problème de la prise de décision automatisé. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples. Un exemple consiste en la description d'un cas avec la classification correspondante. Un système d'apprentissage doit alors,à partir de cet ensemble d'exemples, extraire une procédure de classification, il s'agit en effét d'extraire une régle générale à partir des données observées. La procédure générée devra classifier correctement les exemples de l'échantillon et avoir un bon pouvoir prédictif pour classifier correctement de nouvelles descriptions. Dans ce chapitre, une définition du concept d'apprentissage automatique est donnée.

2.7.1 Méthode des k-plus proches voisins

La méthode des k plus proches voisins (*Knn k-nearest neighbor*) [Indyk 1998] se base sur une comparaison directe entre le vecteur caractéristique représentant l'entité à classer et les vecteurs caractéristiques représentant des entités de référence. La comparaison consiste en un calcul de distances entre ces entités. L'entité à classer est assignée à la classe majoritaire parmi les classes des k entités les plus proches au sens de la distance utilisée. Notons par $X_p = (x_{p_1}, x_{p_2}, .., x_{p_N})$ le vecteur caractéristique de l'entité p, avec N le nombre de caractéristiques et par p et q deux entités à comparer. Les distances suivantes sont usuellement employées par les classificateurs Knn : Distance Euclidienne :

$$D(x_p, x_q) = \sqrt{\sum_{i=1}^{N} (x_{p_i} - x_{q_i})^2}$$

2.7.2 Arbres de décision

Le formalisme des arbres de décision permet de classifier un nouvel objet en testant ses caractéristiques les unes à la suite des autres. La classification se fait à travers une séquence de questions dans laquelle chaque question dépend de la réponse à la question précédente. Cette séquence de questions est représentée par un arbre de décision dont les feuilles terminales représentent les classes. Dans la phase de construction de ce classifcateur, les exemples de l'ensemble d'apprentissage sont divisés récursivement par des tests définis sur les caractéristiques pour obtenir des sous-ensembles d'exemples ne contenant que des exemples appartenant tous à une même classe. Les algorithmes existants [Breiman 1984, Quinlan 1986, Quinlan 1993]

différent essentiellement par leur façon de choisir, à une étape donnée, et parmi les caractéristiques disponibles, la caractéristique de segmentation ansi que le critère d'arrêt.

2.7.3 Approche Bayesienne

Un classificateur bayésien [Jensen 2001, Neapolitan 2003] est basé sur une approche probabiliste employant la régle de Bayes. Notons $P(C_i)$ la probabilité a priori d'une classe C_i, $P(x)$ la probabilité d'observer un vecteur caractéristique x et $P(x_j/C_i)$ la probabilité d'observer le vecteur x sachant que la classe est C_i. La régle de Bayes permet alors de calculer la probabilité a posteriori de la classe C_i quand x est observé :

$$P(C_i/x) = \frac{P(x/C_i)P(C_i)}{\sum_j P(x/C_j)P(C_j)} \tag{2.26}$$

Dans la pratique, puisque le dénominateur de la formule de Bayes ne dépend pas de C_i, nous ne nous intéressons qu'au numérateur. Les probabilités $P(C_i)$ de chaque classe ainsi que les distributions $P(x_j/C_i)$ doivent être préalablement estimées à partir d'un échantillon d'apprentissage. Le vecteur x est assigné à la classe C_i si :

$$\forall i \neq j, \qquad P(C_i/x) > P(C_j/x \tag{2.27}$$

L'analyse discriminante se présente comme un cas particulier de l'approche Bayésienne. Dans ce cas, les données d'apprentissage sont modèlisées par des distributions Gaussiennes. Sur la base des paramétres estimés, des fonctions discriminantes sont construites permettant de classer tout vecteur de caractèristiques.

La régression logistique

Elle admet une hypothèse differente portant sur les probabilités a posteriori. Introduite initialement pour résoudre des problèmes de classification binaire, cette méthode modélise le log du rapport des probabilités a posteriori par des modéles linéaires :

$$log\frac{P(C_1)}{P(C_2)} = a_0 + a_1x_1 + ... + a_dx_d \tag{2.28}$$

et comme la somme des deux probabilités a posteriori vaut 1 , il en résulte que ces probabilités sont modélisées par des lois logistiques :

$$P(C_1/x)) = \frac{exp^{a_0+a_1x_1+...+a_dx_d}}{1 + exp^{a_0+a_1x_1+...+a_dx_d}} \tag{2.29}$$

$$P(C_2/x)) = \frac{1}{1 + exp^{a_0+a_1x_1+...+a_dx_d}} \tag{2.30}$$

2.7.4 Réseaux de neurones

Les réseaux de neurones sont inspirés de la structure neurophysiologique du cerveau. Un neurone formel est l'unité élémentaire d'un système modélisé par un réseau de neurones artificiels. A la réception de signaux provenant d'autres neurones du réseau, un neurone formel réagit en produisant un signal de sortie qui sera transmis à d'autres neurones du réseau. Le signal reçu est une somme pondérée des signaux provenant de diffirents neurones. Le signal de sortie est une fonction de cette somme pondérée :

$$y_j = f(\sum_{i=1}^{N} w_{ij} x_i) \tag{2.31}$$

avec y_j la sortie du neurone formel j en question, $x_i, (i = 1, .., N)$ les signaux reçus par le neurone j de la part des neurones i, et w_{ij} les poids des interconnexions entre les neurones i et j. Selon l'application, la fonction f, appelée fonction d'activation, est le plus souvent une fonction identité, sigmoide, tangente hyperbolique ou une fonction linéaire par morceaux. En classifiation, les réseaux de neurones [Dietz 1989, Broadbent 1988] permettent d'introduire de la non-linéarité dans la séparation entre les classes grâce au choix de la fonction d'activation. Les neurones sont ainsi organisés en trois couches ou plus : les cellules d'entrée associées aux données, les neurones de sortie associés chacun à une classe, et les neurones cachés qui sont entre les neurones d'entrée et les neurones de sortie. L'apprentissage du classificateur consiste à faire évoluer les poids w_{ij} par des méthodes d'optimisation non linéaires pour minimiser une fonction de coût qui constitue une mesure de l'écart entre les réponses réelles du réseau et les réponses désirées.

La théorie de l'apprentissage. Celle-ci s'est lentement précisée, en particulier sous l'influence de [Vapnik 1963]. Selon sa formulation, apprendre c'est approcher avec une fonction simple. Cette idée possède un certain degré de justification théorique : en approchant avec une fonction simple on borne supérieurement la probabilité d'erreur. Mais surtout, elle permet d'associer à une question d'apprentissage un problème d'optimisation quadratique convexe sous contraintes linéaires, donc très bien connu.

2.7.4.1 Classification binaire

Le problème de la classification binaire est en un sens le plus simple des problèmes d'apprentissage : on considère des données d'apprentissage (x_i, y_i), telles que $y_i = \pm 1$.

Definition 2.7.1 *On dit que l'ensemble de couples* $N = (x_i, y_i) avec i = 1, ..., n$

où $x_i \in R^d$ *et* $y_i = \pm 1$ *est un nuage binaire.*

▷ *On note* N^+ *l'ensemble des* x_i *pour lesquels* $y_i = +1$

▷ N^- *l'ensemble des* x_i *pour lesquels* $y_i = -1$

On pose le problème de l'apprentissage de la manière suivante : on cherche une fonction affine $f(x) =< w, x > +b$ telle que les points bleus soient d'un coté de la courbe de niveau 0 de f et les points rouges de l'autre (voir figure 43). On cherche f dans l'idée que si un nouveau point x se présente, il suffira de regarder le signe de $f(x)$ pour décider si le y non-observé associé à x est 1 ou -1 :

$$y = \varphi(x) = sgn(< w, x > +b) \tag{2.32}$$

où sgn() désigne la fonction signe. Il est facile d'imaginer deux nuages de points tels qu'aucune droite ne sépare les rouges des bleus. On dit dans ce cas que les données sont non-séparables.

Definition 2.7.2 *On dit qu'un nuage binaire N est séparable s'il existe un hyperplan H séparant* N^+ *et* N^-.

Nous nous limitons pour l'instant au cas séparable. La deuxième remarque est que les droites qui séparent les points rouges et bleus ne sont pas équivalentes du point de vue de la qualité de prédiction qui leur est associée. On a représenté figure 3 un ruban qui sépare les points bleus et rouges.

2.7.5 Séparateurs à Vaste Marge (SVMs)

Definition 2.7.3 *Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais Support Vector Machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression.*

Les SVM sont une généralisation des classifieurs linéaires. Les SVM ont été développés dans les années 1990 à partir des considérations théoriques de Vladimir Vapnik sur le développement d'une théorie statistique de l'apprentissage : la Théorie de Vapnik-Chervonenkis.[Vapnik 1998]

Les SVM ont rapidement été adoptés pour leur capacité à travailler avec des données de grandes dimensions, le faible nombre d'hyper paramètres, leurs garanties théoriques, et leurs bons résultats en pratique.

Les SVM ont été appliqués à de très nombreux domaines (bio-informatique, recherche d'information, vision par ordinateur, finance1...). Selon les données, la performance des machines à vecteurs de support est de même ordre, ou même supérieure, à celle d'un réseau de neurones ou d'un modèle de mixture gaussienne. Dans le cas de données linéairement séparables, il n'est pas possible de déterminer de manière unique un hyperplan séparateur en se basant sur le seul critère de minimisation du nombre d'observations mal classées. En 1998, V. Vapnik (Vapnik [1998]) a défini un critère d'optimalité basé sur la "marge" pour séparer des classes linéairement séparables et l'a généralisé à des frontières non linéaires grâce à un changement d'espace. Nous nous intéressons ici à la classification binaire c'est-à-dire à l'approximation de fonctions de décision permettant de distinguer entre deux classes, et ceci par des hyperplans $h : \mathcal{X} \longmapsto \mathcal{Y}$, d'équation

$$h(x) = \langle w, x \rangle + \omega_0 = \sum_{i=0}^{d} \omega_i x_i = 0 \tag{2.33}$$

en supposant que X soit de dimension d (par exemple, $X \in \mathbb{R}^d$). On utilisera alors la fonction de décision

$$g(x) = signe(\langle w, x \rangle + w_0) \tag{2.34}$$

pour prédire la classe ($+ ou -$) de l'entrée x. Selon le principe de minimisation du risque empirique classique, un tel " meilleur " hyperplan sera celui qui minimise le nombre d'exemples mal classés par cette fonction de décision dans l'échantillon d'apprentissage $S = \langle (x_1, u_1), ..., (x_m, u_m) \rangle$. La distance d'un point x' à l'hyperplan d'équation

$$h(x) = w^T x + \omega_0 \tag{2.35}$$

est égale à :

$$\langle w, x' \rangle \tag{2.36}$$

puisque w est un vecteur orthogonal à $h(x)$. Afin de pouvoir comparer plusieurs hyperplans séparateurs de vecteurs directeurs w différents, on normalise la distance en la divisant par la norme du vecteur

$$w : \langle w, x/\|w\| \rangle \tag{2.37}$$

Lorsqu'il existe une séparatrice linéaire entre les points d'apprentissage, il en existe en général une infinité. On peut alors chercher parmi ces séparatrices celle qui sépare " au mieux " les deux nuages de points . Cet hyperplan optimal dont la distance minimale aux exemples d'apprentissage est la plus grande possible. Dans ce cas, la marge normalisée, appelée aussi marge géométrique vaut :

$$2/\|w\|. \tag{2.38}$$

Chercher un hyperplan séparateur de marge maximale présente plusieurs avantages. Dabord, même si les arguments théoriques sont délicats, il est montré que la capacité de l'espace d'hypothèses constitué par les hyperplans séparateurs diminue lorsque la marge augmente. Ainsi, on peut espérer que l'hyperplan de marge maximale séparant les exemples des deux classes est l'hypothèse satisfaisant au mieux un risque empirique régularisé, favorisant donc la performance en généralisation. Ensuite, il existe un unique hyperplan séparateur de marge maximale et, le problème étant convexe, il peut être résolu par des méthodes de programmation quadratique, donc de manière efficace. L'hyperplan séparateur a pour équation :

$$w^T x + \omega_0 = 0 \tag{2.39}$$

avec x un vecteur caractéristique et w un vecteur de coéficients. Maximiser la marge revient à maximiser la plus petite distance séparant un point de l'espace des observations à l'hyperplan séparateur, ce qui peut s'écrire sous la forme du problème d'optimisation suivant : (formulation primale du problème) La recherche de l'hyperplan optimal revient donc à résoudre le problème d'optimisation suivant qui porte sur les paramètres

$$Minimiser \quad \frac{1}{2}\|w\|^2 \quad Sc \quad u_i(w^T x_i + \omega_0) \geq 1 \quad avec \quad i = 1, ..., m \tag{2.40}$$

avec x_i est le vecteur caractéristique de la i^{me} observation et u_i est la classe correspondante. Notons que $u_i = 1$ ou $u_i = -1$ (cas de classification binaire). Ce problème admet une solution unique qui ne dépend que des points situés sur la marge (les points supports), donc les plus dificiles à classer, d'où la robustesse de cette méthode de classification. La fonction de décision des séparateurs à vastes marges (SVM) s'écrit ainsi :

$$f(x) = \omega_0 + \sum_{Support} \alpha_i y_i x_i \tag{2.41}$$

avec α_i les multiplicateurs de Lagrange. Dans le cas de données non séparables linéairement dans leur espace d'origine, un changement d'espace, généralement de dimension plus grande, peut les rendre séparables. A une frontière linéaire dans l'espace transformé, correspond une frontière non linéaire dans l'espace d'origine. L'hyperplan optimal dans l'espace transformé s'écrit :

$$f(x) = \omega_0 + \sum_{Support} \alpha_i y_i < \phi(x_i), \phi(x) > \tag{2.42}$$

avec $\phi(x)$ la transformé de x dans le nouvel espace. Pour éviter de calculer explicitement les transformés des points dans le nouvel espace, on choisit une transformation qui permet le calcul du produit scalaire $< \phi(x_i), \phi(x) >$ en fonction de x_i et de

$$< \phi(x_i), \phi(x) > = \ K(x_i, x). \tag{2.43}$$

K est appelé fonction noyau (ou kernel). C'est ce que l'on appelle l'" astuce des noyaux " (kernel trick). La nouvelle formulation implique, d'une part, de trouver une fonction de redescription Φ adéquate, et, d'autre part, de calculer des produits scalaires dans un espace $\mathcal{F}$ dont la dimension peut être grande. Cependant, il peut parfois être possible de ne pas avoir à effectuer explicitement ces produits scalaires dans $\mathcal{F}$ grâce à lutilisation de fonctions noyau.

2.7.6 Fonctions noyaux

Les méthodes à noyaux permettent de trouver des fonctions de décision non linéaires, tout en s'appuyant fondamentalement sur des méthodes linéaires. Une fonction noyau correspond à un produit scalaire dans un espace de redescription des données, souvent de grande dimension. Grâce à l'utilisation de fonctions noyau, il devient ainsi possible d'utiliser des techniques simples et rigoureusement garanties, et traiter des problèmes non linéaires. Les méthodes d'apprentissage dites à noyaux [1] sont actuellement très en vogue. Ils ont introduit la notion d'espace de redescription [2]. Supposons que l'on cherche à résoudre cette fois ci un problème non linéaire dans X on peut être tenté d'utiliser pour ce faire une fonction de redescription ϕ de X dans un nouvel espace $\mathcal{F}$, de telle manière que le problème devienne un problème linéaire dans ce nouvel espace.[Boser 1992] Dans ce cas, la solution du problème linéaire, dans l'espace de redescription $\mathcal{F}$, prendra la forme primale

$$h(x) = \langle w, \phi(x) \rangle = 0,$$

que l'on peut ré-exprimer en mettant en évidence la base de fonctions non linéaires

$$\phi_1(x), \phi_2(x), ...,$$

éventuellement infinie, de l'espace $\mathcal{F}$:

$$h(x) = \sum_{j=1}^{\infty} \omega_j \phi_j(x) + \omega_0 \tag{2.44}$$

En résolvant, cette fois-ci dans l'espace de redescription, on obtient à nouveau une expression duale :

$$h(x) = \sum_{i=1}^{m} \alpha_i \langle \Phi(x_i), \Phi(x) \rangle \tag{2.45}$$

Et, en rapprochant les équations 2.44 et 2.48, on tire :

$$h(x) = \sum_{j=1}^{\infty} \omega_j \phi_j(x) + \omega_0 \;=\; \Psi^T \Phi(x) \tag{2.46}$$

[1] kernel-based methods

[2] feature space

$$= \sum_{i=1}^{m} \alpha_i \Phi(x_i).\Phi(x) = 0 \tag{2.47}$$

d'où

$$\Psi^T = \sum_{i=1}^{m} \alpha_i \Phi(x_i) \tag{2.48}$$

Definition 2.7.4 *(Fonction noyau (Kernel function))*

Une fonction noyau est une fonction $\kappa : x, x' \in \mathcal{X}^2$ *satisfait :*

$$\kappa(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

où :Φ *est une fonction de* $\mathcal{X}$ *vers un espace de redescription* $\mathcal{F}$ *doté d'un produit scalaire :* $\Phi : x \mapsto \Phi(x) \in \mathcal{F}$

L'utilisation de fonctions noyau permet ainsi de calculer implicitement un produit scalaire dans un espace de dimension éventuellement infini par un calcul n'impliquant qu'un nombre fini de termes, ce nombre étant le nombre m des exemples d'apprentissage.

Definition 2.7.5 *(Espace de Hilbert) Un espace de Hilbert est un espace vectoriel sur* $\mathbb{R}$ *ou* $\mathcal{C}$*, muni d'un produit scalaire dont l'espace normé associé est complet. Dans le cas des espaces fonctionnels de fonctions numériques continues sur un intervalle* $[a, b]de\mathbb{R}$*, le produit scalaire utilisé est généralement :*

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx$$

la norme associé $\|f\|$ est alors définie par :

$$\|f\|^2 = \int_a^b [f(x)]^2 dx$$

Lorsque la norme est bornée sur $\mathbb{R}$ tout entier, f^2 est intégrable et on parle de fonctions de carrés intégrables.

Espace de Hilbert à noyau reproduisant (**RKHS**) [3] Etant donnée une fonction noyau, il est possible de lui associer un espace de Hilbert de fonctions régulières.

Definition 2.7.6 *(**RKHS**) Soit $\mathcal{F}$ une espace de Hilbert de fonctions réelles définies sur un ensemble indexé $\mathcal{X}$:*

$$\mathcal{F} = \{ \sum_{i=1}^{m} \alpha_i \kappa(x_i, \cdot) : \ m \in \mathcal{N}, \ x_i \in \mathcal{X}, \ \alpha_i \in \mathbb{R}, \ \ i = 1, ..., m \ \ \}$$

$\mathcal{F}$ est un (**RKHS**) doté d'un produit scalaire noté :$\langle ., . \rangle_{\mathcal{F}}$ (et de la norme

$$\|f\|_{\mathcal{F}} \ = \ \sqrt{\langle f, f \rangle}$$

si il existe une fonction $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ ayant les propriétés suivantes :

- Pour tout élément $x \in \mathcal{X}, \kappa(x, \cdot)$ appartient à $\mathcal{F}$,
- La fonction κ est une fonction noyau reproduisante

c'est-à-dire telle que pour toute fonction $f \in F$, on a :

$$\langle f(\cdot), \ \kappa(x, \cdot) \rangle_{\mathcal{F}} \ = \ \langle \sum_{i=1}^{m} \alpha_i \kappa(x_i, \cdot) \kappa(\cdot, x) \ \rangle_{\mathcal{F}} = \ f(x)$$

Le fait que la fonction noyau soit reproduisante signifie que toute fonction $f \in \mathcal{F}$ est égale à un produit scalaire qui est aussi une combinaison linéaire finie de fonctions de base. Le produit scalaire sur $\mathcal{F}$ est alors défini comme suit. Soient les fonctions $f, g \in \mathcal{F}$ définies par :

$$f(x) \ = \ \sum_{i=1}^{m} \alpha_i \kappa(x_i, x), \qquad \text{et} \qquad g(x) \ = \ \sum_{j=1}^{n} \beta_j \kappa(z_j, x)$$

alors :

$$\langle f, g \rangle \ = \ \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i \beta_i \kappa(x_i, x_j) \ = \ \sum_{i=1}^{m} \alpha_i g(x_i) \ = \ \sum_{j=1}^{n} \beta_j f(x_j)$$

Une caractéristique cruciale de ces fonctions analysantes $\kappa(x_i, \cdot)$ est donc qu'elles dépendent directement des points x de l'espace $\mathcal{X}$. Cela est capital dans l'estimation de fonctions à partir de points particuliers, car, dans ce cas, les fonctions analysantes utilisées pour estimer la fonction vont dépendre des points connus $\{x_i\}_{1 \leq i \leq m}$. La fonction estimée prendra donc la forme :

[3] (**RKHS** Reproducing Kernel Hilbert Space)

$$h(x) = \sum_{i=1}^{m} \alpha_i \kappa(x, x_i)$$

Grâce à l'utilisation judicieuse de la propriété de reproduction, le problème de recherche d'une solution générale f dans l'espace de fonctions $\mathcal{F}$ se ramène à la recherche d'un vecteur $\alpha \in \mathbb{R}^n$ où n désigne le nombre d'exemples disponibles pour l'apprentissage. Le lien entre la fonction f et le vecteur α est donné par [Scholkopf 2002]

Quelques noyaux usuels On cite quelques noyaux usuels (Tab.2.4)

TAB. 2.4 – *Quelques noyaux usuels*

Noyau linéaire	$k(x_i, x_j) = x_i^T x_j$
Noyau gaussien	$k(x_i, x_j) = exp(-\frac{\|x-y\|^2}{2\sigma^2})$
Noyau polynomial	$k(x_i, x_j) = (x_i^T x_j + 1)^d$

Similarité L'étape de généralisation des méthodes d'apprentissage consiste à mettre en relation une nouvelle donnée avec une base d'apprentissage par l'intermédiaire d'informations extraites de ces données. Cette prise de décision se base sur une certaine mesure de similarité, la classe d'une nouvelle donnée étant choisie comme étant celle présentant le plus de similarité. Nous venons de voir que l'algorithme des SVM est uniquement basé sur l'utilisation de produits scalaires. Si on considère deux données $x, x' \in \mathbb{R}^N$, leur produit scalaire s'écrit : D'un point de vue géométrique,

$$\langle x_i, x_i' \rangle = \sum_{i=1}^{N} x_i x_i'$$

correspond au cosinus de l'angle entre les vecteurs x, x' pour peu que ces derniers soient normalisés à 1. Cette opération code par conséquent une mesure de similarité entre les données puisque plus elles seront similaires, plus l'angle qu'elles décrivent sera faible et leur produit scalaire important. A l'inverse, si elles tendent à être orthogonales, leur produit scalaire tendra lui vers zéro. Ce produit scalaire permet de plus de doter l'espace considéré d'une métrique définie par :

$$\|x\| = \sqrt{\langle x, x \rangle}$$

Ainsi par l'intermédiaire de ce produit scalaire on peut réaliser toute sorte d'algorithme mettant en oeuvre des opérations relevant du domaine de la géométrie, et en particulier l'algorithme des SVM qui vise à maximiser une distance.

2.8 Conclusion

A travers ce chapitre, nous avons passé en revue le principe de l'apprentissage automatique et ses objectifs. Nous nous sommes ensuite intéréssés aux problèmes de classification qui peuvent être abordés de différentes façons selon différents algorithmes. Parmi ces algorithmes, nous retenons les Séparateurs à Vaste Marge connus pour leur pouvoir de généralisation et leur souplesse d'utilisation. Dans le prochain chapitre , nous allons developper le fondement mathématique du SVM, vu comme un problème d'optimisation quadratique convexe.

Chapitre 3

Séparateurs à Vaste Marge (SVMs)

3.1 Introduction

Les Support Vector Machines souvent traduit par l'appellation de Séparateur à Vaste Marge (SVMs) sont une classe d'algorithmes d'apprentissage initialement définis pour la discrimination de deux classes. L'origine des machines à vecteurs de support (SVMs) remonte à 1975 lorsque Vapnik et Chervonenkis proposèrent le principe du risque structurel et la dimension VC pour caractériser la capacité d'une machine d'apprentissage. A cette époque, ce principe n'a pas trouvé place et il n'existait pas encore un modèle de classification solidement appréhendé pour être utilisable. Il a fallu attendre jusqu'à 1982 pour que Vapnik propose un premier classifieur basé sur la minimisation du risque structurel baptisé SVM [Vapnik 1982]. Ce modèle était toutefois linéaire et on ne connaissait pas encore un moyen d'induire des frontières de décision non linéaires. En 1992, Boser *et al.*[Boser 1992] proposèrent l'introduction des noyaux pour étendre le SVM au cas non linéaire.

Les séparateurs à vastes marges reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau. Ces deux notions existaient depuis plusieurs années avant qu'elles ne soient mises en commun pour construire les SVM. L'idée des hyperplans et marge maximale a été explorée dès 1963 par Vladimir Vapnik [Vapnik 1963] et en 1973 par Richard Duda [Duda 2000]. Les fondations théoriques des SVMs ont été explorées par Vapnik et ses collègues dans les années 70 avec le développement de la Théorie de Vapnik-Chervonenkis [Risi 2008] et par Valiant [Valiant 1984]. L'idée des fonctions noyaux, non plus, n'est pas nouvelle : le théorème de Mercer [Mercer 1909] date de 1909, et l'utilité des fonctions noyaux dans le contexte de l'apprentissage artificiel a été montrée dès 1964 par Aizermann, Bravermann et Rozoener [Aizermann 1964]. Ce n'est toutefois qu'en 1992 que ces idées seront bien comprises et rassemblées par Boser, Guyon et Vapnik [Boser 1992] dans un article, qui est l'article fondateur des séparateurs à vaste marge.

Cortes *et al.*[Cortes 1982] proposèrent une version régularisée des SVMs qui tolèrent des erreurs d'apprentissage tout en les pénalisant. L'introduction des variables ressorts qui permet de résoudre certaines limitations pratiques importantes, ne sera

introduite qu'en 1995. À partir de cette date, qui correspond à la publication du livre de Vapnik [Vapnik 1998], les SVMs gagnent en popularité et sont utilisés dans de nombreuses applications.

3.2 Objectifs d'un SVM

Soit un nuage de points (voir 3.1) de natures différentes (points rouges, points bleus). Lors de l' apprentissage d'un SVM, deux objectif sont poursuivis, séparer les deux classes et les repousser l'une de l'autres pour avoir une marge maximale.

3.2.1 Hyperplan optimal

Le premier objectif recherché est de trouver une frontière de décision (hyperplan séparateur) qui puisse séparer le nuage de points en deux régions en commettant un minimum d'erreurs, c'est à dire, (trouver l'hyperplan optimal). La figure 3.1 montre qu'il existe en effet plusieurs d'hyperplans séparateurs dont les performances en apprentissage sont identiques (le risque empirique est le même), mais dont les performances en généralisation peuvent être très différentes. Pour résoudre ce problème, il a été montré qu'il existe un unique hyperplan (l'optimal), défini comme l'hyperplan maximisant la marge entre les échantillons.

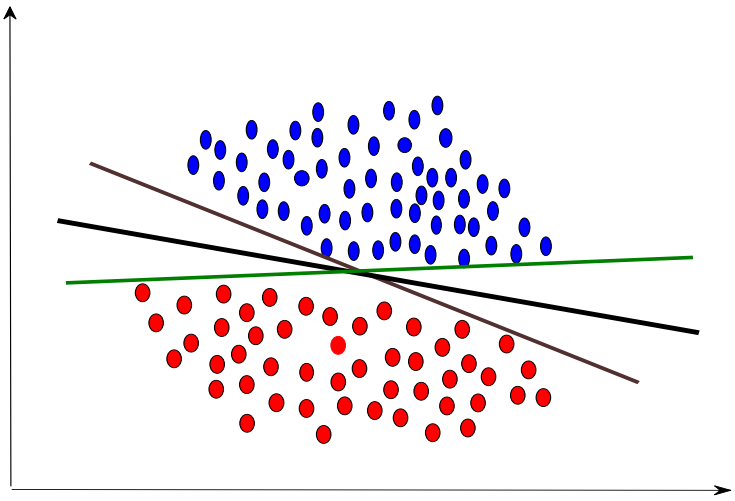


FIG. 3.1 – *Principe d'hyperplan séparateur, il en existe plusieurs. Celui qui correspond au minimum d'erreurs est l'hyperplan optimal.*

3.2.2 Marge maximale

Un autre objectif des séparateurs à vaste marge comme le terme l'indique, est de repousser le plus possible les deux classes l'une de l'autre. Ceci revient à maximiser la distance entre les points les plus proches du plan séparateur H (voir figure 3.2). Cette distance est appelée "Marge M". Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsqu'on classe un nouvel exemple. De plus, si on trouve un classifieur qui se comporte le mieux vis-a-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer de nouveaux exemples.

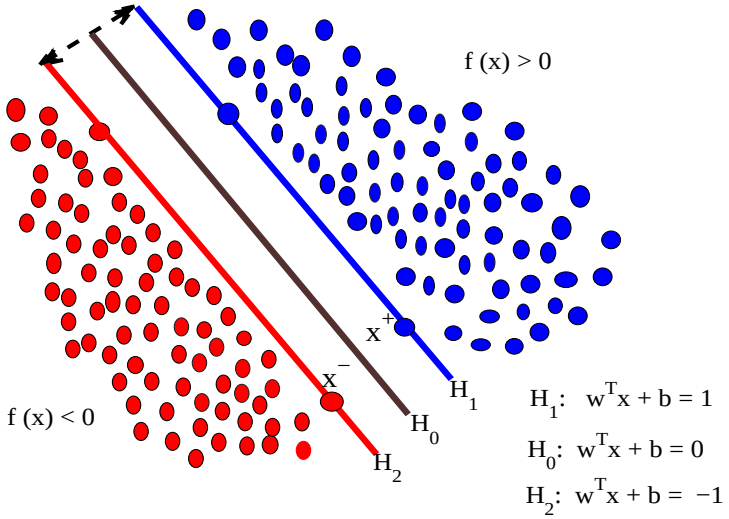


FIG. 3.2 – *L'hyperplan optimal H (en gras) avec la marge maximale M.*

Notons que ces deux objectifs sont etroitement liés car l'hyperplan optimal est justement celui qui maximise la marge.

3.3 Formulation du problème SVM

Le fondement mathématique des Séparateurs à Vaste Marge est expliqué dans plusieurs ouvrages comme [Vapnik 1982],[Vapnik 1998] et [Cortes 1982].

3.3.1 calcul de la marge

Soient x^+ et x^- deux points de classes différentes situés respectivement sur les frontières positive et négative délimitant la marge maximale. Pour simplifier le

problème d'optimisation, on considère que x^+ et x^- sont situés sur les hyperplans canoniques tels que $f(x^+) = +1$ c'est à dire point appartenant à l'hyperplan

$$H_1 : w^T x^+ + b = +1$$

et $f(x^-) = -1$ point appartenant à l'hyperplan

$$H_2 : w^T x^- + b = -1$$

H_1 *et* H_2 sont parallèles et distincts.

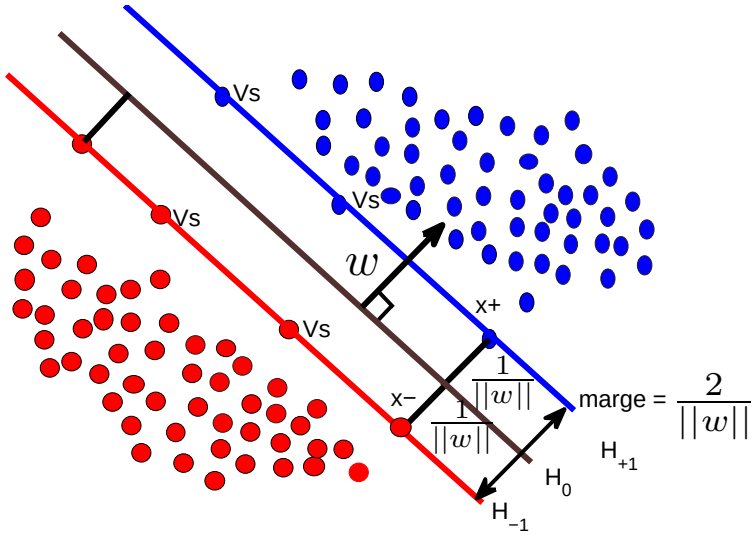


FIG. 3.3 – *marge*

On sait que la distance d'un point quelconque x à H est définie par :

$$d_{x,H} = \frac{|w^T x + b|}{\|w\|}$$

$$w^T x^+ + b = +1$$

$$w^T x^- + b = -1$$

$$x^+ = x^- + \lambda w$$

$$w(x^- + \lambda w) + b = 1$$

$$(wx^- + \lambda \|w\|^2) + b = 1$$

$$\underbrace{wx^- + b} + \lambda \|w\|^2 = 1$$

$$-1 + \lambda\|w\|^2 = 1 \Rightarrow \lambda = \frac{2}{\|w\|^2}$$

Nous somme intéressé par la marge

$$M = \|x^+ - x^-\| = \|\lambda w\| = \lambda\|w\| \;\Rightarrow\; M = \frac{2}{\|w\|^2}\|w\| = \frac{2}{\|w\|}$$

$$M = \frac{2}{\|w\|} \tag{3.1}$$

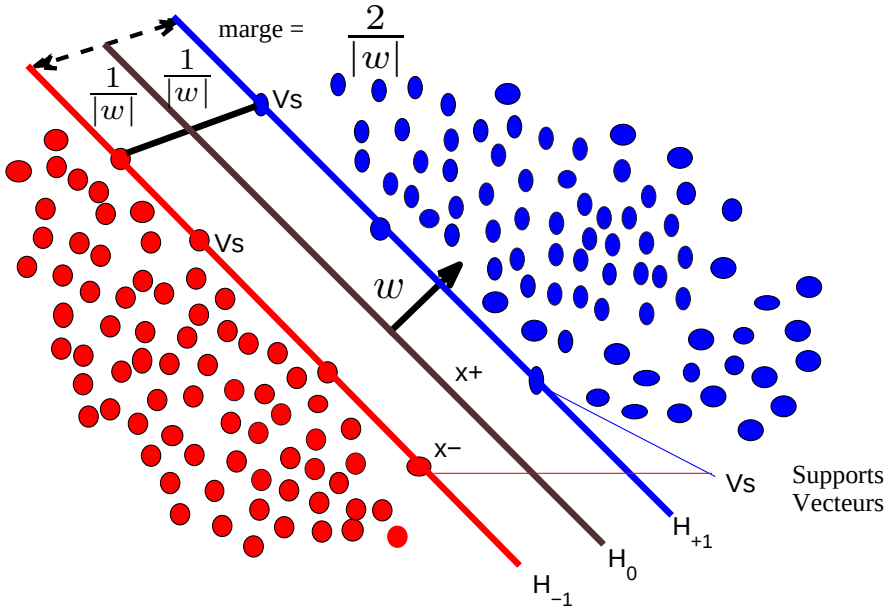


FIG. 3.4 – Exemple SVM Linéaire à marge dur avec support vecteurs

Nous déduisons à partir de là que maximiser la marge revient a minimiser $\|w\|$. Notons qu'il est plus aisé de minimiser $\|w\|^2$ plutôt que $\|w\|$.

3.3.2 position des contraintes

Pour tous les exemples de la base d'apprentissage, les deux inequations suivantes sont vérifiées :

$$x_i.w + b \geq 0 \quad pour \quad y_i = +1$$
$$x_i.w + b \leq 0 \quad pour \quad y_i = -1$$

Pour créer une marge entre les deux classes, une contrainte est posée vis à vis des exemples d'apprentissage. Elle est énoncée comme suit :

$$x_i.w + b \geq +1 \quad pour \quad y_i = +1 \qquad x_i.w + b \leq -1 \quad pour \quad y_i = -1$$

Ces deux conditions peuvent être combinées en une seule inéquation :

$$\forall i,\ y_i(x_i.w + b) \geq 1 \tag{3.2}$$

Cette contrainte signifie que le SVM tient compte non seulement de la position des exemples par rapport à l'hyperplan ($signe(f(x))$), mais aussi de leurs distances par rapport à ce dernier. (fig.(3.3),(3.4)).

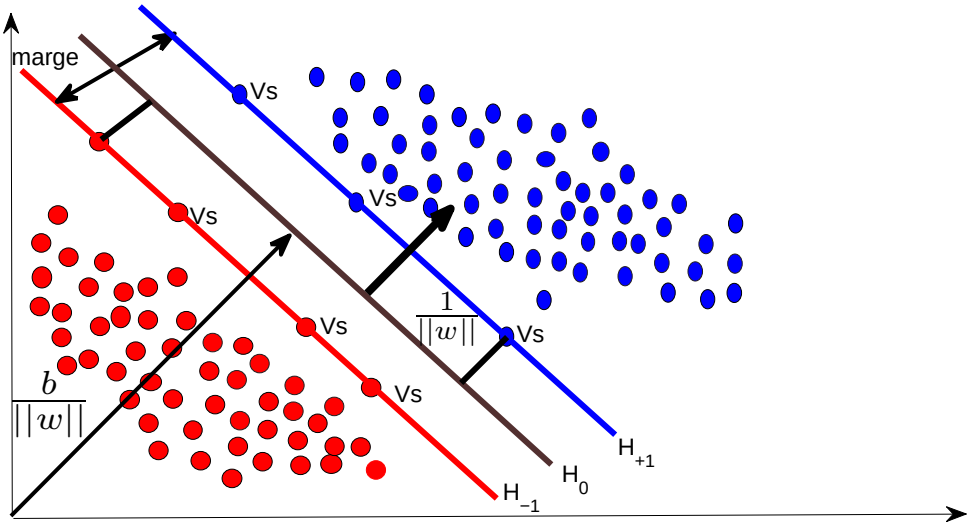


FIG. 3.5 – *hyperplan optimal, marge de séparation et vecteur de support*

3.4 Cas de données linéairement séparables

On se place ici dans le cas où le problème est linéairement séparable. Même dans ce cas simple, le choix de l'hyperplan séparateur n'est pas évident car il existe en effet une infinité d'hyperplans séparateurs. Pour résoudre ce problème, il a été montré qu'il existe un unique hyperplan optimal, défini comme étant celui qui maximise la marge entre les échantillons et l'hyperplan séparateur [Vapnik 1998].

3.4.1 Marge dure

Formulation primale

Le problème d'optimisation à résoudre lors de l'apprentissage vise à maximiser la

marge (3.3.1) sous contrainte que tous les points soient correctement classés (3.2). Un tel problème est énoncé comme suit :

$$\begin{cases} \min\limits_{w,b} & \frac{1}{2}\|w\|^2 \ , \\ Sc & y_i(w.x_i + b) \geq 1, \quad i = 1, \ldots, n \ . \end{cases} \tag{3.3}$$

Cette écriture du problème, appelée formulation primale, implique le réglage de $d + 1$ paramètres, d étant la dimension de l'espace des entrées $\mathcal{X}$. Cela est possible avec des méthodes de programmation quadratique [1] pour des valeurs de d assez petites, mais devient inenvisageable pour des valeurs de d dépassant quelques centaines. Heureusement, il existe une transformation de ce problème dans une formulation duale que l'on peut résoudre en pratique.

Expression duale du problème

D'après la théorie de l'optimisation, un problème d'optimisation possède une forme duale dans le cas où la fonction objectif et les contraintes sont strictement convexes. Dans ce cas, la résolution de l'expression duale du problème est équivalente à la solution du problème original. Ces critères de convexité sont réalisés dans le problème (3.3). Pour résoudre ces types de problèmes, on utilise une fonction que l'on appelle Lagrangien qui incorpore des informations sur la fonction objectif et sur les contraintes et dont le caractère stationnaire peut être utilisé pour détecter des solutions. Plus précisément, le lagrangien est défini comme étant la somme de la fonction objectif et d'une combinaison linéaire des contraintes dont les coefficients $\alpha_i \geq 0$ sont appelés multiplicateurs de Lagrange ou encore variables duales.

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1] \tag{3.4}$$

où les α_i sont les coefficients de Lagrange qui doivent être positifs ou nuls. Le problème (3.3) doit satisfaire les conditions de KKT (Karush-Kuhn-Tucker) qui consistent à annuler les dérivées partielles du Lagrangien 3.4 par rapport aux variables primales $w,\ b$.

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \displaystyle\sum_{i=1}^{n} \alpha_i y_i = 0 \ , \\ \dfrac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \displaystyle\sum_{i=1}^{n} \alpha_i y_i x_i \ . \end{cases} \tag{3.5}$$

[1] *Les problèmes d'optimisation pour lesquels la fonction et les contraintes sont linéaires ressortent des techniques de programmation linéaire, tandis que les problèmes d'optimisation dans lesquels la fonction est quadratique et les contraintes linéaires ressort des techniques de la programmation quadratique*

Complémentarité :

$$\alpha_i^*(y_i(w^{*T}x_i + b^*) - 1) \quad = \quad 0 \qquad\qquad i = 1, \ldots, n \tag{3.6}$$

Admissibilité primale :

$$y_i(w^{*T}x_i + b^*) \quad \geq 1 \qquad\qquad i = 1, \ldots, n \tag{3.7}$$

Admissibilité duale :

$$\alpha_i^* \quad \geq 0 \qquad\qquad i = 1, \ldots, n \tag{3.8}$$

Les conditions de complémentarité permettent de définir l'ensemble $\mathcal{A}$ des indices des contraintes actives (ou saturées) à l'optimum dont les multiplicateurs de Lagrange $\alpha_i^* > 0$ sont strictement positifs :

$$\mathcal{A} = \{i \in [1, n] \quad | \quad y_i(w^{*T}x_i + b^*) = 1\} \tag{3.9}$$

Pour les autres contraintes, la condition de complémentarité implique que leur mutiplicateur de Lagrange est égal à zéro et que l'observation associée vérifie strictement l'inégalité

$$\forall\, i \notin \mathcal{A} \quad y_i(w^{*T}x_i + b^*) > 1$$

observation inutile contrainte non saturée	observation importante contrainte active, vecteur support
$i \notin \mathcal{A}$ $\alpha_i = 0$ $y_i(w^{*T}x_i + b^*) > 1$	$i \in \mathcal{A}$ $\alpha_i > 0$ $y_i(w^{*T}x_i + b^*) = 1$

Le problème $\frac{1}{2}\|w\|^2$ étant convexe, $w(\alpha)$ est unique. En réinjectant les valeurs obtenues par les conditions KKT (3.5) dans l'equation (3.4), nous obtenons la forme duale du problème (3.3) comme suit :

$$\begin{cases} \min\limits_{\alpha} & \frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\alpha_i\alpha_i y_i y_j (x_j^T x_i) - \sum\limits_{i=1}^{n}\alpha_i \\ Sc & \sum\limits_{i=1}^{N}\alpha_i y_i = 0 \\ & \alpha_i \geq 0, \qquad i = 1, \ldots, n \end{cases} \tag{3.10}$$

L'ecriture matricielle de ce problème sera la suivante :

$$\begin{cases} \min\limits_{\alpha \in \mathbb{R}^n} & \frac{1}{2}\alpha^T G \alpha - e^T \alpha \\ Sc & y^T \alpha = 0 \\ & \alpha_i \geq 0, \qquad i = 1, \ldots, n \end{cases} \tag{3.11}$$

TAB. 3.1 – *Comparaison entre les formulations primale et duale*

Primal	Dual
$\min_{w,\ b} \quad \frac{1}{2}\|w\|^2$ $Sc \quad y_i(wx_i + b) \geq 1$ $i = 1, \ldots, n$	$\min_{\alpha \in \mathbb{R}^n} \quad \frac{1}{2}\alpha^T G\alpha - e^T\alpha$ $Sc \quad y^T\alpha = 0$ $\alpha_i \geq 0$ $i = 1, \ldots, n$
d + 1 inconnues (les influences des variables)	n inconnues (les influences des observations)
n contraintes linéaires	n contraintes de boites
QP de forme générale	G matrice des influences de chaque couple de points
préférable lorsque $d < n$	préférable lorsque $d \geq n$

avec G, matrice d'influence symétrique $n \times n$ de terme général $G_{ij} = y_i y_j x_j^T x_i$. Ces deux formulations équivalentes primale et duale méritent d'être mises en parallèle pour mieux faire ressortir leurs avantages respectifs et permettre de décider quand il est préférable d'utiliser la formulation primale et quand utiliser la formulation duale. La table 3.1 récapitule les principales caractéristiques des SVMs linéaires primales et duales.

Le problème (3.11) est un problème quadratique de dimension n (taille de l'ensemble d'apprentissage). Sa résolution revient à chercher les indices i des α_i^* positifs correspondant aux points x_i^* qui sont Vecteurs Supports (VS). Pour chaque nouveau point x à classer, la fonction de décision sera donnée par :

$$f(x) = \sum_{i=1}^{n} \alpha_i^* y_i^* (x_i^* \cdot x) + b \tag{3.12}$$

avec, $\sum_{i=1}^{n} \alpha_i^* y_i^* x_i^* = w$. La valeur de b peut être calculée en utilisant n'importe quel vecteur support x^* en posant : $\alpha^*[y^*(w.x^* + b) - 1] = 0$.

Conséquences

Il y a trois remarques intéressantes à faire à propos du résultat précédent :

1. La première découle de l'une des conditions de KKT, qui donne $\alpha_i[y_i f(x_i) - 1] = 0$ pour $i = 1, \ldots, N$ d'où $\alpha_i = 0$ ou bien $y_i f(x_i) = 1$. Les seuls points pour lesquels les contraintes du lagrangien sont actives sont donc les points pour lesquels $y_i f(x_i) = 1$. Ces points sont situés sur les hyperplans canoniques. En d'autres termes, seuls les vecteurs supports participent à la définition de l'hyperplan optimal.
2. La deuxième remarque découle de la première. Seul un sous-ensemble restreint de points est nécessaire pour le calcul de la solution. Ceci est donc efficace au niveau de la complexité.
3. La dernière remarque est que l'hyperplan solution ne dépend que du produit scalaire entre le vecteur d'entrée et les vecteurs supports. Cette remarque est à l'origine de la deuxième innovation majeure des SVMs : le passage à un espace de caractéristiques $\mathcal{F}$ grâce à la fonction noyau.

3.4.2 Marge souple

Les données d'apprentissage peuvent être bruitées et non séparables, même dans l'espace $\mathcal{F}$. En présence d'une donnée aberrante, la maximisation de la marge de séparation peut conduire à l'obtention d'un hyperplan qui n'est plus du tout optimal. En effet, il suffit qu'une observation des deux classes viole la contrainte (3.2)pour que ce problème n'ait plus de solution. Pour contourner ce problème, Cortes et Vapnik [Cortes 1982, Vapnik 1998] introduisèrent la notion de classifieurs à marge molle en introduisant des variables ressort ξ_i (*slack variables*) pour relâcher sensiblement les contraintes sur la marge. Les contraintes (3.2) deviennent alors :

$$y_i f(x_i) \geq 1 - \xi_i, \qquad \xi_i > 0 \qquad (i = 1, ..., n)$$

.

$$y_i(w^T x_i + b) < 1 \qquad \Rightarrow \qquad \xi_i = 1 - y_i(w^T x_i + b) > 0 \tag{3.13}$$

La figure (3.6) montre une telle situation.

Definition 3.4.1 *Pour toute observation (x_i, y_i), on definit la variable d'écart par :*

$$\xi_i = max(0, 1 - y_i(w^T x_i + b)) = |1 - y_i f(x_i)|_+ \tag{3.14}$$

Cette quantite mesure l'echec d'un point à atteindre la marge. En introduisant un paramètre de pénalisation de chaque erreur, on associe à cette définition une fonction

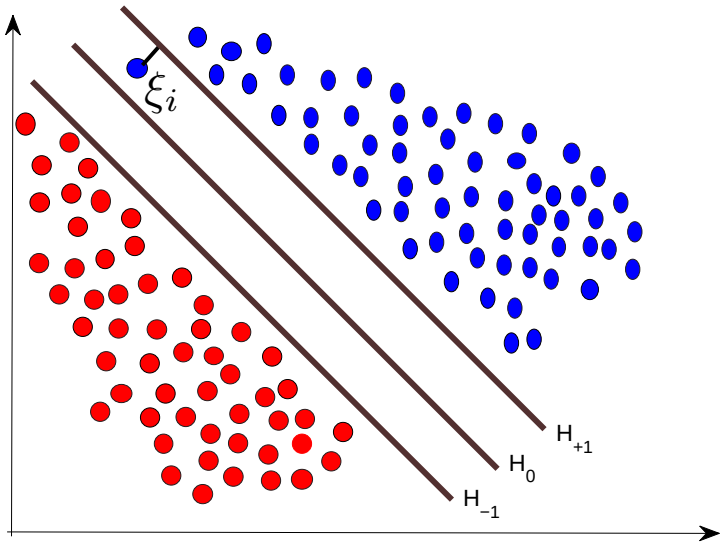


FIG. 3.6 – *hyperplan sous optimal lié à la présence d'une donnée aberante*

*de coût appelée "cout charnière" ou "*hinge loss*" relativement à sa forme indiquée par la figure 3.7.*

La fonction de perte qu'on cherchera à minimiser est définie par :

$$\ell(y_i,\ f(x_i)) = C\sum_{i=1}^{n} |1 - y_i f(x_i)|_+ = C\sum_{i=1}^{n} \xi_i \tag{3.15}$$

Cette fonction recommandée par V.Vapnick [Vapnik 1998] est une fonction convexe, elle est couramment appelée "*hinge loss*". Elle garantie une solution unique au problème.

Formulation du problème primal

Dans le contexte défini précédemment, nous devons poser un problème d'optimisation dont l'objectif serait de maximiser la marge et minimiser la fonction de pertes (le coût de la classification). Le problème (3.3) devient alors :

$$\begin{cases} \min\limits_{w,b,\xi} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \ , \\ Sc & y_i(w^T x_i + b) \geq 1 - \xi_i \ , \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \ . \end{cases} \tag{3.16}$$

La constante C est un paramètre depénalisation des erreurs. Autrement dit, il détermine la tolérance du SVM aux exemples mal classés. Elle permet le contrôle du

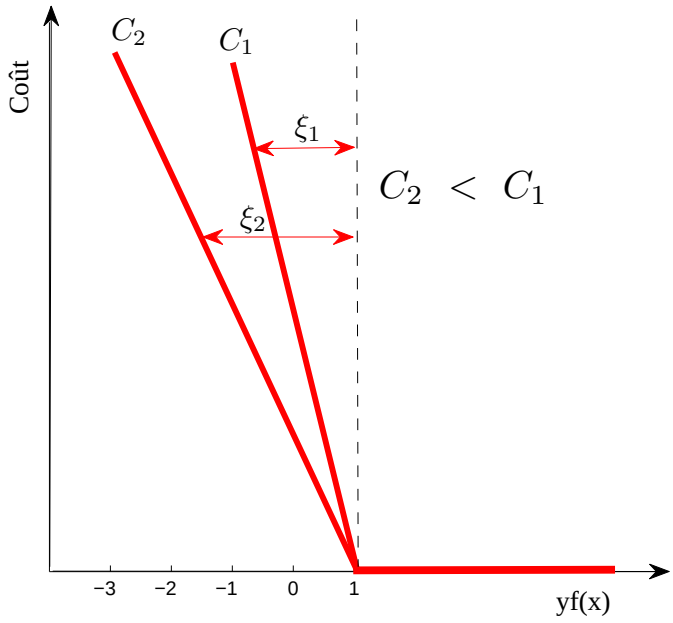


FIG. 3.7 – *Représentation du compromis entre la largeur de la marge souple et le coût d'une erreur.*

compromis entre la valeur de l'erreurs de classification et la largeur de la marge (voir figure 3.7). Plus C est grand, plus on pénalise les mauvaises classifications et la complexité de la classe de fonctions de décision devient grande. Le choix automatique d' un tel paramètre est un problème statistique majeur.

Remarque 3.1 *On constate que si toutes les variables d'écart $\xi_i = 0$, on retrouve le problème des SVM linéairement séparables.*

Comme nous l'avons fait précédemment, on recherche un point selle du Lagrangien comme solution du problème (3.16)

Formulation du problème dual

Le Lagrangien du problème (3.16) est donné par :

$$\mathcal{L}(w,b,\xi,\alpha,\beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(w^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^{n}\beta_i\xi_i \quad (3.17)$$

Avec α et β, des coefficients de Lagrange positifs ou nuls. Les conditions d'optimalité fournies à l'égard du Lagrangien (3.17) se traduisent par ses dérivées partielles nulles par rapport aux variables primales $w,\ b,\ \xi$(conditions de KKT).

$$\begin{cases} \dfrac{\partial\mathcal{L}(w,b,\xi,\alpha,\beta)}{\partial b} = 0 \Rightarrow \displaystyle\sum_{i=1}^{n}\alpha_i y_i = 0 \ , \\ \dfrac{\partial\mathcal{L}(w,b,\xi,\alpha,\beta)}{\partial w} = 0 \Rightarrow w = \displaystyle\sum_{i=1}^{n}\alpha_i y_i x_i \\ \dfrac{\partial\mathcal{L}(w,b,\xi,\alpha,\beta)}{\partial \xi} = C - \alpha_i - \beta_i \ = \ 0 \quad \Rightarrow \beta_i = C - \alpha_i \end{cases} \quad (3.18)$$

Les conditions de KKT 3.18 injectées dans le Lagrangien (3.17), nous permettent d'énoncer le problème dual à résoudre :

$$\begin{cases} \max\limits_{\alpha} & \displaystyle\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j (x_i^T x_j) \ , \\ Sc & \displaystyle\sum_{i=1}^{n}\alpha_i y_i = 0 \ , \\ & 0 \le \alpha_i \le C; \quad i = 1,\ldots,n \ . \end{cases} \quad (3.19)$$

Ce problème est similaire au problème (3.10) avec une contrainte supplémentaire sur les coefficients de Lagrange α_i. La plupart des méthodes d'optimisation sont basées sur des conditions d'optimalité du second ordre (contrainte duales). La correspondance entre les contraintes primales et les contraintes duales du problème précédent est récapitulée sur la table 3.2.

3.5 Cas de données non linéairement séparables

Les machines à vecteurs supports présentées ci-dessus ne concernent que les modèles linéaires. Il serait complètement illusoire de se dire que l'on pourrait séparer efficacement n'importe quel jeu de données par un simple hyperplan. Si par exemple les données des deux classes se chevauchent sévèrement (intrinsèquement non séparables), aucun hyperplan séparateur ne serait satisfaisant.

Par ailleurs, nous remarquons que dans les situations précédentes, les données n'interviennent dans le problème dual que par leurs produits scalaires $\langle x_i, x_j\rangle$. Ceci

TAB. 3.2 – *Relation entre les contraintes primales et duales.*

les contraintes primales	les contraintes duales
$y_i f(x_i) > 1$ et $\xi = 0$	$\alpha_i = 0$
$y_i f(x_i) = 1$ et $\xi = 0$	$0 < \alpha_i < C$
$y_i f(x_i) < 1$ et $\xi > 0$	$\alpha_i = C$

a donnée naissance à l'astuce du noyau qui fait d'ailleus le point fort des SVMs car elle permet de traiter le cas non-linéaire.

L'astuce du noyau Consiste à utiliser une transformation non linéaire pour redécrire le problème dans un espace de plus grande dimension, dans lequel elles pourront être séparées par un hyperplan. L'idée de Boser *et al* [Boser 1992] est que des modèles peuvent être considérablement enrichis en projetant les données non linéairement séparables dans un espace caractéristique de plus grande dimension à l'aide d'une fonction non lineaire choisie a priori. Cette opération permet d'augmenter la séparabilité des données (voir figure 3.8).

l'idée consiste alors à trouver une fonction, que l'on nomme noyau, qui correspond à un produit scalaire dans l'espace de redescription, mais qui ne nécessite pas de redécrire les données dans cet espace pour en calculer sa valeur. On peut alors appliquer le même algorithme dans ce nouvel espace, ce qui se traduit par une frontière de décision non linéaire dans l'espace initial.

Considérons l'application non linéaire définie par :

$$\phi : \mathcal{X} \mapsto \mathcal{F}$$

$$x \mapsto \phi(x)$$

Il suffit alors d'appliquer l'algorithme d'apprentissage dans $\mathcal{F}$ et non dans $\mathcal{X}$ en considérant l'ensemble $(\phi(x_i),\ y_i) \in \mathcal{F} \times \mathcal{Y}$ avec $i = 1, .., n$ et $\mathcal{Y} = \{+1, -1\}$. L'espace $\mathcal{F}$ ainsi obtenu est appelé espace des caracteristiques [2]. ou aussi espace transformé. Tout ce qui reste à faire serait de resoudre le probleme dans l'espace $\mathcal{F}$, en remplacant $\langle x_i, x_j \rangle$ par $\langle \phi(x_i), \phi(x_j) \rangle$ Sous certaines hypotheses sur ϕ, le produit scalaire $\langle \phi(x_i), \phi(x_j) \rangle$ peut se calculer facilement a l'aide d'une fonction symetrique K, dite noyau, definie par :

[2] *Feature space en anglais*

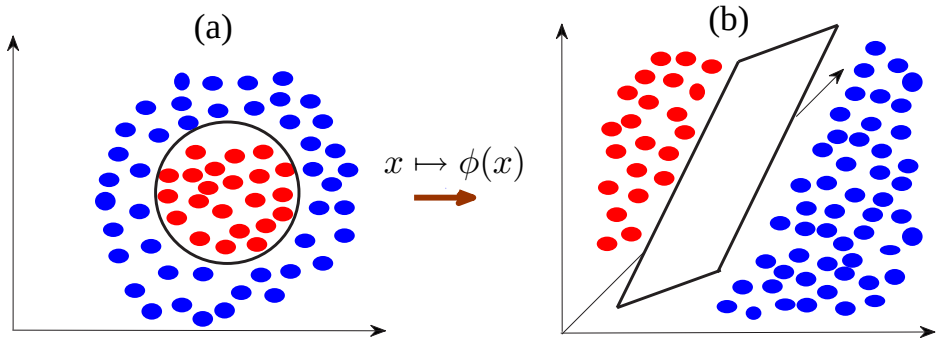


FIG. 3.8 – *a)Aucun hyperplan separateur b)hyperplan separateur dans un espace plus elevé*

$$K(x_i, x_j) =< \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) > \tag{3.20}$$

Dans la pratique on choisit un noyau K qui satisfait les conditions de Mercer afin de garantir la decomposition. L'hyperplan séparateur obtenu dans l'espace $\mathcal{F}$ est appele hyperplan optimal généralisé.

La figure 3.8 montre un exemple de projection de R^2 dans R^3.

Pour les données non linéairement séparable, le problème dual (3.19) devient :

$$\begin{cases} \max\limits_{\alpha} & \sum\limits_{i=1}^{n} \alpha_i - \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \ , \\ Sc & \sum\limits_{i=1}^{n} \alpha_i y_i = 0 \ , \\ & 0 \leq \alpha_i \leq C; \quad i = 1, \ldots, n \ . \end{cases} \tag{3.21}$$

La séparation est toujours linéaire, mais dans un espace différent. la fonction de décision dans ce cas est donnée par :

$$f(x) = \sum_{i=1}^{n} \alpha_i^* y_i^* K(x_i, x_j) + b \tag{3.22}$$

Ce problème reste un programme quadratique avec des contraintes de boites.

La demonstration reprend la démarche adoptée dans le cas linéaire puisque le lagrangien s'écrit :

$$\mathcal{L} = \frac{1}{2}\|f\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(f(x_i)+b) - 1 + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i \tag{3.23}$$

où les $\alpha_i \geq 0$ et les $\beta_i \geq 0$ sont les multiplicateurs de Lagrange associés aux contraintes. Les conditions d'optimalité de Karush, Kuhn et Tucker du problème permettent de caractériser la solution du problème. Les exemples sont maintenant réparties dans trois ensembles $I_C, \quad I_0, \quad I_\alpha$:

1. $I_C \Rightarrow \alpha_i = C$,
2. $I_0 \Rightarrow \alpha_i = 0$,
3. $I_\alpha \Rightarrow 0 \leq \alpha \leq C$,

avec l'ensemble des contraintes actives : $\mathcal{A} = I_\alpha \cup I_C$.

Remarque 3.2 *La plupart des variantes présentées dans le cas linéaire s'étend au cas non linéaire en remplaçant le vecteur w par la fonction f.*

3.6 Algorithmes de résolution

Dans la section précédente nous avons vu que l'apprentissage des SVM se ramène à la maximisation d'une forme quadratique convexe sous des contraintes linéaires. Dans ces cas il n'y a pas de problèmes de minimums locaux et la solution peut être trouvée en utilisant des algorithmes efficaces. Par contre les méthodes classiques de résolution sont inadaptées aux problèmes de grande taille. Pour gérer les problèmes de grande taille, il existe des méthodes dites de décomposition. Elles reviennent à décomposer le problème en plusieurs petits sous-problèmes tels que la résolution de chacun d'eux fournisse une approximation toujours meilleure de l'optimum. L'algorithme SMO(*Sequential Minimal Optimization,*) proposé par Platt [Platt 1999] est un cas extrême de ces méthodes.(voir Annexe)
Ils existent plusieurs algorithmes de résolution du problème quadratique convexe, différentes méthodes de resolution sont détaillés (voir Annexe)
En pratique : la taille n des échantillons est très grande $\Rightarrow$ G est de grande dimension, dense et mal conditionnée.
Dans ce qui suit, nous donnons les étapes de résolution ainsi que les différents aspects liés à la mise en oeuvre de SimpleSVM. Cet algorithme s'est révelé être non seulement très efficace en terme de temps de calcul et de précision mais aussi fertile en extensions potentielles.

3.6.1 Simple SVM

L'algorithme SimpleSVM a été proposé en 2003 par Vishwanathan *et al* [Vishwanathan 2003]. Il a été ensuite amélioré en 2005 par G.Loosli *et al* [Loosli 2006] en créant un *SVM-Toolbox* sous Matlab. Cet algorithme est Basé sur le principe de contraintes actives. Il permet de résoudre efficacement un problème quadratique sous contraintes.

3.6.1.1 SVM : un problème quadratique sous contraintes

Le problème dual, que l'on écrit sous forme vectorielle pour plus de clarté

$$\begin{cases} \max_\alpha -\frac{1}{2}\alpha' G\alpha + 1'\alpha \\ \qquad y'\alpha = 0 \\ \qquad 0 \leq \alpha_i \leq C \qquad\qquad i \in [1, ..., n] \end{cases} \tag{3.24}$$

G désigne la matrice noyau pondérée par les étiquettes des points, telle que

$$G_{ij} = y_i y_j k(x_i, x_j)$$

. Ce système dual est un programme quadratique convexe qui présente deux particularités : il y a n inconnues avec (2n + 1) contraintes et G est une matrice semi-définie positive. La solution du problème SVM sera alors donnée par la résolution d'un problème d'optimisation quadratique en dimension n sous contraintes de boîte.

3.6.1.2 La propriété utilisée : la parcimonie

Notons tout d'abord que les formes à optimiser sont convexes et par là même n'admettent qu'une seule solution optimale. Notons par ailleurs que les contraintes sont linéaires et en nombre fini : la solution optimale α vérifie pour chacune de ses composantes α_i les conditions de Karuch Kuhn Tucker (KKT), c'est-à-dire

$$\alpha_i = 0 \quad \Rightarrow \quad (f(x_i) + b)y_i > 1 \tag{3.25}$$

$$0 < \alpha_i < C \quad \Rightarrow \quad (f(x_i) + b)y_i = 1 \tag{3.26}$$

$$\alpha_i = C \quad \Rightarrow \quad (f(x_i) + b)y_i < 1 \tag{3.27}$$

Le principe général des méthodes de décomposition et/ou de contraintes actives repose sur l'observation que seuls les points non contraints dans la solution nécessitent le calcul de leur coefficient : c'est ***la parcimonie*** [Loosli 2006]. En effet, les autres ont pour coefficient une valeur fixe donnée par le problème : celle de la borne qui les contraint (0 et C). Partant de ce constat, il suffirait de connaître la répartition des points dans trois groupes (contraints à 0, contraints à C et non contraints)

pour avoir très rapidement (en une résolution de système linéaire sur les points non contraints) la solution du problème. On appellera par la suite les groupes I_0, I_C *et* I_w , respectivement pour les points contraints à 0, à C et non contraints. Ce constat a amené à différentes techniques de décompositions et différents algorithmes que nous allons détailler après avoir donné quelques outils communs à plusieurs méthodes.

3.6.1.3 Outils utilisés

Directions admissibles Les contraintes de boîte ' restreignent la solution à un hypercube de dimension n . La contrainte d'égalité $\Sigma y_i \alpha_i$ restreint en plus la solution à un polytope $\mathcal{F}$ de dimension n - 1. On considère un point admissible $\alpha \in \mathcal{F}$ et une droite contenant ce point. Cette droite indique une direction admissible si son intersection avec le polytope contient des points autres que α Les algorithmes de direction admissible mettent à jour de manière itérative la position d'un point admissible α_t. En premier lieu ils choisissent une direction u_t et ensuite ils suivent cette direction à la recherche du point admissible α_{t+1}. qui maximise la fonction coût. L'optimum est atteint quand aucune amélioration n'est plus possible[Zoutendijk. 1960].

Méthodes de contraintes actives Le principe des méthodes de contraintes actives est de répartir le plus efficacement possible les points dans les trois groupes I_0, I_C *et* I_S et de trouver les valeurs des $\alpha_i, i \in I_S$ par la résolution d'un système linéaire. Pour les SVM, l'algorithme est baptisé SimpleSVM [Vishwanathan et al., 2003] (toutefois il s'applique à tout problème posé sous la forme quadratique convexe sous contraintes de boîtes) et permet de trouver itérativement la répartition en trois groupes. A chaque étape, on résout pour le groupe I_w la minimisation sans contrainte. Si une des valeurs de α ainsi calculées viole les contraintes (c'est-à-dire que la solution trouvée ne se situe plus dans la boîte de contraintes) alors on projette α dans l'espace admissible (ce qui revient en pratique à changer de groupe le point indicé par la valeur qui viole les contraintes). Le schéma itératif de ce type d'algorithme est le suivant : Il reste donc à définir comment choisir la répartition des groupes, comment calculer les coefficients et comment évaluer la performance de la solution ainsi trouvée.

Forme primale

$$\begin{cases} \min\limits_{f,b,\xi} & \frac{1}{2}\|f\|^2_{\mathcal{H}} + C\sum_{i=1}^{n}\xi_i & \\ Sc & y_i(f(x_i)+b) \geq 1-\xi_i & i=1,\ldots,n \\ & \xi i \geq 0 & i=1,\ldots,n \end{cases} \tag{3.28}$$

où C est un scalaire permettant de régler la régularité de la fonction de décision, b un scalaire appelé le biais et les ξ_i des variables d'écart. La solution de ce problème

Algorithme 3 Schéma itératif de l'algorithme SimpleSVM

1 : *choisir une répartition initiale*

2 : *calculer une première solution*

3 : ***tant que*** *la solution courante n'est pas optimale* **faire**

4 : *mettre à jour à la répartition des groupes*

5 : *calculer les coefficients associés au groupe* I_S

6 : ***fin tant que***

est aussi le point selle du lagrangien :

$$\mathcal{L}(f, b, \xi, \alpha, \beta) = \frac{1}{2}\|f\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(f(x_i)+b) - 1 + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i \quad (3.29)$$

où les $\alpha_i \geq 0$ et les $\beta_i \geq 0$ sont les multiplicateurs de Lagrange associés aux contraintes. Les valeurs à identifier ici sont les variables d'écart ξ et le biais b. On tire une partie des conditions de Kuhn et Tucker :

$$\begin{array}{lll} \frac{\partial L(f,b,\xi,\alpha,\beta)}{\partial f} = 0 & \Rightarrow & f(x) - \sum_{i=1}^{n}\alpha_i y_i k(x_i, x) = 0 \\ \frac{\partial L(f,b,\xi,\alpha,\beta)}{\partial b} = 0 & \Rightarrow & \sum_{i=1}^{n}\alpha_i y_i = 0 \\ \frac{\partial L(f,b,\xi,\alpha,\beta)}{\partial \xi} = 0 & \Rightarrow & C - \alpha_i - \beta_i = 0. \qquad i \in [1, ..., n] \end{array} \quad (3.30)$$

On en déduit que $f(x) = \sum_{i=1}^{n}\alpha_i y_i k(x_i, x)$. C'est grâce à cette relation que l'on peut éliminer f du lagrangien pour aboutir à la formulation duale suivante, parfois appelée le dual de Wolfe :

Forme duale

$$\begin{cases} \max\limits_{\alpha\in\mathbb{R}^n} -\frac{1}{2}\alpha^T G\alpha + e^T\alpha \\ \qquad y^T\alpha = 0 \\ \qquad 0 \leq \alpha_i \leq C \qquad i \in [1, ..., n] \end{cases} \quad (3.31)$$

où G est la matrice d'influence de terme général

$$G_{ij} = y_i y_j k(x_i, x_j) \;\; et \;\; e = [1, ..., 1]^T.$$

La solution du problème SVM sera alors donnée par la résolution d'un problème d'optimisation quadratique en dimension n sous contraintes de boîte. Pour résoudre le problème dual, le Lagrangien est encore une fois utilisé :

3.6.1.4 Relations entre variables primales et duales

Nous avons donné dans (3.16) et (3.19) les formes primales et duales de SVM binaires à marge douce. Nous rappelons ici ces deux formes pour faire apparaître les équivalences entre les paramètres des deux formulations.

Dans le problème précédent (3.31), chaque inconnue α_i peut être interprétée comme l'influence de l'exemple (x_i, y_i) dans la solution. Du fait que seuls les points frontière sont importants pour la discrimination et qu'ils sont a priori peu nombreux, un grand nombre de ces coefficients α vont être nuls (Steinwart, 2003) [Steinwart 2003]. Il sera alors pertinent de répartir les m inconnues du problème en trois groupes de points $[1, m] = I_s \cup I_c \cup I_0$, définis en fonction de la valeur du multiplicateur de Lagrange α associé

- $[I_s]$ le groupe des points supports est celui des vecteurs supports candidats, c'est-à dire pour lesquels $0 < \alpha_i < C$. Ces points sont à l'intérieur de la boîte et satisfont les contraintes. Il est aussi appelé l'ensemble de travail (working set), car il contient les variables sur lesquelles il faut " travailler " pour connaître leur valeur.
- $[I_c]$ le groupe des points saturés est celui des vecteurs pour lequel $\alpha_i = C$. Ces points sont sur le bord de la boîte. Si ces points apparaissent trop proches de la classe opposée, voire complètement mélangés, limiter leur contribution à la frontière de décision permet de régulariser la solution. Dans la solution finale ces points auront tous la valeur de leur α fixée à C : ils seront contraints et contribueront à la décision.
- $[I_0]$ le groupe des points inactifs est celui des points pour lesquels $\alpha_i = 0$. Ces points se situent également sur l'arête de la boîte. Dans ce cas, les vecteurs sont loin de la frontière entre les classes et ne contribuent en rien à la solution. Dans la solution finale ces points auront tous la valeur de leur α fixée à 0 : ils seront contraints.

Ces trois ensembles nous aident à reformuler le problème d'optimisation (3.31) :

Forme duale

$$\begin{cases} \max\limits_{\alpha \in \mathbb{R}^n} -\frac{1}{2}\alpha^T G \alpha + e^T \alpha \\ avec \qquad \alpha^T y = 0 \\ et \qquad 0 < \alpha_i < C \qquad i \in I_s \\ et \qquad \alpha_i = C \qquad i \in I_C \\ et \qquad \alpha_i = 0 \qquad i \in I_0 \end{cases} \tag{3.32}$$

La relation entre les paramètres de la formulation duale que nous venons d'obtenir (3.32) et les paramètres initiaux du problème (3.28) est importante. Il est possible de la rendre explicite en passant une fois encore par l'écriture d'un lagrangien, cette fois celui du problème dual (3.32) vu comme un problème de minimisation :

$$\mathcal{L}(\alpha, \lambda, \mu, \nu) = \frac{1}{2}\alpha^T G\alpha - e^T\alpha - \lambda\alpha^T y - \nu^T\alpha + \mu^T(\alpha - Ce) \tag{3.33}$$

ou les multiplicateurs de Lagrange α, μ, et ν doivent être positifs. Ce lagrangien nous ramene au probleme primal et peut être comparé à celui obtenu à partir du probleme primal (3.28) après avoir remplacé la variable f par α mais avant simplification, soit :

$$\mathcal{L}(\alpha, b, \xi, \beta) = \frac{1}{2}\alpha^T G\alpha - e^T\alpha - b\alpha^T y + \xi^T(\alpha - \beta + Ce) \tag{3.34}$$

3.6.1.5 les équivalences entre les multiplicateurs de Lagrange

les multiplicateurs α, μ, et ν de l'equation (3.33)sont comparés aux multiplicateurs l'equation (3.34)

- $[0 < \alpha < C]$: dans ce cas la condition de Kuhn et Tucker stipulant l'annulation du gradient du lagrangien appliquée à l'equation (3.33) par rapport à α s'écrit :

  $$G\alpha + \lambda y - e = 0.$$

  Dans ce cas aussi la contrainte initiale est saturée, c'est-à-dire

  $$y_i(f(x_i) + b) = 1$$

  que l'on peut réécrire en utilisant les mêmes notations que précédemment l'annulation du gradient du lagrangien appliquée à l'equation (3.34) par rapport à α

  $$G\alpha + by - e = 0.$$

  Le multiplicateur de Lagrange λ associé à la contrainte d'égalité est donc égal au biais $\lambda = b$.

- $[\alpha = C]$: dans ce cas $\xi \neq 0, \nu = 0$ et

  $$\mu = -G\alpha - by + e = \xi.$$

  Si le multiplicateur de Lagrange est positif, la variable d'ecart l'est aussi. Par consequent

  $$y_i(f(x_i) + b) < 1$$

  doit être verifie pour ce cas.
- $[\alpha = 0]$:dans ce cas $\xi = \mu = 0$ et

  $$\nu = G\alpha + by - e$$

Si le multiplicateur de Lagrange est positif, alors

$$y_i(f(x_i) + b) > 1$$

doit être vérifié pour ce cas.
Ces relations sont résumées dans le tableau (3.3). Vérifier ces conditions d'optimalité revient à calculer

$$G\alpha + by - e$$

puis à s'assurer que pour tous les points non supports ($\alpha = 0$) cette quantité soit positive et que pour tous les points support saturés ($\alpha = C$) cette quantité soit bien négative.

La table (3.3) indique la répartition de l'ensemble d'apprentissage en trois sous ensembles définis par les contraintes du problème (3.34).

Esemble	Contraintes Initiales	Contraintes Primales	Contraintes Duales
I_S	$y_i(f(x_i) + b) = 1$	$0 < \alpha < C, \quad \xi = 0$	$\nu = 0, \quad \mu = 0$
I_C	$y_i(f(x_i) + b) = 1 - \xi_i$	$\alpha = C, \quad \xi > 0$	$\nu = 0, \quad \mu > 0$
I_0	$y_i(f(x_i) + b) > 1$	$\alpha = 0, \quad \xi = 0$	$\nu > 0, \quad \mu = 0$

TAB. 3.3 – Ensemble de partition

3.6.1.6 Points contraints et non contraints

Pour faire apparaître les équivalences entre les multiplicateurs de Lagrange et les variables du problème primal, les trois groupes I_w, I_0 *et* I_C sont considérés successivement et explicités dans le Lagrangien :[Loosli 2006]

$$\mathcal{L}(\alpha, \mu, \eta, \upsilon) = -\frac{1}{2}\alpha' G\alpha + \mathbf{1}'\alpha - \mu y'\alpha - \eta'(\alpha - C\mathbf{1}) + \vartheta'\alpha \tag{3.35}$$

avec μ, η et υ des multiplicateurs de Lagrange positifs.

$$\mathcal{L}(\alpha_w, \alpha_C, \mu, \eta_w, \eta_0, \upsilon_w, \upsilon_C) = -\frac{1}{2}\alpha'_w G_{(w,w)}\alpha_w - \frac{1}{2}\alpha'_C G_{(C,C)}\alpha_C - \alpha'_C G_{(C,w)}\alpha_w \tag{3.36}$$

$$+ \mathbf{1}'_w\alpha_w + \mathbf{1}'_C\alpha_C \tag{3.37}$$

$$-\mu(y'_w\alpha_w + y'_C\alpha_C) \tag{3.38}$$

$$-\eta'_w(\alpha_w - C\mathbf{1}_w) + \eta'_0(C\mathbf{1}_0) \tag{3.39}$$

$$+\vartheta'_w\alpha_w + \vartheta'_C\alpha_C \tag{3.40}$$

Points non contraints : I_w. Si on ne considère que les points du groupe I_w, tous ont un multiplicateur α compris entre 0 et C. Les multiplicateurs η_w et ϑ_w correspondants aux points non contraints sont nuls puisque les contraintes de boîte ne sont pas actives. L'annulation du gradient de (3.36)par rapport à $\alpha_{(w)}$ donne :

$$G_{(w,.)}\alpha + \mu y_w - \mathbf{1}_w = 0.$$

Par ailleurs, dans l'espace primal, les points non contraints sont situés sur la marge, c'est-à-dire que

$$y_i(f(x_i) + b) - 1 = 0.$$

Sous forme vectorielle pour tous les points de I_w, cela équivaut à

$$G_{(w,.)}\alpha + by_w - \mathbf{1}_w = 0_w.$$

Par identification, on obtient la relation $b = \mu$.

Points contraints a C : I_C. Les points de I_C sont tous bornés à C, ainsi seule la contrainte superieure de boite est active,$\nu = 0 \;\; et \;\; \eta > 0$. L'annulation du gradient (3.36) par rapport à α_C donne .

$$-G_{(C,.)}\alpha - \mu y_C + \mathbf{1}_C = \eta_C.$$

Par ailleurs, dans l'espace primal, les points contraints à la borne C sont situés du mauvais côté de la marge par rapport à leur classe et les variables d'écart associées ξ_C sont strictement positives, c'est-à-dire que

$$y_i(f(x_i) + b) - 1 = -\xi_i.$$

Sous forme vectorielle pour tous les points de I_C, cela equivaut à

$$G_{(C,.)}\alpha + by_C - \mathbf{1}_C = -\xi_C.$$

Par identification, on obtient la relation $\xi_C = \eta_C$. Par ailleurs, ces valeurs sont nulles pour les autres groupes de points car la contrainte de boite a C est inactive et par consequent, $\xi = \eta$.

Points contraints à 0 : I_0. Les points de I_0 sont tous bornes à 0, ainsi seule la contrainte inférieure de boite est active, $\nu_0 > 0 \;\; et \;\; \eta_0 = 0$. L'annulation du gradient du Lagrangien par rapport à α_0 donne

$$G_{(0,.)}\alpha + \mu y_0 - \mathbf{1}_0 = \nu_0 > 0.$$

Par ailleurs, dans l'espace primal, les points contraints à la borne 0 sont situés du bon côté de la marge par rapport à leur classe, cĄ'est-a-dire que

$$y_i(f(x_i) + b) - 1 > 0.$$

Sous forme vectorielle pour tous les points de I_0, cela equivaut à

$$G_{(0,.)}\alpha + by_0 - \mathbf{1}_0 > 0.$$

Récapitulatif : Ces relations sont résumées dans le tableau (3.4). Vérifier ces conditions d'optimalité revient à calculer $G\alpha + by - 1$ puis à s'assurer que pour tous les points de

- $I_0(\alpha = 0)$ cette quantité soit positive
- et que pour tous les points de $I_C(\alpha = C)$ cette quantité soit bien négative.

Points	Contraintes	Contraintes Initiales	Condition d'optimalité
non cont-	$0 < \alpha < C$	$y_i(f(x_i) + b) = 1$	$G_{(w,.)}\alpha + by_w - \mathbf{1}_w = 0.$
$raints: \quad I_w$	$\eta_w = 0$ et $\vartheta_w = 0$		$G_{(w,.)}\alpha + by_w - \mathbf{1}_w = 0.$
contraints	$\alpha = C,\ \nu_C = 0$	$y_i(f(x_i) + b) = 1 - \xi_i$	$G_{(C,.)}\alpha + by_C - \mathbf{1}_C = -\xi_C.$
à $C: \ I_C$	et $\quad \xi_C = \eta_C > 0.$		$G_{(C,.)}\alpha + by_C - \mathbf{1}_C < 0$
contraints	$\alpha = 0,\ \nu_0 > 0,$	$y_i(f(x_i) + b) > 1$	$G_{(0,.)}\alpha + by_0 - \mathbf{1}_0 = \nu_0 > 0.$
à $0: \ I_0$	$\eta_0 = 0$		$G_{(0,.)}\alpha + by_0 - \mathbf{1}_0 > 0$

TAB. 3.4 – Condition d'optimalité pour les points I_w, I_C et I_0

La table (3.4) indique la répartition de l'ensemble d'apprentissage en trois sous ensembles définis par les contraintes du problème (3.36).
Les groupes de points contraints et non contraints sont représenté dans la figure (3.9)

Pour résoudre efficacement le problème, il est astucieux de prendre en compte la situation des points au regard des contraintes vérifiées ou non.

3.6.1.7 Structure et détails de SimpleSVM

L'objectif de tout algorithme SVM est double : il faut d'une part arriver à répartir l'ensemble d'apprentissage dans ces trois groupes, puis une fois la répartition connue, résoudre le problème. Il s'avère que cette seconde phase est relativement plus facile. Supposons que l'on connaisse la répartition des points (I_s, I_C et I_0 donnés) : les contraintes d'inégalités ne sont plus nécessaires (elles sont contenues implicitement dans la définition des trois ensembles de points). Seules les valeurs des α_i pour $i \in I_s$ demeurent inconnues, car par définition $\alpha_i = C$ pour $i \in I_C$ et $\alpha_i = 0$ pour

$i \in I_O$. La valeur des α_i pour $i \in I_s$ est ainsi donnée par la solution du problème d'optimisation suivant :

$$\begin{cases} \max\limits_{\alpha_w \in \mathbb{R}^{|I_w|}} & \frac{1}{2}\alpha'_w G_{(w,w)} - (\mathbf{1}'_w - C\mathbf{1}'_C G_{(C,w)})\alpha_w \\ avec & \alpha'_w y_w + C\mathbf{1}'_C y_C = 0 \end{cases} \tag{3.41}$$

Il est ici important de noter que la dimension de ce problème est égale au cardinal de l'ensemble I_w (qui peut être bien inférieur à n, la dimension initiale du problème). Les conditions de Kuhn Kurush et Tucker nous donnent le système à résoudre pour obtenir la valeur des coefficients α_w encore inconnus :

$$\begin{pmatrix} G_{(w,w)} & y_w \\ y_w & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{1}'_w - C\mathbf{1}'_C G_{(C,w)} \\ C\mathbf{1}'_C y_C \end{pmatrix} \tag{3.42}$$

Si la solution de ce système contient une composante violant les contraintes (une composante négative ou supérieure à C), elle prouve que la répartition initiale des points I_w, I_0 et I_C est fausse. Cette composante violant les contraintes doit être enlevée de I_w pour passer dans $I_0 ou I_C$. Si maintenant toutes les composantes de la solution α vérifient les contraintes, il n'est pas encore certain que nous ayons trouvé l'optimum du problème initial. Il faut encore vérifier que les contraintes de positivité sont respectées pour les multiplicateurs de Lagrange associés. Alors, nous allons vérifier que pour tous les points de I_0, $G_{(0,.)}\alpha + by_0 - 1_0 > 0$, et que pour tous les points de I_C, $G_{(C,.)}\alpha + by_C - 1 < 0$. Si tel n'est pas le cas, le point violant le plus les contraintes devra changer d'appartenance et quitter I_0 ou I_C pour passer dans I_w. Nous venons là de donner le principe de l'algorithme. Il s'agit d'un algorithme itératif qui va, à chaque itération, ajouter ou enlever un seul point de l'ensemble I_w. Nous verrons qu'à chaque itération le coût décroît strictement, garantissant ainsi la convergence de la méthode. Enfin, puisqu'entre chaque itération les matrices $G(w,w)$ ne diffèrent que d'une ligne et d'une colonne, il est possible de calculer la nouvelle solution directement à partir de l'ancienne, réduisant ainsi la complexité de chaque itération de $\mathcal{O}(|I_w|^3)$ à $\mathcal{O}(|I_w|^2)$.

3.6.1.8 Méthode des contraintes actives

Une autre manière d'introduire l'algorithme est de le présenter comme une méthode de gradient projeté. Le principe consiste à partir de l'intérieur du domaine des contraintes admissibles et d'y rester en recherchant à chaque itération une direction de descente admissible qui nous amène sur la frontière du domaine. Une telle direction peut être obtenue en projetant le gradient sur cette frontière Minoux [1983] ([Minoux 1983]). C'est cette projection qui va nous amener à faire passer un point de I_w vers I_0 ou I_C.

3.6.1.9 Activation d'une contrainte : projection

Si l'on active une contrainte, cela veut dire que l'on fixe à α l'une des bornes (0 ou C) et par consequent que le point correspondant est inactif dans le classifieur. Supposons que l'on connaisse un point admissible $\widetilde{\alpha}_{old} = (\alpha_{old}, \lambda_{old})$. L'optimum non contraint $\widetilde{\alpha}^*$ est donne par l'equation (3.42) :

$$\widetilde{\alpha}^* = H^{-1}\widetilde{e} \quad avec \quad H = \begin{pmatrix} G_{(w,w)} & y_w \\ y_w & 0 \end{pmatrix} \quad et \quad \widetilde{e} = \begin{pmatrix} \mathbf{1}'_w - C\mathbf{1}'_C G_{(C,w)} \\ C\mathbf{1}'_C y_C \end{pmatrix} \tag{3.43}$$

On choisit la direction de descente $\mathbf{d} = \widetilde{\alpha}^* - \widetilde{\alpha}_{old}$, celle du gradient, qui garantit que le coût va décroître. Afin de rester dans le domaine admissible $0 < \alpha < C$, nous allons recenser toutes les composantes de α^* violant les contraintes, et rechercher le pas de descente t^* minimisant le coût tout en nous maintenant dans le domaine admissible, c'est-à-dire

$$t^* = \max_t \ (\ \alpha \ = \ \alpha_{old} \ + \ t\mathbf{d} \ \mid \ 0 < \alpha < C) \tag{3.44}$$

Pratiquement, le pas de descente est donné directement par la recherche du minimum suivant :

$$t^* = min \ (\ \frac{\alpha_{old}}{\mathbf{d}}, \ \ \frac{C \ - \ \alpha_{old}}{\mathbf{d}} \ ) \tag{3.45}$$

Cela revient à projeter sur la frontière de la boîte et à éliminer de l'ensemble I_w la variable correspondant au plus grand pas possible. La nouvelle solution est alors :

$$\widetilde{\alpha}_{new} = \ \widetilde{\alpha}_{old} \ + \ t^*\mathbf{d} \tag{3.46}$$

Proposition(Décroissance du coût)
Le coût diminue strictement à chaque itération (étape 3.1 de l'algorithme (4)).
Démonstration
Cette propriété découle de la convexité du coût et nous donnons ici le détail. Considérons le coût :

$$q(\alpha) = \frac{1}{2}\alpha^T H\alpha \ - \ \widetilde{\alpha}_{old}^T\widetilde{e}$$

la direction de descente d est celle du gradient, elle vérifie :

$$\begin{aligned} \mathbf{d} &= \widetilde{\alpha}^* \ - \ \widetilde{\alpha}_{old} \\ &= H^{-1}\widetilde{e} \ - \ \widetilde{\alpha}_{old} \end{aligned}$$

et donc

$$H\mathbf{d} = \widetilde{e} \ - \ H\widetilde{\alpha}_{old}$$

On a alors :

$$q(\widetilde{\alpha}_{new}) - q(\widetilde{\alpha}_{old}) = q(\widetilde{\alpha}_{old} + t\mathbf{d} \ ) - q(\widetilde{\alpha}_{old})$$

$$q(\widetilde{\alpha}_{new}) = \; q(\widetilde{\alpha}_{old} + t\mathbf{d}) = \frac{1}{2}(\widetilde{\alpha}_{old} + t\mathbf{d})^T H(\widetilde{\alpha}_{old} + t\mathbf{d}) \; - \; (\widetilde{\alpha}_{old} + t\mathbf{d})^T \widetilde{e}$$

$$q(\widetilde{\alpha}_{old}) = \frac{1}{2}\widetilde{\alpha}_{old}^T H \widetilde{\alpha}_{old} \; - \; \widetilde{\alpha}_{old}^T \widetilde{e}$$

$$q(\widetilde{\alpha}_{old} + t\mathbf{d}\;) - q(\widetilde{\alpha}_{old}) = \frac{1}{2}(\widetilde{\alpha}_{old} + t\mathbf{d})^T H(\widetilde{\alpha}_{old} + t\mathbf{d}) \; - \; (\widetilde{\alpha}_{old} + t\mathbf{d})^T \widetilde{e}$$

$$-(\frac{1}{2}\widetilde{\alpha}_{old}^T H \widetilde{\alpha}_{old} \; - \; \widetilde{\alpha}_{old}^T \widetilde{e})$$

$$= \frac{1}{2}(\widetilde{\alpha}_{old} + t\mathbf{d})^T H(\widetilde{\alpha}_{old} + t\mathbf{d}) \; - \; \underline{\widetilde{\alpha}_{old}^T \widetilde{e}} - \frac{1}{2}\widetilde{\alpha}_{old}^T H \widetilde{\alpha}_{old} \; + \; \underline{\widetilde{\alpha}_{old}^T \widetilde{e}} - (t\mathbf{d})^T \widetilde{e}$$

$$= \frac{1}{2}(\widetilde{\alpha}_{old} + t\mathbf{d})^T H(\widetilde{\alpha}_{old} + t\mathbf{d}) \; - \; \frac{1}{2}\widetilde{\alpha}_{old}^T H \widetilde{\alpha}_{old} \; - (t\mathbf{d})^T \widetilde{e}$$

$$= \underline{\frac{1}{2}\widetilde{\alpha}_{old}^T H \widetilde{\alpha}_{old}} + \frac{1}{2}\widetilde{\alpha}_{old} H t\mathbf{d} + \frac{1}{2}(t\mathbf{d})^T H \widetilde{\alpha}_{old} + t\mathbf{d} \; - \; \underline{\frac{1}{2}\widetilde{\alpha}_{old}^T H \widetilde{\alpha}_{old}} \; - t\mathbf{d}^T \widetilde{e}$$

$$= \frac{1}{2}\widetilde{\alpha}_{old} H t\mathbf{d} + \frac{1}{2} t\mathbf{d}^T H(\widetilde{\alpha}_{old} + t\mathbf{d}) \; - t\mathbf{d}^T \widetilde{e}$$

$$= \frac{1}{2}\widetilde{\alpha}_{old} H t\mathbf{d} + \frac{1}{2} t\mathbf{d}^T H \widetilde{\alpha}_{old} + \frac{1}{2} t\mathbf{d}^T H t\mathbf{d} \; - t\mathbf{d}^T \widetilde{e}$$

$$= \frac{1}{2} t^2 \mathbf{d}^T H \mathbf{d} + tH\widetilde{\alpha}_{old}^T \mathbf{d} - t\widetilde{e}^T \mathbf{d}$$

$$q(\widetilde{\alpha}_{new}) - q(\widetilde{\alpha}_{old}) = \frac{1}{2} t^2 \mathbf{d}^T H \mathbf{d} + t(\underline{H\widetilde{\alpha}_{old} - \widetilde{e}})^T \mathbf{d}$$

$$= \frac{1}{2} t^2 \mathbf{d}^T H \mathbf{d} - t\mathbf{d}^T H \mathbf{d}$$

$$= (\frac{1}{2} t^2 - t)\mathbf{d}^T H \mathbf{d} \quad < \; 0$$

car le premier terme est négatif pour $0 < \; t \; < 2$ et par définition, le pas est compris entre 0 *et* 1. Par ailleurs le second est positif puisque la matrice H est définie positive et que le vecteur $\mathbf{d}$ n'a pas toutes ses composantes nulles sauf la dernière.

3.6.1.10 Désactivation d'une contrainte

Désactiver une contrainte signifie permettre au α concerné de prendre une valeur libre. Cela revient à faire passer le vecteur correspondant dans le groupe I_w, des vecteurs supports candidats. Cela s'avère indispensable lorsque le vecteur $\widetilde{\alpha}^*$ calculé précédemment est admissible (lorsque $t = 1$). Dans ce cas, pour qu'il existe encore une direction de descente (pour que le coût continue à décroître) il faut trouver soit

- un point de I_0 pour lequel $y_i(f(x_i) + b) - 1 < 0$,
- soit un point de I_C pour lequel $y_i(f(x_i) + b) - 1 > 0$.

Ces contraintes correspondent au fait que le gradient est convexe au point concerné et qu'il est possible de diminuer le coût en entrant dans le domaine admissible. De manière intuitive, on vérifie dans le primal que la solution candidate classe bien les points du jeu d'apprentissage. Si ce n'est pas le cas, alors il faut réajuster la solution en ajoutant au groupe des actifs un des points mal " classé " (au sens de I_w, I_0 *et* I_C). L'algorithme (4) résume le principe des itérations de l'algorithme du simpleSVM. Nous allons reprendre la preuve de convergence de l'algorithme donnée dans Vishwanathan et al. [2003]. L'algorithme (4) s'arrête lorsque le vecteur $\widetilde{\alpha}^*$

Algorithme 4 SimpleSVM

1 (I_w, I_0 *et* I_C). ← *initialiser*

2 (α, λ) ← *resoudre le Systeme Sans Contraintes* (I_w)

si $min\alpha_w < 0$ *ou* $max\alpha_w > C$

alors

3.1 projeter α *à l'intérieur du domaine admissible*

3.2 transférer le point associé de I_w *vers* I_0 *ou* I_C

aller à 2

finsi

4 *calculer les multiplicateurs de Lagrange* $\eta(I_C) et \nu(I_0)$

si $min\nu(I_0) < e$ *ou* $min\eta(I_C) < e$

alors

5 *transférer un point associé de* I_0 *ou* I_C *vers* I_w

aller à 2

sinon le minimum est atteint

finsi

calculé précédemment est admissible et tous les points de I_0 *et* I_C vérifient leurs contraintes. Il n'existe alors plus de direction de descente admissible. Comme la solution dépend de la répartition des points dans les ensembles I_w, I_0 *et* I_C, qu'il n'existe qu'un nombre fini de points donc de combinaisons et que le coût associé à l'algorithme décroît strictement à chaque itération, l'algorithme ne peut pas boucler et va atteindre la solution globale en un temps fini.

3.6.1.11 Réglage des hyper-paramètres

Nous cherchons à fournir des méthodes capables de trouver elle-même les hyper-paramètres les plus adaptés aux données présentées. Compte tenu de l'importance de cette question pour rendre le potentiel des SVM plus accessible, beaucoup de travaux ont été menés dans cette direction. La plupart des méthodes reposent sur une évaluation externe de la performance telle que la validation croisée. D'autres méthodes utilisent une mesure incorporée à l'algorithme lui-même (pour les méthodes à noyaux, voir Argyriou et al [2006] [Argyriou 2006], Micchelli and Pontil [2005]). A côté de ces méthodes empiriques pour le contrôle des paramètres, les chemins de régularisations ont été développés et largement étudiés ces dernières années Bach et al [2005] [Bach 2005], Gunter and Zhu [2005], Hastie et al [2004] [Zhu 2003]. Les chemins de régularisation offrent une façon élégante et efficace d'accéder à toutes les solutions optimales d'un problème au regard d'un paramètre. Dans le cas de la classification binaire qui nous intéresse ici, nous cherchons le meilleur compromis entre la performance en apprentissage et la régularité de la solution et ce avec comme objectif d'obtenir la meilleure performance en généralisation. Toutefois, connaître l'ensemble du chemin de régularisation n'est pas suffisant car il faut ensuite choisir la solution à utiliser. Pour cela nous proposons d'évaluer au fur et à mesure de l'avancement sur le chemin la performance en généralisation de la solution proposée. Cela permet d'éviter une évaluation externe ou une approximation Wahba [1999] [Wahba 1998]. Contrairement aux approches existantes du chemin de régularisation pour les SVMs, nous proposons de commencer par la solution vide (aucun vecteur support plutôt que tous les points vecteurs supports). Puisque nous évaluons notre solution à chaque étape, nous pouvons arrêter l'apprentissage lorsque l'évaluation indique une dégradation des performances en généralisation. Ainsi il n'est pas nécessaire de calculer les solutions qui prennent beaucoup trop de vecteurs supports (et qui vont à l'encontre de l'idée de parcimonie qui est un des atouts maîtres des SVMs). Éviter de calculer ces solutions permet aussi de ne pas calculer l'ensemble de la matrice de Gram et permet donc d'envisager de traiter des problèmes de plus grande dimension. Nous allons dans un premier temps présenter les chemins de régularisation pour les SVMs.

3.6.1.12 Chemins de régularisation

Le chemin de régularisation est constitué par l'ensemble des solutions possibles pour un problème donné en fonction d'un paramètre de compromis. Pour les SVM classiques, ce paramètre est C. A chaque valeur de C, le SVM définit les trois groupes I_w, I_0 *et* I_C et les valeurs des coefficients α correspondants. Nous rappelons sur la figure (3.9)la signification de ces groupes, à savoir que :

– I_0 contient les indices des points se trouvant bien classés et du bon côté de la

marge ($y_i f(x_i) + b > 1$) et par conséquent ont un coût nul (les SVM classiques utilisant le coût charnière),
- I_w contient les indices des points se trouvant sur la marge ($y_i f(x_i) + b = 1$) et se trouvant à la charnière entre un coût nul et un coût non nul,
- I_C contient les indices des points se trouvant du mauvais côté de la marge ($y_i f(x_i) + b < 1$) et ayant un coût non nul.

Parcourir l'ensemble de ces solutions en faisant varier C produit le chemin complet de régularisation.

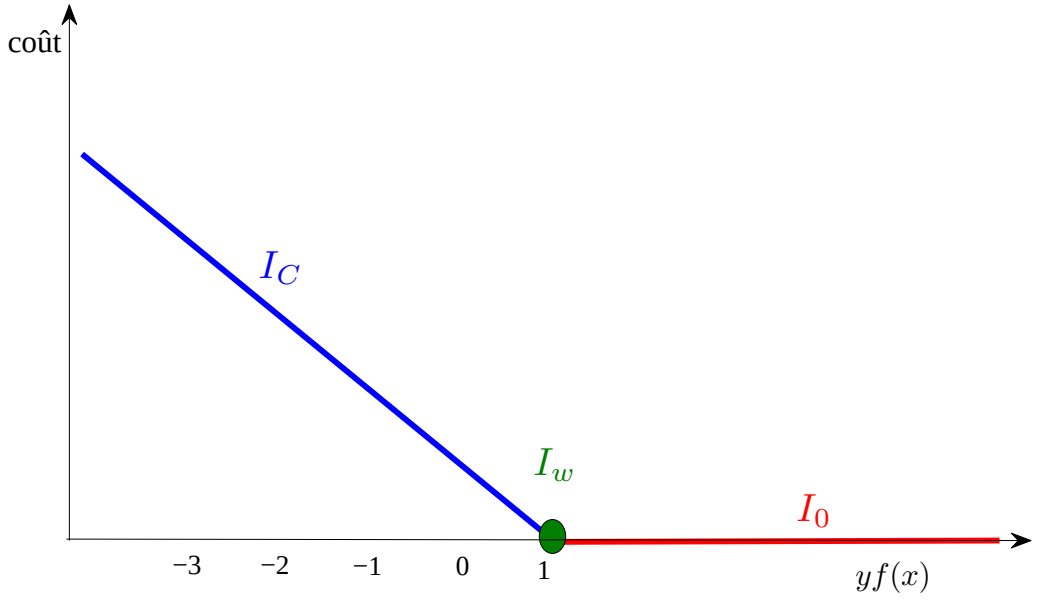


FIG. 3.9 – Fonction de coût charnière. Les groupes de points sont indiqués par rapport à leur coût. I_0 correspond à un coût nul (points bien classés),I_w correspond à un coût nul aussi mais correspond au point de coupure et I_C correspond à un coût non nul (points du mauvais côté de leur marge).

3.6.1.13 Résolution du système linéaire

Le coeur de l'algorithme et aussi la partie la plus gourmande en temps de calcul est la résolution du système de la forme :

$$\begin{pmatrix} G & y \\ y & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \lambda \end{pmatrix} = \begin{pmatrix} e \\ f \end{pmatrix} \tag{3.47}$$

Deux stratégies différentes peuvent être utilisées pour résoudre le système linéaire de type $Gx = b$. Puisque nous savons la matrice G définie positive, la factorisation de Cholesky [http ://www.seas.ucla.edu/chol.pdf] est la plus efficace. Mais dans le cas où la matrice G est singulière il est préférable d'utiliser la factorisation QR [http ://perso.univ rennes1.fr/PolyF04cours.pdf].

3.6.1.14 Validation croisée

La validation croisée est la méthode la plus populaire de réglage des hyper-paramètres des méthodes d'apprentissage. Cela consiste à discrétiser les espaces de chacun des hyper-paramètres à régler puis de teste successivement toutes les combinaisons possibles. L'apprentissage s'effectue sur un sous-ensemble de la base et le test sur les données restantes, que l'on appelle ensemble de validation. le classifieur choisi est à son tour évalué sur un ensemble de test indépendant. Il existe divers variantes, la plus commune consiste à séparer la base d'apprentissage en k ensembles et de tester k fois la sélection d'hyperparamètres : l'apprentissage se fait sur $k-1$ sous ensembles et le test sur le k^{ime} et ce successivement jusqu'à ce que chaque sous-ensemble ait servi de base de validation. La moyenne des performances en validation sert à évaluer la qualité des hyper-paramètres. L'inconvénient majeur de cette méthode est le temps de calcul. Pour p hyper-paramètres à régler pouvant chacun prendre v valeurs, le nombre d'apprentissages à effectuer est $k(v^p)$ **Conclusion** Les SVM sont connus comme étant performants mais lents. Les algorithmes sont plus pénalisés par le nombre d'observations, c'est-à-dire le nombre de vecteurs supports potentiels, que par le nombre de variables. Néanmoins, des versions performantes des algorithmes permettent de prendre en compte des bases de données volumineuses pour des temps de calcul acceptables. Ici nous montrons qu'il est possible de résoudre le problème des SVM dans des temps tout à fait raisonnables. Dans cet article nous avons présenté les aspects liés à la mise en oeuvre d'un algorithme rapide.Nous avons donné une vision générale et le fondement mathématique des SVM. Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des classes de données. Nous avons exposé les cas linéairement séparable et les cas non linéairement séparables qui nécessitent l'utilisation de fonctions noyaux (kernel) pour changer d'espace. Cette méthode est applicable pour des taches de classification a deux classes, mais il existe des extensions pour la classification multi-classe. Les principaux résultats de cette étude sont d'une part une boîte à outils disponible qui fournit des méthodes rapides pour SVM et d'autre part des expérimentations montrant les capacités de cet algorithme en terme de temps de calcul et de complexité. En outre, le fait de pouvoir initialiser la méthode à partir d'une solution candidate arbitraire permet de facilement régler les hyper paramètres par des méthodes de validation croisée. De même il est envisa-

geable de prendre en compte l'aspect multi-classes directement. D'un point de vue plus général, nous nous intéressons ici à la question de la complexité dans les méthodes d'apprentissage et les SVM en particulier. Notre objectif est de profiter des particularités du problème des SVM pour diminuer la complexité de la résolution à des ordres allant de $O(n)$ à $O(n^2)$. Les SVMs représentent aujoud'hui l'une des méthodes les plus utlisées grâce à leur pouvoir de généralisation. Dans le prochain chapitre nous verrons comment introduire la notion de rejet.

Chapitre 4

Optimisation du coût de la classification

4.1 Introduction

Lors de la conception d'un système de reconnaissance de formes, l'objectif principal est de minimiser les erreurs de classification. Le principe de l'option de rejet est d'enrichir ce fonctionnement en permettant au classifieur de ne pas répondre. Dans ce cas on dit que la forme est rejetée. Le principal intérêt du rejet est de pouvoir éviter une erreur de classification. Tant que les classifieurs ne seront pas parfaits, le rejet permettra de détecter et de gérer les erreurs potentielles. Ainsi, il est important de différentier deux types de rejet, correspondant à deux catégories de données délicates.

- Le rejet d'ambiguïté consiste, comme son nom l'indique, à filtrer les exemples ambigus.[Chow 1970].
- Le second type de rejet concerne les données aberrantes qui ne correspondent à aucune des classes du problème.

En effet, malgré les performances et la capacité de généralisation des classifieurs SVMs, les erreurs de classification ne peuvent pas être complètement éliminés. Dans certaines applications, elles peuvent produire de graves conséquences désignées par coût de la classification. L'objectif poursuivi à travers ce chapitre est de modifier l'algorithme SVM exposé dans le chapitre précédent afin de minimiser la coût de la classification et ce, en introduisant une troisième décision à la sortie du classifieur, qui consistera à rejeter les exemples incertains dont la probabilité d'appartenance à l'une des classes est autour de 0.5.

4.2 Etat de la question

Le rejet est un problème général de reconnaissance de formes. Son intérêt est visible dans plusieurs domaines : diagnostic [Boatas 2000], reconnaissance du montant sur un chèque [Kaufmann 2000, Kwok 1999, Knerr 1997, Lethelier 1995] et la reconnaissance des codes postaux sur le courrier [Liu 2004]. Dans ces cas, il est important de savoir quand la classification peut être mauvaise car une erreur peut avoir un coût important. Ces formes rejetées peuvent ensuite faire l'objet soit d'une vérification manuelle comme c'est le cas dans les deux exemples précédents, soit d'une classification par un autre classifieur plus performant. Beaucoup de stratégies de rejet sont proposées, utilisant différentes modélisations, différentes sources d'information et différents classifieurs. La principale architecture, la plus ancienne d'après la littérature est basée sur des seuils de décision. Ensuite ce formalisme a été étendu pour s'appliquer à d'autres contextes. Pour améliorer la fiabilité des classificateurs, de nouveaux algorithmes d'apprentissage ont été introduits. Parmi ces méthodes, on cite la *conformal prediction* qui consiste à déterminer les niveaux de confiance. Ainsi, tout exemple classé avec un niveau de confiance inférieur à un seuil prédéfini doit être rejeté. Ceci a également motivé les chercheurs à introduire une option de rejet dans classifieurs. Tout exemple dont la probabinité d'appartenance à une classe est proche de 1/2 est rejeté car jugé ambigu. Cette approche a été introduite depuis 1957 à travers la publication de C.K.Chw [Chow 1957] puis un autre travail [Chow 1970] sur le compromis erreur-rejet. Cette approche n'a pas bénéficié d'un grand intérêt de la part de la communauté du "*machine learning*". Nous notons cependant quelques remarquables travaux [Kwok 1999, Tortorella 2004] où les auteurs proposèrent une méthode de rejet explicitement pour les SVMs. La zone de rejet est délimitée par deux hyperplans parallèles à l'hyperplan optimal. Les seuils de décision sont calculés sur les scores du SVM. En 2002, Un algorithme d'apprentissage SVM avec rejet fut proposé par [Fumera 2002] où la zone de rejet est délimitée durant la phase d'apprentissage. Néanmoins, cette méthode manque de convexité et impose un calcul très lourd. D'autres classifieurs SVMs avec option de rejet et basés sur la fonction de pertes [1] sont récemment proposés dans [Bartlett 2008, Herbei 2006, Wegkamp 2010]. La méthode de rejet que nous utilisons dans cette thèse est basée sur l'interprétation probabilistique des SVMs introduite par Yves Grandvalet en 2006 [Grandvalet 2006]. Nous introduisons une fonction de perte à double charnière ("*double hinge loss*") dédiée à un problème de classification avec option de rejet généralisant celle proposée par [Bartlett 2008] à une classification asymétrique. L'approche probabiliste nous permet d'aboutir à une règle de décision donnant de meilleurs résultats :

[1] *double hinge loss*

4.2.1 Les deux natures de rejet

Les systèmes de reconnaissance avec option de rejet ont la capacité de ne pas classer une forme afin d'éviter de faire des erreurs. Ces erreurs peuvent avoir principalement deux causes qui correspondent à deux états du classifieur :

- le classifieur hésite entre deux classes possibles, il est indécis.
- le classifieur n'a pas appris à reconnaître la forme à classer, il est surpris.

A ces deux types d'erreurs sont associés deux types de rejet : le rejet d'ambiguïté et le rejet de distance. Plusieurs auteurs confirment cette vision des choses [Dubuisson 1993, Frélicot 1995] même si cette différenciation n'est pas faite dans toutes les publications sur ce domaine. De plus, ces mêmes publications expliquent qu'il faut des approches différentes pour apprendre ces deux natures de rejet efficacement. C'est pourquoi il est important de savoir avant toute chose à quel type de rejet l'option sera confrontée. Ces deux natures de rejet correspondent à des contextes d'utilisation différents. Par exemple certaines taches égigent un minimum de confusions (reconnaissance de chèques, de codes postaux, etc.). Dans ce cas on utilisera le rejet d'ambiguïté [Chow 1970, Dubuisson 1993, Fumera 2000] pour minimiser le risque d'erreur. Dans un autre contexte, le rejet de distance utilise des connaissances intrinsèques pour délimiter le domaine de validité d'un classifieur spécialisé en rejetant les formes n'appartenant pas aux classes apprises [Dubuisson 1993]. Le rejet de distance consiste alors à détecter quand une donnée ne correspond pas aux données d'apprentissage ce qui est proche de l'objectif de la détection de nouveauté [Markou 2003]. Ces deux types de rejet peuvent être utilisés (ensemble ou non) pour définir des cascades de classifieurs dans le but de réduire la complexité de la reconnaissance ou d'en améliorer les performances [Anquetil 1999, Milgram 2004].

4.2.2 Définition du rejet d'ambiguïté

Le but du rejet d'ambiguïté est d'accroître la fiabilité du classifieur en rejetant les formes pour lesquelles le classifieur a de fortes chances de faire une mauvaise classification, c'est à dire d'associer la forme à une mauvaise classe. Ces erreurs sont proches des frontières de décision car elles activent fortement au moins deux classes de façon équivalente. Un rejet d'ambiguïté peut être vu comme un refus de classification pour éviter une erreur de confusion. Une technique classique pour réaliser le rejet d'ambiguïté est de définir un couloir autour des frontières de décision. Toutes les formes à l'intérieur sont considérées comme des erreurs potentielles et sont donc rejetées. La figure (4.1) montre un exemple en deux dimensions de rejet d'ambiguïté.

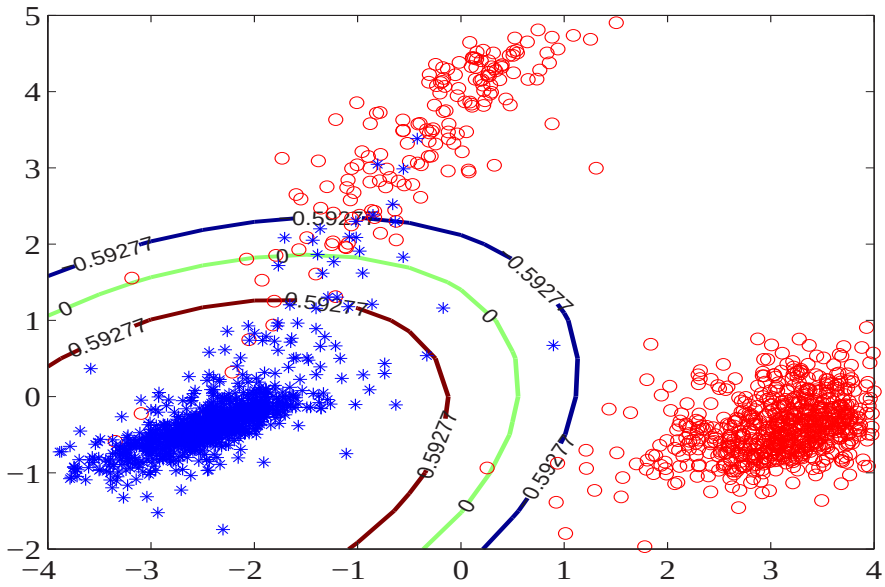


FIG. 4.1 – Cette figure montre un couloir de rejet délimité par les deux lignes $+0.59277$ et -0.59277. La classe positive est représentée par des étoiles bleues alors que la classe négative est représentée par des ronds rouges. La ligne 0 correspond à $f(x) = 0$ ou $P(Y = 1 \mid X = x) = 0.5$.

4.2.3 Définition du rejet de distance

Le rejet de distance est basé sur la délimitation des connaissances du classifieur. En effet un classifieur ne peut classer que ce qu'il a appris à classer. Il faut donc rejeter ce qui n'appartient à aucune classe apprise. Si une forme est trop éloignée des connaissances du classifieur il faut la rejeter. Une technique classique pour le rejet de distance est de déterminer la probabilité d'appartenir (ou le degré d'appartenance) aux données d'apprentissage.

4.2.3.1 Discussions sur le rejet d'ambiguïté et le rejet de distance

Ces définitions ont jusque là été naturellement guidées par les deux sources d'erreurs : la forme n'est pas reconnue soit parce qu'elle ne ressemble pas aux données d'apprentissage, soit parce qu'elle correspond fortement à au moins deux classes. Nous proposons d'ajouter à cette approche la nature des connaissances utilisées pour réaliser le rejet, des connaissances soit discriminantes soit intrinsèques. Le rejet d'ambiguïté est basé sur la notion de discrimination. En effet les erreurs

potentielles du classifieur sont liées aux capacités discriminantes du classifieur. Ces capacités changent d'un classifieur à l'autre, les frontières changent et donc les zones à rejeter changent. Le rejet de distance est très lié à la capacité descriptive des classifieurs. En effet l'efficacité du rejet de distance va dépendre des capacités du classifieur utilisé à faire la différence entre les données utilisées lors de l'apprentissage et les autres à rejeter. Les classifieurs basés sur une modélisation intrinsèque des connaissances comme les RBFN, les k-PPV. Ces deux natures de rejet ne sont pas confrontées aux mêmes difficultés d'apprentissage.

- Pour le rejet d'ambiguïté, les erreurs à rejeter sont fortement mélangées à des formes bien classées.
- Pour le rejet de distance, la zone à rejeter est quasiment infinie puisque qu'il faut rejeter tout ce qui n'est pas à accepter, que ce soit très différent ou peu différent.

4.2.3.2 Utilisation de seuils

Une première formulation du problème de rejet d'ambiguité a été proposée par Chow [Chow 1970]. Elle s'applique à des classifieurs probabilistes. Il y a rejet lorsque la probabilité *a posteriori* que la forme X appartienne à la meilleure classe est plus faible qu'un certain seuil T :

$$\max_i \mathrm{P}(i|X) < T \tag{4.1}$$

Grâce aux propriétés des classifieurs probabilistes, il démontre trois propriétés importantes (avec $T = 1 - t$) :

1. le taux d'erreur et le taux de rejet sont des fonctions monotônes de t.
2. t est une borne maximale pour le taux d'erreur.
3. t est un compromis différentiel entre le taux d'erreur et le taux de rejet.

$$\frac{dE(t)}{dR(t)} = -t \tag{4.2}$$

On en déduit le principe suivant : plus le seuil t est bas, plus il y a du rejet et moins il y a d'erreurs. La plupart des publications suivantes sur le rejet sont basées sur ce principe. En effet il s'applique à tous les classifieurs probabilistes. Chow a montré que cette option de rejet nécessite de faire un compromis entre le taux de rejet et le taux d'erreurs. Cette architecture a donc été beaucoup utilisée [Markou 2003]. Le rejet proposé par Chow est par construction, un rejet d'ambiguïté. Dans [Dubuisson 1993] l'approche de Chow est étendue pour y inclure un rejet de distance. Pour cela les auteurs fixent un seuil sur la probabilité $P(X)$ d'apparition de X. Si cette valeur est trop faible, la forme est rejetée. Plus récemment,

les travaux de Fumera et al [Fumera 2000] ont montré que l'option de rejet de Chow [Chow 1970] n'est optimale que lorsque les probabilités a *posteriori* d'appartenir à chaque classe sont bien estimées, ce qui n'est généralement pas le cas dans des contextes réels. C'est pourquoi les auteurs proposent d'utiliser un seuil par classe plutôt qu'un seul seuil global. La règle de rejet devient alors :

$$\max_i \mathrm{P}(i|X) < \mathrm{P}(k|X) < T_k \tag{4.3}$$

Pour décider du rejet, il faut considérer la classe qui maximise la probabilité d'appartenance de X et la comparer au seuil correspondant. Pour apprendre l'option de rejet il s'agit alors de donner une valeur à chaque seuil Ti. Les auteurs ont montré que pour chaque option de rejet de Chow définie par le seuil T, il existe un ensemble Ti de seuils au moins aussi efficaces. Si $A(T_1, ..., T_k)$ est une fonction d'efficacité du rejet, on a :

$$\forall\ T,\ \exists\ T_1, ..., T_k\ :\ A(T)\ \leq\ A(T_1, ..., T_k)$$

La difficulté est de trouver la combinaison de seuils qui permet d'opérer mieux qu'avec un seul seuil. Dans un classifieur deux propriétés peuvent influencer une option de rejet : sa capacité de classification et la disponibilité d'information sur la qualité de la classification.

4.2.3.3 Évaluation du rejet d'ambiguïté

Le but de l'évaluation d'une option de rejet d'ambiguïté est de vérifier si seules les erreurs de classification sont rejetées. Pour cela les éléments d'une base de teste D_E de N_{Tot}, exemples à reconnaître sont triés en trois groupes :

- exemples acceptés et correctement classés :N_{Corr} ;
- exemples acceptés mais mal classés :N_{Err} ;
- exemples rejetés : N_{Rej}.

A partir de ces valeurs qui dépendent d'un paramètres (ρ) d'apprentissage du rejet, on peut calculer quatre taux :

- le taux d'erreur $E(\rho)$ défini par l'équation (4.4) ;
- le taux de rejet $R(\rho)$ défini par l'équation (4.5) ;
- la performance $P_{erf}(\rho)$ défini par l'équation (4.6) ;
- la fiabilité $Fiab(\rho)$ défini par l'équation (4.7).

On peut remarquer que ces taux sont liés par les équations (4.8) et (4.9).

$$E(\rho) = \frac{N_{Err}}{N_{Tot}} \tag{4.4}$$

$$R(\rho) = \frac{N_{Rej}}{N_{Tot}} \tag{4.5}$$

$$P_{erf}(\rho) = \frac{N_{Corr}}{N_{Tot}} \tag{4.6}$$

$$Fiab(\rho) = \frac{N_{Corr}}{N_{Tot} + N_{Err}} \tag{4.7}$$

$$R(\rho) + E(\rho) + P_{erf}(\rho) = 1 \tag{4.8}$$

$$Fiab(\rho) = \frac{P_{erf}}{1 - R(\rho)} \tag{4.9}$$

Il s'agit de minimiser en même temps le taux d'erreur et le taux de rejet. Autrement dit, maximiser la performance globale. A chaque option de rejet apprise, correspond un point de fonctionnement $[E(\rho), R(\rho)]$ qui dépend de l'architecture choisie, du type des classifieurs et surtout des paramètres (ρ) d'apprentissage du rejet. Par exemple pour l'option de rejet de Chow ces taux dépendent de la valeur du seuil T (voir l'équation 4.1). L'optimisation du taux de rejet et du taux d'erreur est un compromis appelé : compromis $Erreur - Rejet$. Le compromis optimum est étudié dans la publication de Chow [Chow 1970] mais il se limite au classifieur Bayesien idéal. Plusieurs articles comme [Fumera 2000] ont étudié ce compromis dans le cas général.

4.2.4 Le compromis erreur-rejet (ER)

Depuis les travaux de Chow sur le compromis erreur-rejet, les courbes de compromis ER sont souvent utilisées pour évaluer les performances des classifieurs avec option de rejet. Ces courbes sont construites en faisant varier les paramètres (ρ) d'apprentissage et en mesurant pour chaque valeur les taux $E(\rho)$ et $R(\rho)$ pour construire la courbe correspondante. Cette courbe permet de choisir le point de fonctionnement du rejet qui correspond le mieux au besoin de l'application cible. La figure 4.2 présente un exemple type de courbe ER. Notons que la pente de cette courbe représentant la variation de l'erreur sur la variation du rejet est aussi un paramètre indicatif des performances du classifieur. Une pente très raide signifie qu'un grand pourcentage des exemples rejetés seraient mal classés par le classifieur. Ceci sera mis en évidence dans nos prochains chapitres. Il est possible de donner un coût de l'erreur $(C_E > 0)$, un coût du rejet $(C_R > 0)$, généralement $(C_E > C_R)$. On peut alors tracer la courbe du coût total en fonction des paramètres (ρ) :

$$C(\rho) = C_E.E(\rho) + C_R.R(\rho) \tag{4.10}$$

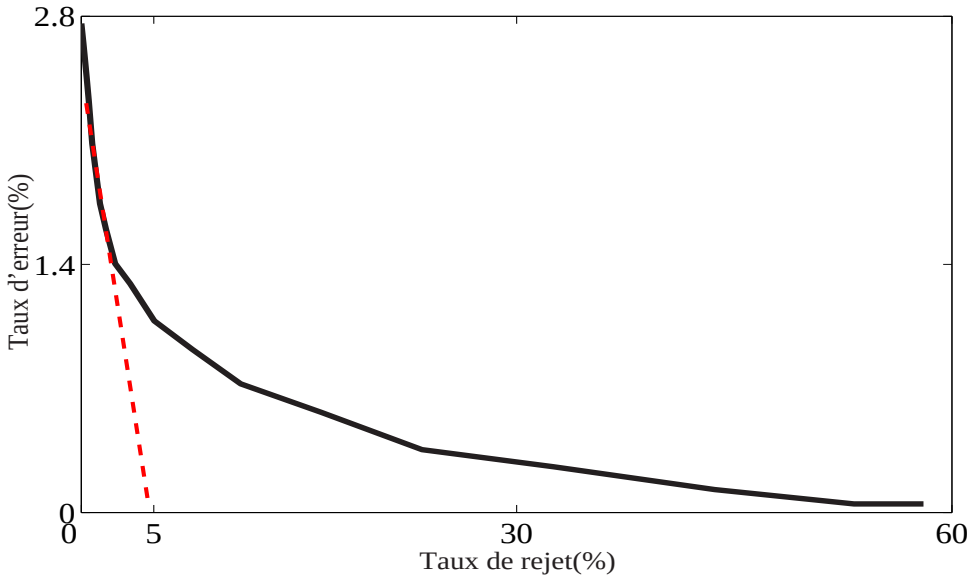


FIG. 4.2 – Courbe de compromis erreur-rejet

4.2.5 coût du rejet

Soient deux classes C_1 *et* C_2. On considère le problème de classification binaire avec les deux contraintes suivantes :

$$\begin{cases} P(D_1/C_2) < e_{12} \\ P(D_2/C_1) < e_{21} \end{cases} \quad \text{où } e_{12} \textit{ et } e_{21} \in [0\ 1] \tag{4.11}$$

avec $P(D_i/C_j)$ la probabilité de décider la classe i conditionnellement à la classe j et e_{ij} la borne supérieure pour la probabilité $P(D_i/C_j)$. Le problème à résoudre diffère selon qu'il existe ou non au moins une partition (Z_1, Z_2) de $\mathbb{R}^n$ telle que les contraintes soient vérifiées. S'il existe au moins un couple (Z_1, Z_2), le problème consiste à rechercher celui qui minimise un critère, qui de façon générale est un coût $\overline{c}$ qui s'exprime selon

$$\overline{c} = c_{11}P(D_1/C_1)P_1 + c_{12}P(D_1/C_2)P_2 + c_{21}P(D_2/C_1)P_1 + c_{22}P(D_2/C_2)P_2 \tag{4.12}$$

où P_i est la probabilité *a priori* de la classe i.
En prenant $c_{ii} = 0, c_{12} = c_{21} = 1$, le critére $\overline{c}$ est égal à la probabilité d'erreur P_e. S'il n'existe pas de partition (Z_1, Z_2) telle que les contraintes soient vérifiées, il est nécessaire, pour satisfaire les contraintes, d'introduire une zone de rejet, dans laquelle aucune décision sur les réalisations n'est prise. Le problème consiste alors à rechercher la partition (Z_1, Z_2, Z_R) qui minimise la probabilité de rejet P_R.

4.3 Position du problème

Considérons un problème de classification binaire où chaque échantillon appartient à une parmi deux classes. Une fonction discriminante $f : \mathcal{X} \mapsto \mathbb{R}$ associe à chaque observation $x \in \mathcal{X}$ une des deux classes étiquetées par $+1$ ou -1. Si $f(x)$ est vue comme la représentation d'une probabilité conditionnelle $P = \mathbb{P}(Y = 1|X = x)$, on serait peu confiant en classifiant un exemple x de faible valeur $\mid f(x) \mid$ qui correspondrait à une probabilité P autour de $1/2$. L'approche utilisée dans ce travail, est d'assigner $sgn(f(x_i)) = +1$ ou -1 si $|f(x_i)|$ est supérieure à un seuil δ_i et de rejeter l'exemple x_i sinon. Lors d'une classification binaire, deux types d'erreurs sont toujours définis :

- FP : un Faux Positif est un exemple négatif envoyé par erreur dans la classe positif. à cette erreur, on attribut un côut C_n
- FN : un Faux Négatif est un exemple positif envoyé par erreur dans la classe négative. à cette erreur, on attribut un côut C_p.

Nous considérons ici un problème où certains exemples ne doivent pas être classés. Nous associons de plus un coût R_n à tout exemple négatif rejeté et un coût R_p à tout exemple positif rejeté. En considérant que touts les coûts sont strictement positifs. La décision $'d'$ du classifieur a l'option de rejeter les exemples dont la probabilité d'appartenance à chacune des deux classes est proche de 0.5. Cette troisième décision d'abstension est notée ®. Les coûts engendrés par chacune des décisions possibles sont récapitulés dans la table 4.1.

En considérant que touts les coûts sont strictement positifs

TAB. 4.1 – *Coût associé à chaque décision*

		y		$\mathrm{E}_y[d(\mathbf{x})]$
		$+1$	-1	
$d(\mathbf{x})$	$+1$	0	C_n	$C_n\,(1 - \mathrm{P}(Y = 1\|\mathbf{x}))$
	®	R_p	R_n	$R_p\,\mathrm{P}(Y = 1\|\mathbf{x}) + R_n\,(1 - \mathrm{P}(Y = 1\|\mathbf{x}))$
	-1	C_p	0	$C_p\,\mathrm{P}(Y = 1\|\mathbf{x})$

$$C_n > 0, \quad C_p > 0, \quad R_n > 0, \quad R_p > 0$$

En posant $P = \mathrm{P}(Y = 1|\mathbf{x})$, probabilité d'avoir un point positif et $1 - P$, la probabilité d'avoir un point négatif, les coûts relatifs à chaque décision est donné

par trois décisions associées aux fonctions de perte. le coût minimale est alors donné.

$$\mathcal{R}(d) = min\{C_p P,\ C_n(1-P),\ R_p P + R_n(1-P)\} \tag{4.13}$$

En tenant compte de la fonction de risque (4.13), nous pouvons déduire de la figure (4.3) que rejeter un échantillons n'a de sens que si et seulement si le point G est localisé au dessus du segment $[AB]$ où

$$P_G = \frac{C_n}{C_p + C_n}$$

est l'abscisse du point G. A partir de là, on estime que le rejet est valable si et seulement si :

$$C_p P_G > R_p P_G + R_n(1 - P_G)$$

En d'autres terme, si et seulement si

$$\frac{R_p}{C_p} + \frac{R_n}{C_n} < 1 \tag{4.14}$$

correspondant à $0 \leq r \leq 1/2$ dans les approches présentées dans [Bartlett 2008, Herbei 2006, Wegkamp 2010].

4.4 Règle de Bayes

A partir de la Figure 4.3, nous déduisons que le classifieur de Bayes d^* minimisant le risque $\mathcal{R}(d)$ peut être exprimé en utilisant deux seuils de probabilités :

$$P_+ = \frac{C_n - R_n}{C_n - R_n + R_p}\ , \tag{4.15}$$

$$P_- = \frac{R_n}{C_p - R_p + R_n}\ , \tag{4.16}$$

correspondant aux seuils symétriques $P_- = r$ and $P_+ = 1-r$ définis dans [Bartlett 2008, Herbei 2006, Wegkamp 2010] Comme la règle de décision de Bayes est définie à partir de probabilités condtionnelles (voir figure 4.4). Nous commençons d'abords par estimer les probabilités conditionnelles $\widehat{P}(Y = 1|X = x)$ afin de construire la règle de décision :

$$f^*(x) = \begin{cases} +1 & \text{if } \widehat{P}(Y = 1|X = x) > P_+\ , \\ -1 & \text{if } \widehat{P}(Y = 1|X = x) < P_-\ , \\ \text{®} & \text{sinon}\ . \end{cases} \tag{4.17}$$

où $f^*(x)$ correspond à la décision d^* minimisant le risque (4.13).

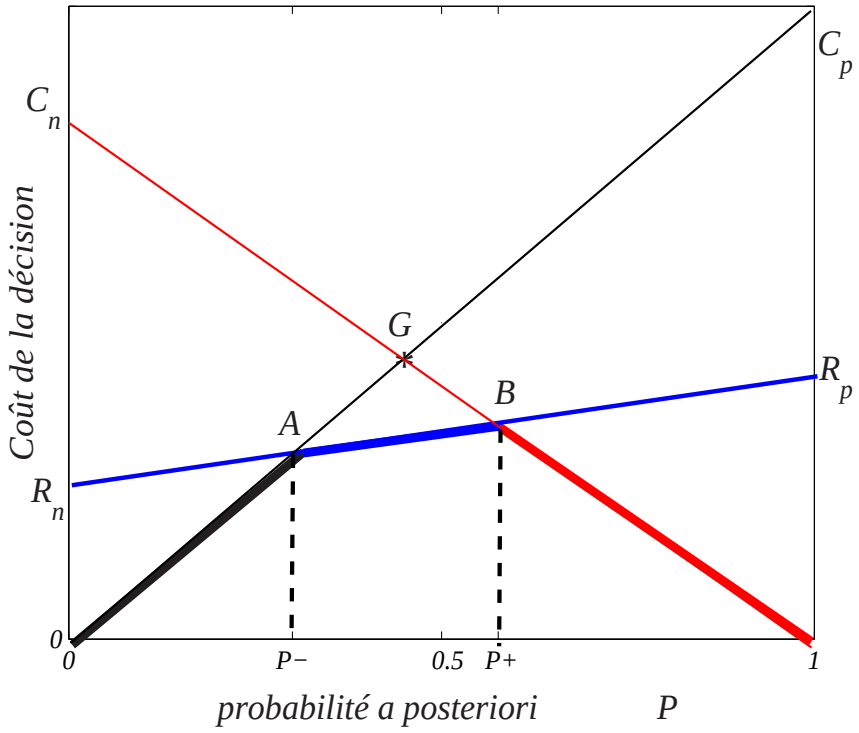


FIG. 4.3 – Coûts prévus en fonction des probabilités *a poteriori*. Nous avons porté en gras la courbe minimale dans chaque intervalle de probabilités.

Cas de coûts symétriques :

Partant de la fonction de risque (4.13), nous pouvons déduire une fonction de risque pour des coûts $C_p = C_n = 1$ [Bartlett 2008, Herbei 2006, Wegkamp 2010]. Sur la figure (4.5), les fonctions de perte pour chaque décision sont :

1. $d_p = C_p P = P$ (tracé noir)
2. $d_n = C_n(1-P) = 1-P$ (tracé rouge)
3. $d_R = R_p P + R_n(1-P) = r$ (tracé bleu)

A partir de la même figure, nous obtenons les probabilités correspondant aux seuils symétriques [Bartlett 2008, Herbei 2006, Wegkamp 2010].

$$P_- = r, \quad P_+ = 1-r \; et \; P_G = 1/2$$

le classifieur de Bayes d^* minimiserait alors le risque $\mathcal{R}(d)$ exprimé par :

$$\mathcal{R}(d) = min\{P, \; (1-P), \; r\} \tag{4.18}$$

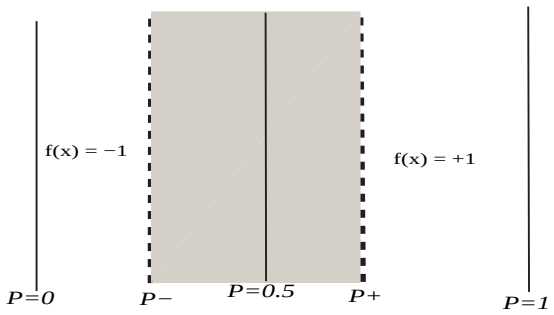


FIG. 4.4 – classification avec zones de rejet d'ambigüités (zone ambrée).

4.5 Classifieur SVM avec coût optimal

Dans ce qui suit, nous allons concevoir un classifieur généralisé c'est à dire avec des coûts supposés assymétriques. Afin de minimiser le risque structurel (4.13) dont le calcul s'avère très complexe, il est souvent remplacé par des fonctions de perte de substitution. Les plus utilisées sont :

- la fonction à charnière (*hinge loss*) fortement recommandée par V.Vapnick [Vapnik 1998] car elle conduit à une solution consistante [Grandvalet 2009, Wegkamp 2010].
- la fonction logistique qui offre la possibilité d'estimer les probabilités *a posteriori*

$$\widehat{P}(Y = 1|X = x) = 1/(1 + exp(-yf(x)))$$

ce qui conduirait au meilleur choix des seuils δ_i. Notons cependant que $\widehat{P}(Y = 1|X = x)$ doit être estimée uniquement au voisinage de P_+ et P_- voir equation (4.17).

4.5.1 Construction de la fonction de perte

Ce type de fonctions a été proposé pour la première fois en 2006 par R.Herbei *et al.* [Herbei 2006]. Elle a été reprise par P.Bartlett en 2008 [Bartlett 2008]. Dans les deux cas, la fonction est destinée pour une classification symétrique. En 2009, Y.Grandvalet *et al* [Grandvalet 2009] l'a généralisée pour une classification assymétrique. Une telle fonction de perte est convexe et linéaire par morceaux. Sur la figure 4.6, nous l'avons représentée pour les exemples positifs. Il est clair qu'elle est tengente à la fonction logistique aux points d'abscisses :

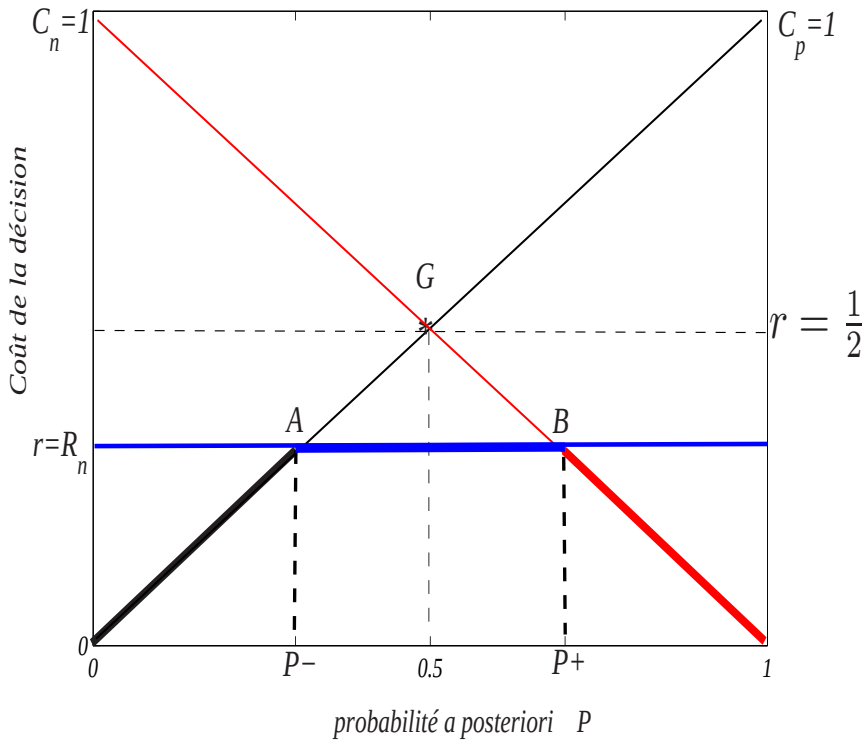


FIG. 4.5 – Coûts prévus en fonction des probabilités *a poteriori*. Nous avons porté en gras la courbe minimale dans chaque intervalle de probabilités pour un coût symétrique .

$$\delta_+ = log(P_+/(1-P_+))$$

$$\delta_- = log(P_-/(1-P_-))$$

Cette proposition retient l'avantage de la fonction charnière pour aboutir à une solution consistante mais aussi l'avantage de la fonction logistique pour estimer les probabilités P_+ et P_- espectivement aux points de tengence δ_+ et δ_-. Ce sont les seules probabilités dont nous avons besoin pour réexprimer la règle de décision :

$$f(x) = \begin{cases} +1 & \text{if } f(x) > \delta_+ \ , \\ -1 & \text{if } f(x) < \delta_- \ , \\ \text{®} & \text{sinon } . \end{cases} \tag{4.19}$$

Ces seuils sont symétriques dans les propositions faites dans [Bartlett 2008, Herbei 2006, Wegkamp 2010] où $\delta_+ = -\delta_- = \delta_o$ et la valeur recommendée de δ_o appartient à l'in-

tervalle $[r, 1-r]$. Pour exprimer la fonction à double charnières généralisée, nous

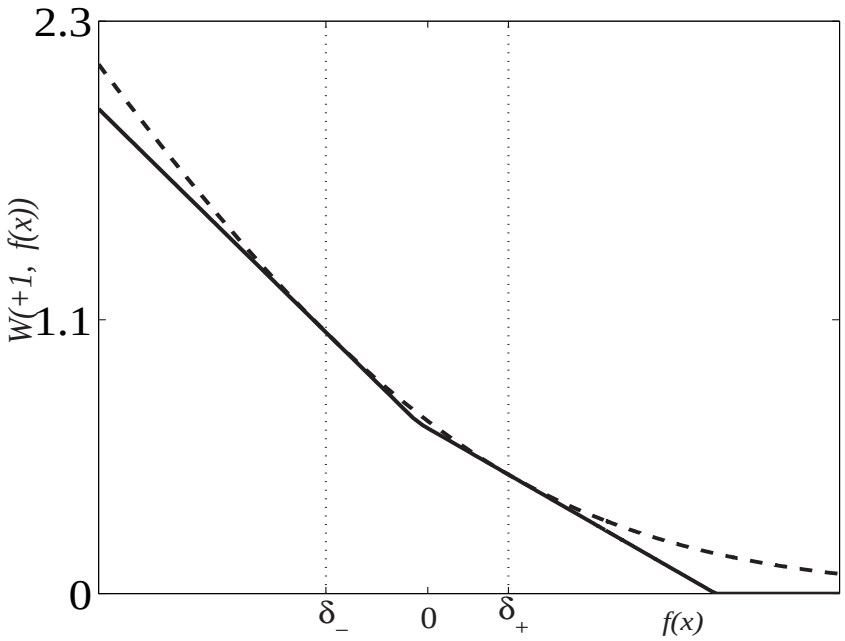


FIG. 4.6 – *Fontion de perte pour les exemples positifs ($y_i = +1$), avec $P_- = 0.35$ et $P_+ = 0.6$ (continu : fonction construite, interrompu : fonction logistique)*

considérons d'abord la fonction de perte logistique φ :

$$\varphi(y, f(x)) = log(1 + exp(-yf(x))) \ . \tag{4.20}$$

telle que $\varphi(+1, f(x)) = log(1+exp(-f(x)))$ pour les exemples positifs et $\varphi(-1, f(x)) = log(1 + exp(f(x)))$ pour les exemples négatifs. Pour exprimer la fonction de perte W, nous allons travailler sur la Figure 4.7 correspondant aux exemples positifs ($y_i = +1$). Soit $W = a_1 f(x) + g_1$ la première pente (de droite à gauche) de la fonction $W(+1, f(x))$ où

$$a_1 = \frac{d[\varphi(+1, f(x))]}{d[f(x)]}|_{\delta_+} = -(1 - P_+)$$

. Au point de tengence δ_+, nous avons $\varphi(+1, f(x)) = W(+1, f(x))$, alors

$$g_1 = -p_+ log(P_+) - (1 - P_+) log(1 - P_+) = H(P_+)$$

La seconde pente $W(+1, f(x))$ est donnée par $W = a_2 f(x) + g_2$ où

$$a_2 = \frac{d[\varphi(+1, f(x))]}{d[f(x)]}|_{\delta_-} = -(1 - P_-)$$

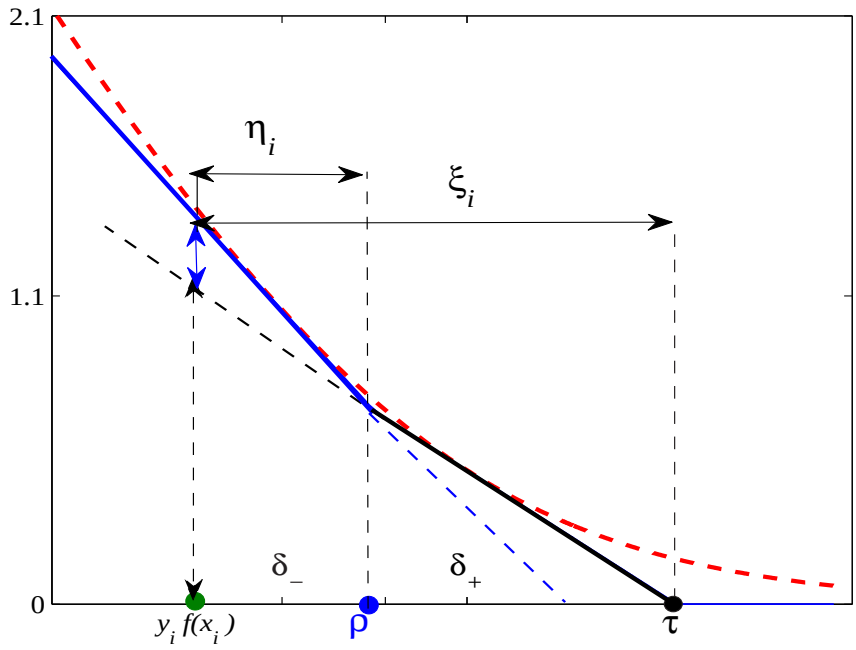


FIG. 4.7 – *Double hinge loss function for positive examples, with* $P_- = 0.4$ *and* $P_+ = 0.7$ *(solid : double hinge, red dashed : likelihood)*

et

$$g_2 = -P_-log(P_-) - (1 - P_-)log(1 - P_-) = H(P_-)$$

.

pour $a_1 f(x) + g_1 = 0$, nous avons

$$f(x) = \tau_+ = \frac{H(P_+)}{1 - P_+}$$

et pour $a_1 f(x) + g_1 = a_2 f(x) + g_2$, nous avons

$$f(x) = \rho_+ = \frac{H(P_-) - H(P_+)}{P_+ - P_-}$$

La fonction de perte W pour les exemples positifs peut être alors exprimée comme suit :

$$W(+1, f(x)) = \begin{cases} -(1 - P_-)f(x) + H(P_-) & \text{si } f(x) < \rho_+ \\ -(1 - P_+)f(x) + H(P_+) & \text{si } \rho_+ \leq f(x) < \tau_+ \\ 0 & \text{sinon,} \end{cases} \quad (4.21)$$

La même stratégie de calcul nous conduit à l'expression de la fonction W pour les exemples négatifs.

$$W(-1, f(x)) = \begin{cases} P_+ f(x) + H(P_+) & \text{si } f(x) > \rho_- \ , \\ P_- f(x) + H(P_-) & \text{si } \tau_- \geq f(x) > \rho_- \\ 0 & \text{sinon.} \end{cases} \tag{4.22}$$

avec $\tau_- = \frac{-H(P_-)}{P_-}$ et $\rho_- = \rho_+ = \rho$.

La fonction de perte ψ_r introduite dans [Bartlett 2008, Herbei 2006, Wegkamp 2010] est donnée par :

$$\psi_r(yf(x)) = \begin{cases} 1 - \frac{1-r}{r} yf(x) & \text{si } \ yf(x) < 0 \\ 1 - yf(x) & \text{si } \ 0 \leq yf(x) < 1 \\ 0 & \text{sinon} \end{cases} \tag{4.23}$$

alors,

$$\psi_r(yf(x)) = \frac{1}{H(r)} W\left(y, \frac{H(r)}{r} f(x)\right) \ .$$

où $H(r) = H(P_-) = H(P_+$ dans le cas d'une classification symetrique. Notons cependant que minimiser $\psi_r(yf(x))$ ou W aboutirait à des solutions équivalentes pour f. En minimisant $\psi_r(yf(x))$, la règle de décision recommendée dans [Bartlett 2008] classifie un example lorsque

$$|f(x)| > \delta_o = \frac{1}{2}$$

alors que la règle de décision construite dans [Grandvalet 2009] classifie un exemple si

$$|f(x)| > \frac{r}{H(r)} log \frac{r}{1-r}$$

Cette dernière rejette plus d'exemples lorsque le coût du rejet est faible et rejette moins d'exemples dans le cas contraire. Ceci sera très visible sur nos résultats.

4.5.2 Apprentissage du classifieur

Comme dans le SVM standard, nous considérons la fonction de perte pour les exemples d'apprentissage. La fonction de perte W (4.21-4.22) conduit à un problème d'optimisation similaire à celui du SVM standard.

4.5.2.1 Problème d'optimisation primal

Soit C_o une canstante de pénalisation à définir par validation croisée, on définit $D = C_o(P_+ - P_-)$, $B_i = C_o(1 - P_+)$ pour les exemples positifs et $B_i = C_o P_-$ pour

les examples négatifs. Le problème primal est

$$min_{f,b}\ \tfrac{1}{2}\|f\|^2_{\mathcal{H}} + \sum_{i=1}^{n} B_i\,|\tau_i - y_i(f(x_i)+b)|_+ + D\sum_{i=1}^{n}|\rho - y_i(f(x_i)+b)|_+ \tag{4.24}$$

où $|\cdot|_+ = max(\cdot, 0)$. Le carré de la norme de f est une fonctionnelle de régularisation dans un espace de Hilbert. Le problème primal (4.24) devient plus lisible en introduisant les variables d'écart ou variables élastiques ξ et η que nous avons illustrées sur la figure (4.7).

$$\begin{cases} min_{f,b,\xi,\eta} & \dfrac{1}{2}\|f\|^2_{\mathcal{H}} + \displaystyle\sum_{i=1}^{n} B_i\xi_i + D\sum_{i=1}^{n}\eta_i\ , \\ Sc & y_i(f(x_i)+b) \geq \tau_i - \xi_i,\ \ i=1,\ldots,n \\ & y_i(f(x_i)+b) \geq \rho - \eta_i,\ \ i=1,\ldots,n \\ & \xi_i \geq 0\ ,\quad \eta_i \geq 0, \qquad i=1,\ldots,n. \end{cases} \tag{4.25}$$

4.5.2.2 Problème dual

le Lagrangian du problème (4.25) est donné par :

$$\begin{cases} L(f,b,\xi,\eta,\alpha,\beta,\upsilon,\omega) = \frac{1}{2}\|f\|^2 + \sum_{i=1}^{n} B_i\xi_i \\ \quad +D\sum_{i=1}^{n}\eta_i - \sum_{i=1}^{n}\alpha_i\left[y_i(f(x_i)+b) - \tau_i + \xi_i\right] \\ \quad -\sum_{i=1}^{n}\beta_i\left[y_i(f(x_i)+b) - \rho + \eta_i\right] - \sum_{i=1}^{n}\upsilon_i\xi_i - \sum_{i=1}^{n}\omega_i\eta_i \end{cases} \tag{4.26}$$

avec : $\upsilon_i \geq 0,\ \omega_i \geq 0,\ \alpha_i \geq 0,\ \beta_i \geq 0,\ and\ i = 1,\ldots,n$. Les conditions de Kuhn-Tucker impliquent :

$$\begin{cases} \dfrac{\partial L}{\partial b} = 0 & \Rightarrow & \displaystyle\sum_{i=1}^{n}(\alpha_i+\beta_i)y_i = 0 \\ \dfrac{\partial L}{\partial f} = 0 & \Rightarrow & \displaystyle\sum_{i=1}^{n} f(\cdot) = (\alpha_i+\beta_i)y_i k(\cdot,x_i) \\ \dfrac{\partial L}{\partial \xi_i} = 0 & \Rightarrow & B_i - \upsilon_i - \alpha_i = 0 \Rightarrow 0 \leq \alpha_i \leq B_i \\ \dfrac{\partial L}{\partial \eta_i} = 0 & \Rightarrow & D - \omega_i - \beta_i = 0 \Rightarrow 0 \leq \beta_i \leq D \end{cases} \tag{4.27}$$

pour $i = 1,\ldots,n$. A partir de ces conditions, nous pouvons éliminer f, ξ et η du Lagrangian précédent.

$$\begin{cases} L(\alpha,\beta) = & \frac{1}{2}(\alpha+\beta)^T G(\alpha+\beta) - \tau^T\alpha - \rho^T\beta \\ Sc & y^T(\alpha+\beta) = 0 \\ & 0 \leq \alpha_i \leq B_i, i=1,\ldots,n \\ & 0 \leq \beta_i \leq D, i=1,\ldots,n \end{cases} \tag{4.28}$$

où $\tau = (\tau_1, \ldots, \tau_n)^T$ et $\rho = (\rho_1, \ldots, \rho_n)^T$ sont les vecteurs seuils appartenant à $\mathbb{R}^n$, G est la matrice d'influence de dimension $n \times n$ et de terme général

$$G_{ij} = y_i y_j k(x_i,\ x_j)$$

et $k(.,.)$ est le noyau reproduisant dans l'espace de Hilbert $\mathcal{H}$. Posons $\gamma = \alpha + \beta$, le problème (4.28) peut être réexprimé comme suit :

$$\begin{cases} max_{\alpha,\gamma} & -\frac{1}{2}\gamma^T G\gamma + (\tau - \rho)^T\alpha + \rho^T\gamma \ , \\ Sc & y^T\gamma = 0 \ , \\ & 0 \leq \alpha_i \leq B_i, \qquad i = 1, \ldots, n \ , \\ & 0 \leq \gamma_i - \alpha_i \leq D, \quad i = 1, \ldots, n \end{cases} \tag{4.29}$$

Le problème (4.29) est un problème quadratique sous contraintes de boite. Si on le comparait au problème gérant le SVM standard, le problème (4.29) présente un deuxième vecteur à optimiser mais nous verrons plus loin que le vecteur α est facilement recouvert par le vecteur γ. Dans le problème précédent, chaque inconnue γ_i peut être interprétée comme l'influence de l'exemple (x_i, y_i) dans la solution. Du fait que seuls les points sur la frontière sont importants pour la discrimination et qu'ils sont *a priori* peu nombreux, un grand nombre de ces coefficients seront nuls.

4.5.2.3 Résolution du problème

Le succès de la méthode des SVMs a entrainé le développement de nombreux algorithmes permettant leur mise en oeuvre. Parmi elles, l'algorithme SMO (*Sequential Minimal Optimization*) est souvent considéré comme le plus efficace. Cependant, l'approche présentée dans [Vishwanathan 2003, Loosli 2005] que nous avons developpée au chapitre précédent (3.6.1) montre une meilleure compléxité et une meilleure performance surtout pour de larges bases de données. Afin de résoudre le problème dual (4.29), nous adoptons la stratégie de répartition active de l'ensemble d'apprentissage. Cette strategy a été proposée dans [Vishwanathan 2003, Loosli 2005] pour l'apprentissage d'un SVM standard. D'abord, l'ensemble d'apprentissage est partitionné arbitrairement en ensemlbles support et non support. La fonction objectif est calculée pour la présente répartition. Tout vecteur violant les contarintes sont deplacés dans un ensemble approprié. Ces deux étapes sont répétées jusqu'à ce qu'un minimum prédéfini est atteint.

La table (4.2) indique la répartition de l'ensemble d'apprentissage en cinq sous ensembles définis par les contraintes du problème (4.25). Leur emplacement sur le graphe de la fonction de perte est illustré par la figure 4.8.

L'appartenance d'un exemple i à l'un des sous ensembles présentera les consé-

TAB. 4.2 – Ensemble d'apprentissage réparti en 5 sous-ensembles

I_0	Ensemble de points bien classés	$I_0 = \{i \mid y_i(f(x_i)+b) > \tau\}$
I_τ	Première charnière de la fonction de coût	$I_\tau = \{i \mid y_i(f(x_i)+b) = \tau\}$
I_B	Première pente de la fonction de coût	$I_B = \{i \mid \rho < y_i(f(x_i)+b) < \tau\}$
I_ρ	Deuxième charnière de la fonction de coût	$I_\rho = \{i \mid y_i(f(x_i)+b) = \rho\}$
I_D	Deuxième pente de la fonction de coût	$I_D = \{i \mid y_i(f(x_i)+b) < \rho\}$

quences suivantes sur les variables duales (α, γ) :

$$\begin{cases} i \in I_0 \Rightarrow \alpha_i = 0 & \gamma_i = 0 & ; \\ i \in I_\tau \Rightarrow 0 \leq \alpha_i \leq B_i & \gamma_i = \alpha_i & ; \\ i \in I_B \Rightarrow \alpha_i = B_i & \gamma_i = B_i & ; \\ i \in I_\rho \Rightarrow \alpha_i = B_i & B_i < \gamma_i < B_i + D & ; \\ i \in I_D \Rightarrow \alpha_i = B_i & \gamma_i = B_i + D & . \end{cases} \tag{4.30}$$

A partir de ces conséquence, on déduit que les seules composantes de γ à calculer sont uniquement celles qui correspondent aux indices $i \in I_\tau \cup I_\rho$. Cependant, α_i est soit constante soit égale à γ_i. Il est alors clair que si la répartition est supposée correcte, le problème 4.29 peut être résolu comme un problème de dimension réduite. Ce dernier devrait aboutir au calcul des γ_i pour $i \in I_\tau \cup I_\rho$.

Posons $I_c = \{I_B, I_D\}$ et $I_h = \{I_\tau, I_\rho\}$. Le problème (4.29) devient :

$$\begin{cases} L(\gamma) = \dfrac{1}{2}\gamma^T G\gamma - (S)^T\gamma \\ Sc : \quad y^T\gamma = 0 \\ \qquad 0 \leq \gamma_i \leq C, \ i = 1, \ldots, n \quad et \quad C_i = B_i + D \end{cases} \tag{4.31}$$

Les relations entre les paramètres de la formulation précédente et les paramètres initiaux du problème (4.25) peuvent être obtenues après formulation du Lagrangien du dual (4.31).

$$L(\gamma, \lambda, \mu, \nu) = \tfrac{1}{2}\gamma^T G\gamma - S^T\gamma + \lambda\gamma^T y - \nu^T\gamma + \mu^T(\gamma - C\mathbb{I}^n) \tag{4.32}$$

où les multiplicateurs de Lagrange λ, μ, ν doivent être positifs ou nuls et $\mathbb{I}^n$ est un vecteur unitaire. Ce Lagrangien peut être comparé au Lagrangien du primal (4.25) reformulé comme suit :

$$\begin{aligned} L = \tfrac{1}{2}\|f\|^2 - \textstyle\sum_{i=1}^n \gamma_i y_i f(x_i) - \sum_{i=1}^n \gamma_i y_i b + \sum_{i=1}^n \alpha_i(\tau - \rho) + \sum_{i=1}^n \gamma_i \rho + \\ \textstyle\sum_{i=1}^n \xi_i(B_i - \alpha_i - \upsilon_i) + \sum_{i=1}^n \eta_i(D - \beta_i - \omega_i) \end{aligned} \tag{4.33}$$

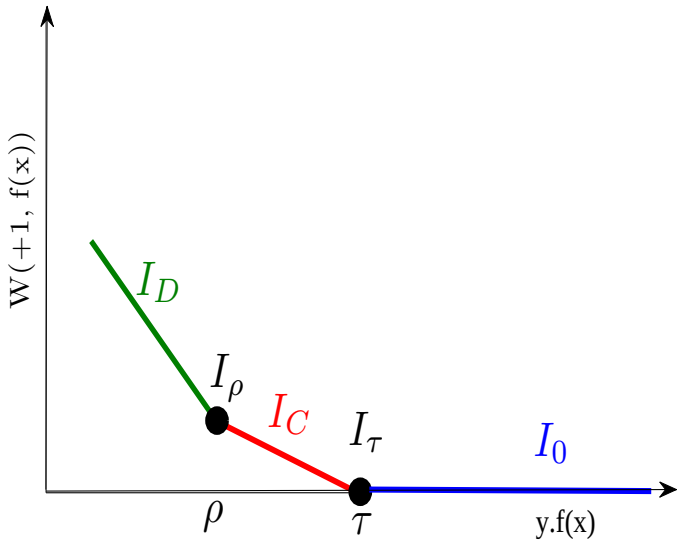


FIG. 4.8 – Fonction de perte construite indiquant les differents groupes de points.

En replaçant la variable f par γ, the problème (4.33) devient :

$$L(\gamma, b, \xi, \eta) = \frac{1}{2}\gamma^T G\gamma + b\gamma^T y - S^T\gamma + \xi^T(\alpha + \upsilon - B) + \eta^T(\gamma + \omega - D\mathbb{I}^n) \quad (4.34)$$

Afin de montrer les liens entre les variables primales et duales, nous devons examiner les conditions de KKT stipulant l'annulation du gradient du Lagrangien (4.34) par rapport aux variables primales γ et dans les differents sous ensembles. La table 4.3 decrit les propriétés de chaque sous ensemble par rapport aux variables initiales et les multiplicateurs de Lagrange.

4.5.3 Algorithme

Considérons une répartition initiale des points, c'est à dire que les ensembles (I_0, I_h et I_c) sont supposés connus. Les seules valeurs de γ non encore connues sont celles dont les indices appartiennet à I_h. Elles seront données par la solution du problème d'optimisation suivant et dont la dimension est largement inférieure à celle du problème initial.

On définit :

$$\gamma_h = \gamma(I_h)$$

TAB. 4.3 – Situation des constraintes pour les cinq types d'exemples

Ens-embles	Contraintes Initiales	Contraintes Primales	Contraintes Duales
I_0	$y_i[f(x_i)+b] > \tau_i$	$\xi_i = \eta_i = 0$	$\mu = 0,\ \nu = G\gamma + by - \tau \neq 0$
I_τ	$y_i[f(x_i)+b] = \tau_i$	$\xi_i = \eta_i = 0$	$\mu = 0,\ \nu = G\gamma + by - \tau = 0$
I_B	$\rho < y_i[f(x_i)+b] < \tau_i$	$\xi_i \neq 0,\ \eta_i = 0$	$\nu = 0,\ \mu = -G\gamma - by + \tau = \xi$
I_ρ	$y_i[f(x_i)+b] = \rho$	$\xi_i \neq 0,\ \eta_i = 0$	$\nu = 0,\ \mu = G\gamma + by - \rho = 0$
I_D	$y_i[f(x_i)+b] < \rho$	$\xi_i \neq 0,\ \eta_i \neq 0$	$\nu = 0,\ \mu = -G\gamma - by + \rho = \eta$

$$
\begin{aligned}
y_h &= y(I_h) \\
G_{hh} &= G(I_h, I_h) \\
c_C &= \textstyle\sum_{(i\in I_B)} B_i y_i + \sum_{(i\in I_D)} C_i y_i \\
S_h &= (\tau(I_\tau)^T \rho(I_\rho)^T)^T - G(I_h, I_D)(B(I_B)^T D\mathbb{I}(I_D)^T)^T
\end{aligned}
$$

Le problème (4.31) devient alors :

$$
\begin{cases} L(\gamma_h) = & \frac{1}{2}\gamma_h^T G_{hh}\gamma_h - S_h^T\gamma_h \\ Sc & y_h^T\gamma_h + c_C = 0 \end{cases}, \tag{4.35}
$$

Les conditions de Kuhn-Tucker nous permettent de définir le système à résoudre afin de trouver les valeurs de γ qui sont encore inconnues.

$$
\begin{cases} G_{hh}\gamma_h = S_h - y_h^T\lambda \\ y_h^T\gamma_h = -c_C \end{cases}, \tag{4.36}
$$

Après la résolution du système, une composante de γ qui violerait les contraintes primales ou duales doivent être deplacées vers un ensembe approprié. Ces itérations sont refaites jusqu'à ce que le cout minimal prédéfini soit atteint.

4.5.4 Complexité du problème

Durant le processus d'apprentissage, le cœur de l'algorithme, phase la plus gourmande en temps de calcul est la résolution du système linéaire(4.36). Sachant que nous avons besoin de refaire ce calcul à chaque itération de l'algorithme avec seulement un point de différence par rapport à l'itération précédente, nous avons opté

pour une méthode de résolution incrémentale dont la complexité est $\mathcal{O}(n^2)$ plutôt que les méthodes directes en $\mathcal{O}(n^3)$. La compléxité de cet algorithme est comparable à celle du SVM standard [Loosli 2005]. La seule surcharge de calcul que présente notre algorithme est qu'il utilise cinq cathégories d'exemples alors que le SVM standard en utilise trois.

4.6 Conclusion

A travers ce chapitre, nous avons exposé une extension aux Support Vector Machines en introduisant une option de rejet d'ambigüités durant la phase d'apprentissage.

Chapitre 5

Résultats expérimentaux

5.1 Préambule

A travers ce chapitre, nous allons mettre en évidence les performances des algorithmes que nous avons developpés.

1. Un premier volet de ce chapitre met l'accent sur l'intérêt du rejet ainsi que la flexibilité des seuils de décision. Comme nous l'avons déjà évoqué, les performances d'un classifieur sont surtout jugées par le coût global de la classification. Cette approche sera soutenue par quelques tests effectués sur plusieurs bases d'étude. Notons qu'en parallèle, une comparaison sera faite entre différents classifieurs visant le même objectif.
2. Parmi les objectifs de notre projet de recherche, l'identification de perturbations dans le réseau électrique. Comme dans tout travail de reconnaissance de formes, la première étape consiste à localiser et à caractériser les formes à identifier. Pour ce faire, nous avons construit une base de données contenant des signaux électriques avec des perturbations. Pour localiser ces dernières, nous avons utilisé des outils de traitement du signal. Notons que la deuxième étape du travail, c'est à dire la classification de ces perturbations par SVMs est en cours de réalisation.

5.2 Performances du classifieur

Pour évaluer les performancs du classifieur utilisé, nous avons considéré différentes bases de données.

5.2.1 Bases de données

La première série de tests est menée en utilisant des données synthétiques, des données relatives à une application biomédicale et des données de très grande di-

mension.

5.2.1.1 Données médicales

Le rejet de cas ambigus s'avère d'une grande importance pour l'aide à la décision dans les applications médicales. Des données biomédicales sont alors utilisées dans ce travail afin de mettre en évidence l'intérêt du rejet. Des enregistrements electrocardiographiques de la base (www.physionet.org /physiobank/database/mitdb) son considérés. Chaque enregistrement est accompagné d'un fichier d'étiquettes. Dans ce fichier, chaque battement cardiaque a été identifié et étiqueté par un cardiologue. Nous avons suivi les recommandations pratiques [**?**] pour former deux classes :

- La classe positive contient les battements ectopics V.
- La classe négative contient les battements normaux et les blocs de branches.

Les enregistrements (102, 104, 107, 217) contenant des battements obtenus par pacemaker sont exclus. 23 enregistrements avec moins de 40 battements ectopics sont également exclus. Cecu nous laisse 21 enregistrements d'intérêt. Chaque battement a été quantifié avec un vecteur de caractéristiques de dimension 7. L'extraction des caractéristiques que nous avons effectuée a été decrite dans un travail antérieur publié dans [Zidelmal 2012].

Base d'apprentissage Pour réaliser l'appprentissage d'un classifieur global, nous avons considéré les battements issus des cinq premières minutes de dix enregistrements sélectionnés de façon aléatoire. Ceci induit une base d'apprentissage de 1500 echantillons.

Base de test De chaque enregistrement, nous utilisons les 25 dernières minutes pour la phase de test. Ainsi, la base d'apprentissage est complètement dissociée de la base de test. Ceci nous permet d'évaluer la capacité de généralisation de notre algorithme.

5.2.1.2 Données synthétiques

Nous avons généré deux classes de données en utilisant la fonction classique dataset. Ces deux classes sont gérées chacune par une densité de propbabilité gaussienne. Ces gaussiennes sont caractérisées par des dispersions ou variances identiques mais des moyennes différentes choisies pour que ces deux classes présentent plusieurs exemples ambigus (Voir figure 5.1).

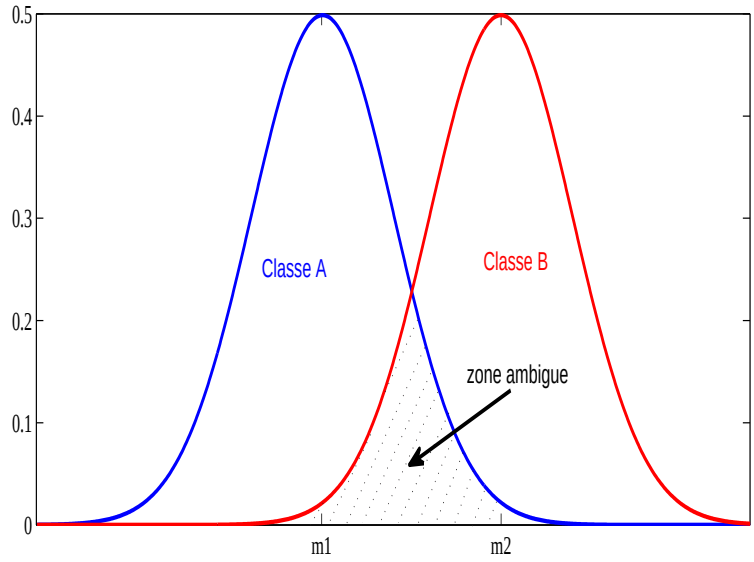


FIG. 5.1 – Génération de deux classes de données selon deux distributions gaussiennes de moyennes respectives m_1 et m_2 avec chevauchement afin de créer des exemples ambigûs.

5.2.1.3 Données de très grande dimension

Pour examiner les performances du classifieur avec des données de grande dimension, nous l'avons testé sur les données *forest CoverType database* enregistrées à partir du répertoir UCI [1]. Nous avons alors opté pour la discrimination de la classe positive *Cottonwood* (2747 exemples) par rapport à la classe négative *Douglas-fir* (17367 exemples).

– Dans ces deux dernières bases d'études, l'apprentissage du classifieur est réalisé en utilisant 60 % de l'ensemble des échantillons. La base de test est dissociée de la base d'apprentissage.

5.2.2 Protocole expérimental

Pour mener à bien un algorithme de classification, quelques protocoles doivent êtres respectés :

[1] (http ://kdd.ics.uci.edu/databases/covertype/)

Prétraitement des données : Nous avons vu au premier chapitre (2.3.1) que les attributs caractérisants une données peuvent être de natures différentes, d'ordres de grandeurs différents et aussi exprimés avec des unités différentes, une normalisation est alors nécessaire. Chaque attribut suit alors une densité de probabilité gaussienne de moyenne nulle et de variance unité.

Choix du noyau : Il n'existe pas de méthodes permettant un choix *a priori*. Généralement, plusieurs noyaux sont testés afin de sélectionner celui qui minimiserait l'erreur empirique. Pour la présente application, nous avons retenu le noyau gaussien.

Reglage des hyperparamètres : Nous avons deux hyperparaètres à regler : le paramètre de dipersion σ du noyau et le paramètre de pénalisation C_o indiqué lors de la position du problème d'optimisation au troisième chapitre (4.5.2). Ces deux paramètres sont estimés par Validation Croisée (*Cross Validation*).

5.2.3 Mise en évidence de la zone de rejet

Une première série de tests a été réalisée afin de mettre en évidence l'approche utilisée pour minimiser le coût de la classification. Pour ce faire, nous avons sélectionné les enregistrements 214 et 221 contenant 3546 battements normaux et 652 ectopics. Comme ces données ne sont accompagnées d'aucune matrice de coût réels, nous avons considéré

$$R_p = R_n = r \quad et \quad P_+ = 1 - \frac{C_p}{C_n} P_- = 1 - \theta P_-$$

où $\theta = 1$ comme dans [Bartlett 2008, Herbei 2006, Wegkamp 2010]

et $\theta \geq 1$ dans [Grandvalet 2009].

Souvent, en pratique, particulièrement dans les applications médicales ou industrielles, les FN sont plus couteux que les FP, c'est à dire ($\theta > 1$).
Les figures 5.2 et 5.3 indiquent respectivement un exemple de zone de rejet induite pour $\theta = 1$ et pour $\theta > 1$. Selon la figure 5.3, le classifieur encourage plus le rejet de FN car plus couteux que les FP.

5.2.4 Mise en évidence du coût de la classification

Les études comparatives de classifieurs sont généralement basées sur leurs taux d'erreurs commises ou sur leurs scores. Cependant, le taux d'erreurs ne doit pas être le seul paramètre qui nous permettrait de juger les performances d'un classifieur.

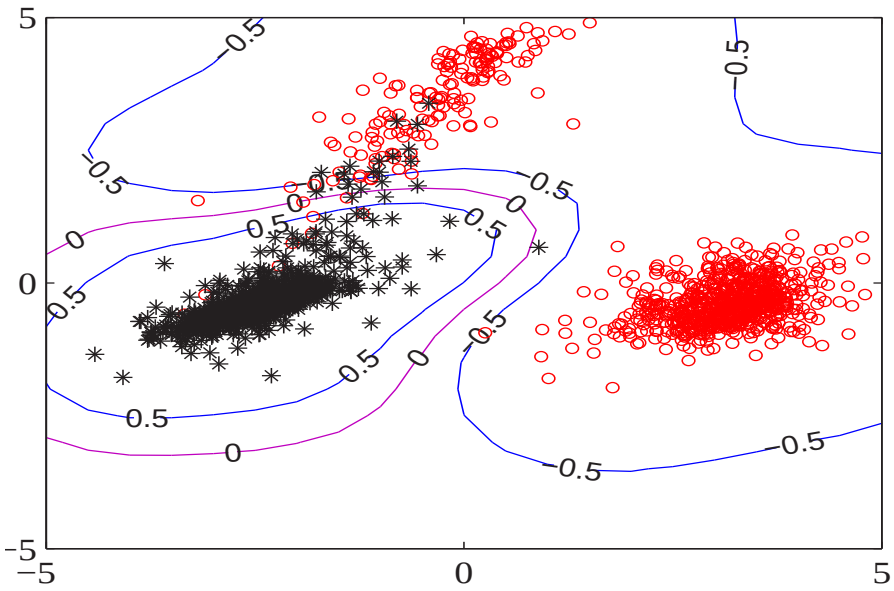


FIG. 5.2 – Zone de rejet induite par les seuils définis en fonction des coûts d'exemples mal classés et celui des échantillons rejetés. Les exemples positifs sont représentés par des étoiles noires alors que les exemples négatifs sont représentés par des ronds rouges. Les lignes $+0.5$ et -0.5 correspondent respectivement à δ_o et $-\delta_o$ et la ligne 0 correspond à $f(x) = 0$ ou $P(Y = 1 \mid X = x) = 0.5$.

Dans beaucoup d'applications, le coût de la classification est un paramètre qui doit être considéré d'autant plus qu'à l'origine, l'objectif d'un classifieur de Bayes avec ou sans rejet est de minimiser le risque ou le coût de la classification.

Nous illustrons ceci tout en comparant le classifieur proposé dans [Bartlett 2008, Herbei 2006, Wegkamp 2010] où le seuil de rejet $\delta_o \in [r, \quad 1-r]$ et classifieur (SVMR) que nous avons considéré où les seuils de rejet sont :

$$\delta_+ = \log \frac{P_+}{1 - P_+}$$

et

$$\delta_- = \log \frac{P_-}{1 - P_-}$$

respectivement pour les exemples positifs et pour les exemples négatifse. Pour cette comparaison, nous réalisons une classification symétrique, c'est à dire, $P_+ = 1 - P_-$.

Les figures 5.4 et 5.5 obtenues respectivement avec les données synthétiques et les enregistrements ECG (214 et 221) montrent que dans tous les cas, la règle de

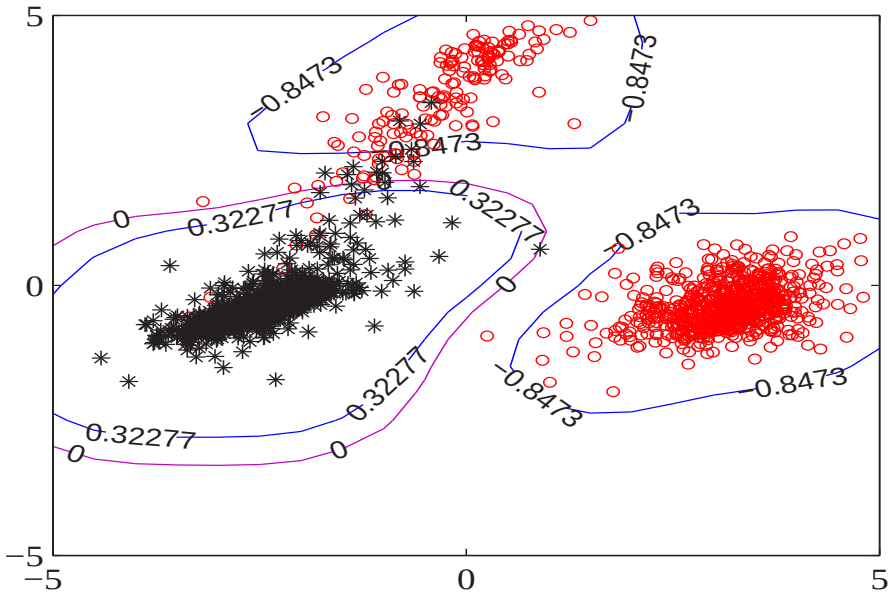


FIG. 5.3 – Zone de rejet induite par les seuils définis en fonction des coûts d'exemples mal classés et celui des échantillons rejetés. Les exemples positifs sont représentés par des étoiles noires alors que les exemples négatifs sont représentés par des ronds rouges. Les lignes $+0.32277$ et -0.8473 correspondent respectivement à δ_+ et δ_- et la ligne 0 correspond à $f(x) = 0$ ou $P(Y = 1 \mid X = x) = 0.5$

décision du classifieur généralisé [Grandvalet 2009] rejette peu d'exemples lorsque le cout du rejet est élevé mais rejette plus d'exemples si le coût du rejet est faible.

la règle de décision définie dans [Bartlett 2008, Herbei 2006, Wegkamp 2010] considérant le seuil de rejet $\delta_o = 1 - r$ comme plus grande valeur de δ_o rejette plus d'échantillons quel que soit le coût du rejet. Pour $\delta_o = r$, cette règle rejette très peu d'exemples ; surtout pour les faibles valeurs de r, elle tend vers un classifieur sans rejet. Pour la valeur intermédiaire $\delta_o = 0.5$ vue comme un compromis entre r et $1 - r$, les règles de décision [Bartlett 2008, Herbei 2006, Wegkamp 2010] et [Grandvalet 2009] sont similaires lorsque $r = 0.24$.

Il a été si bien souligné dans la publication de Chow [Chow 1970] que l'avantage d'un classifieur avec rejet peut être appuyé par le compromis erreur-rejet. Le taux d'erreurs E et le taux de rejet R sont des fonctions monotônes de r. Nous représontons $E(r)$ en fonction de $R(r)$ quand r varie entre 0.5 et 0.12 en prenant $\delta_o = 0.5$. La figure 5.6 illustre le compromis obtenu avec la règle de décision [Grandvalet 2009] (courbes noires) et celui obtenu avec la règle de décision

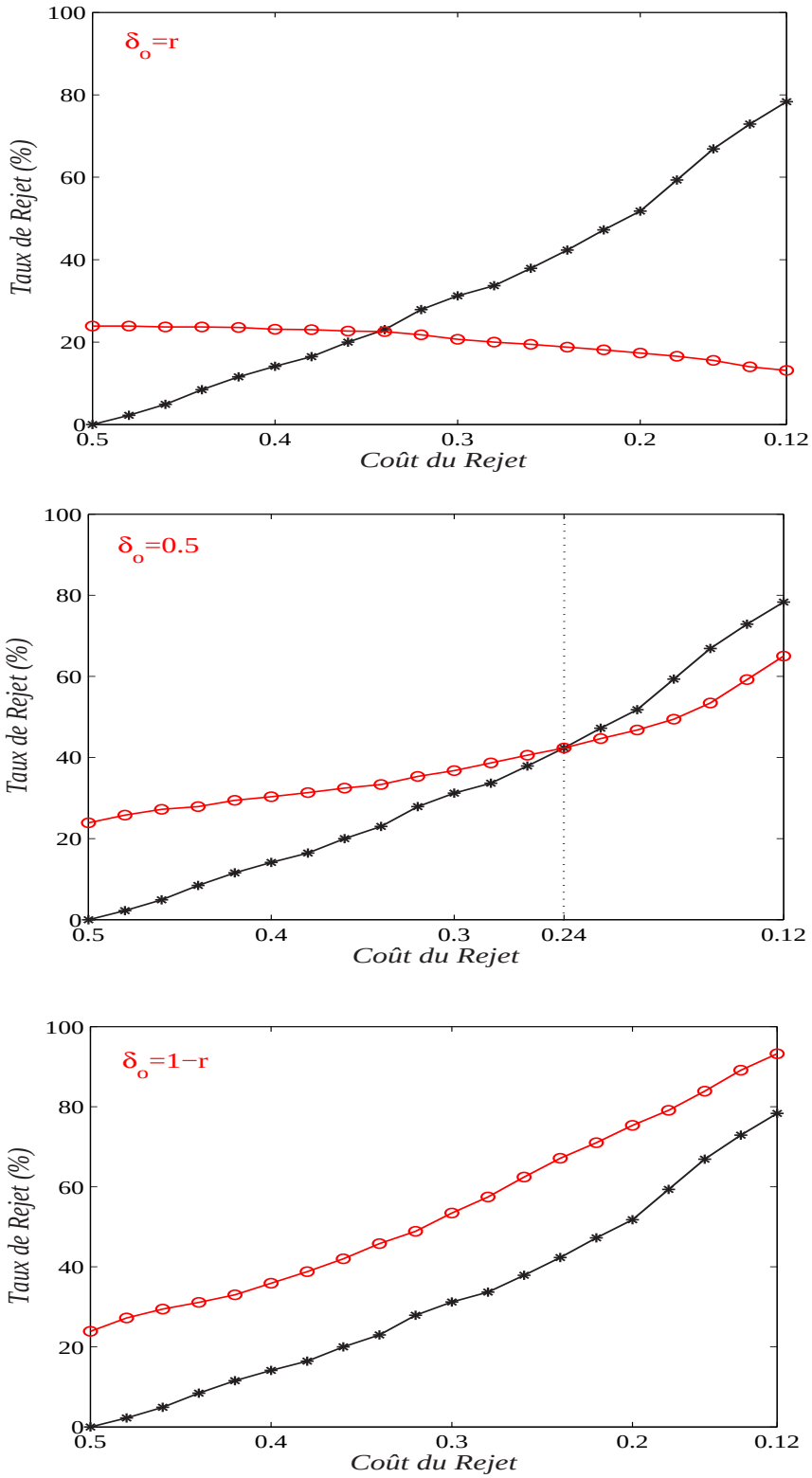


FIG. 5.4 – Comparaison des taux de rejet en fonction du côut de rejet r obtenus respectivement avec le SVMR [Grandvalet 2009] (courbes noires) et avec la règle de décision [Bartlett 2008, Herbei 2006, Wegkamp 2010] (courbes rouges). Ces résultats sont obtenus en utilisant les données synthétiques.

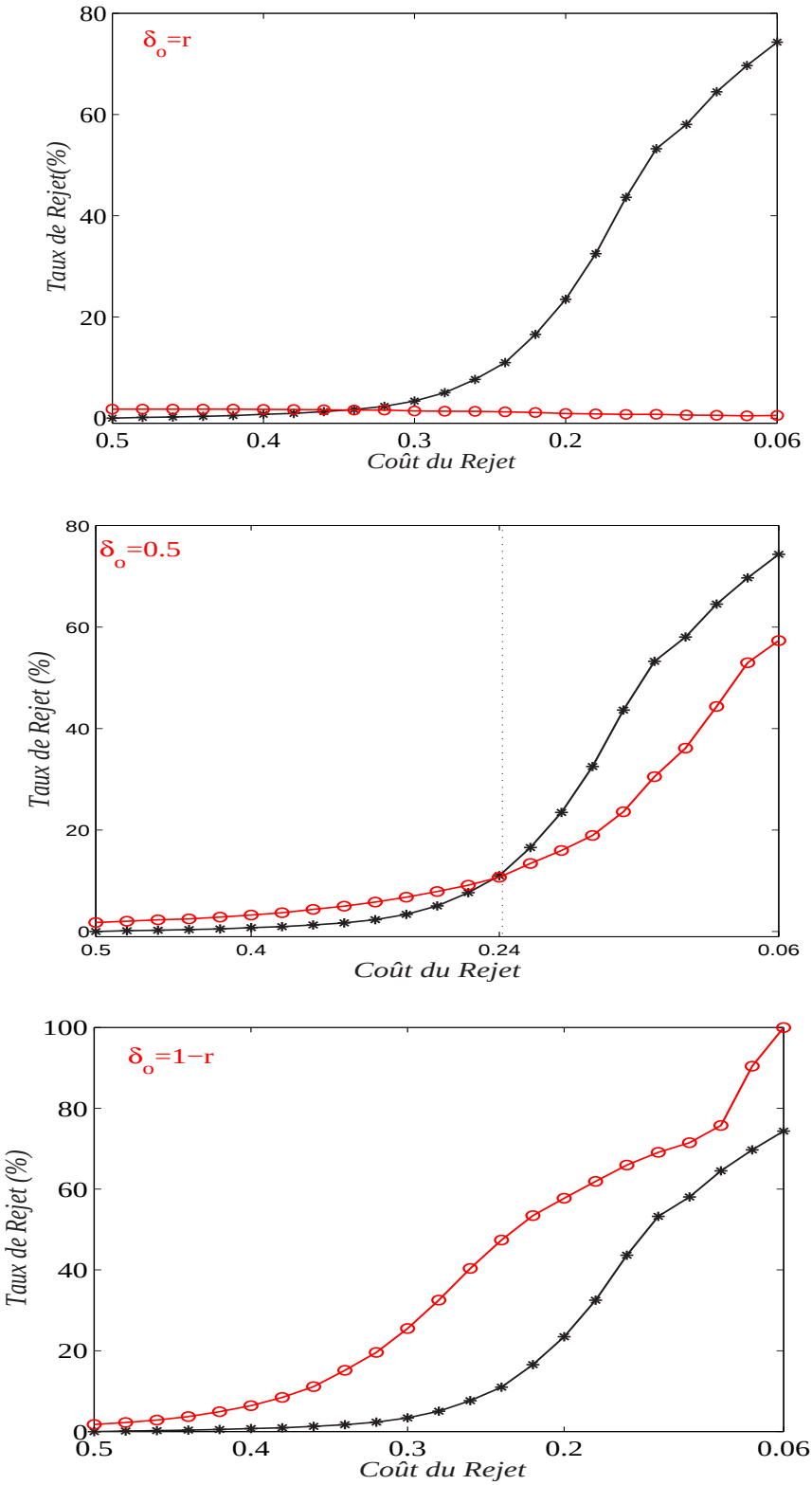


FIG. 5.5 – Comparaison des taux de rejet en fonction du côut de rejet r obtenus respectivement avec le SVMR [Grandvalet 2009] (courbes noires) et avec la règle de décision [Bartlett 2008, Herbei 2006, Wegkamp 2010] (courbes rouges). Ces résultats sont obtenus en utilisant les données médicales.

[Bartlett 2008, Herbei 2006, Wegkamp 2010] (courbes rouges). La différence entre les deux résultats est très claire. Elle est due au fait que les zones de rejets induites par les deux règles sont de tailles différentes.

A partir des résultats de la figure 5.6, nous pouvons déduire un autre paramètre très intéressant indiquant le rapport entre la variation de l'erreur et la variation du rejet (*error-reject ratio*) défini dans [Chow 1970] comme $\frac{\triangle E}{\triangle R}$ (voir lignes interrompues).

Pour les côut élevés du rejet $(0.4 \leq r \leq 0.5)$, la règle [Grandvalet 2009] donne un *error-reject ratio* de -0.58, -0.84 et -0.42 respectivement pour les données synthétiques, pour les données ECG et pour les données forest. Ceci signifie que 58%, 84% and 42% respectivement des echantillons rejetés devraient être mal classés. Le même test a été reconduit en utilisant la règle de décision proposée dans [Bartlett 2008, Herbei 2006, Wegkamp 2010] avec $\delta_o = 0.5$, l'*error-reject ratio* obtenus sont -0.15, -0.23 et -0.22 respectivement avec les données synthétiques, les données ECG et les données forest. Ceci signifie que seulement 15%, 23% and 22% respectivemet des exemples rejetés devraient être mal classés. Il est claire que la règle de décision [Grandvalet 2009] doit aboutir à un meilleur coût de classification.

La dernière série de tests a été réalisée en considérant tous les enregistrements ECG sélectionnés. Les résultats moyens obtenus sont représentés sur les figures 5.7 et 5.8. L'erreur en fonction du rejet diminue jusquà une valeur quasiment constante (Figure 5.7). Un autre tracé intéressant présenté sur la même figure indique l'*error-reject ratio*. Le point d'inflection sur ce tracé indique la plus grande diminution de l'erreur en fonction du taux de rejet.

Deux paramètres statistiques sont également utilisés afin de mettre en évidence les performances du classifieur que nous avons construit. La sensitivité S_e et la prédictivité positive P_p donnés par :

$$S_e = \frac{TP}{TP + FN}; \qquad P_p = \frac{TP}{TP + FP}$$

où les vrais positifs (VP) sont les exemples positifs classés positifs. La figure figure 5.8 (haut) indique la variation du coût de la classification donné par :

$$C_c = [C_p FN + C_n FP + rR_{rej}]/N_{tot} \tag{5.1}$$

où R_{rej} représente le nombre d'exemples rejetés et N_{tot}, le nombre totale d'examples dans la base de test. La même figure montre le coût optimal de classification correspond à un bon compromis erreur-rejet (voirFigure 5.7). La figure 5.8 (bas) montre que la prédictivité positive est proche de 99.8%.

Sur la même figure, nou montrons une sensitivité de plus de $98,2\%$ sans rejet et plus de 99% pour le côut minimal de classification en considérant $R_p = R_n = r$

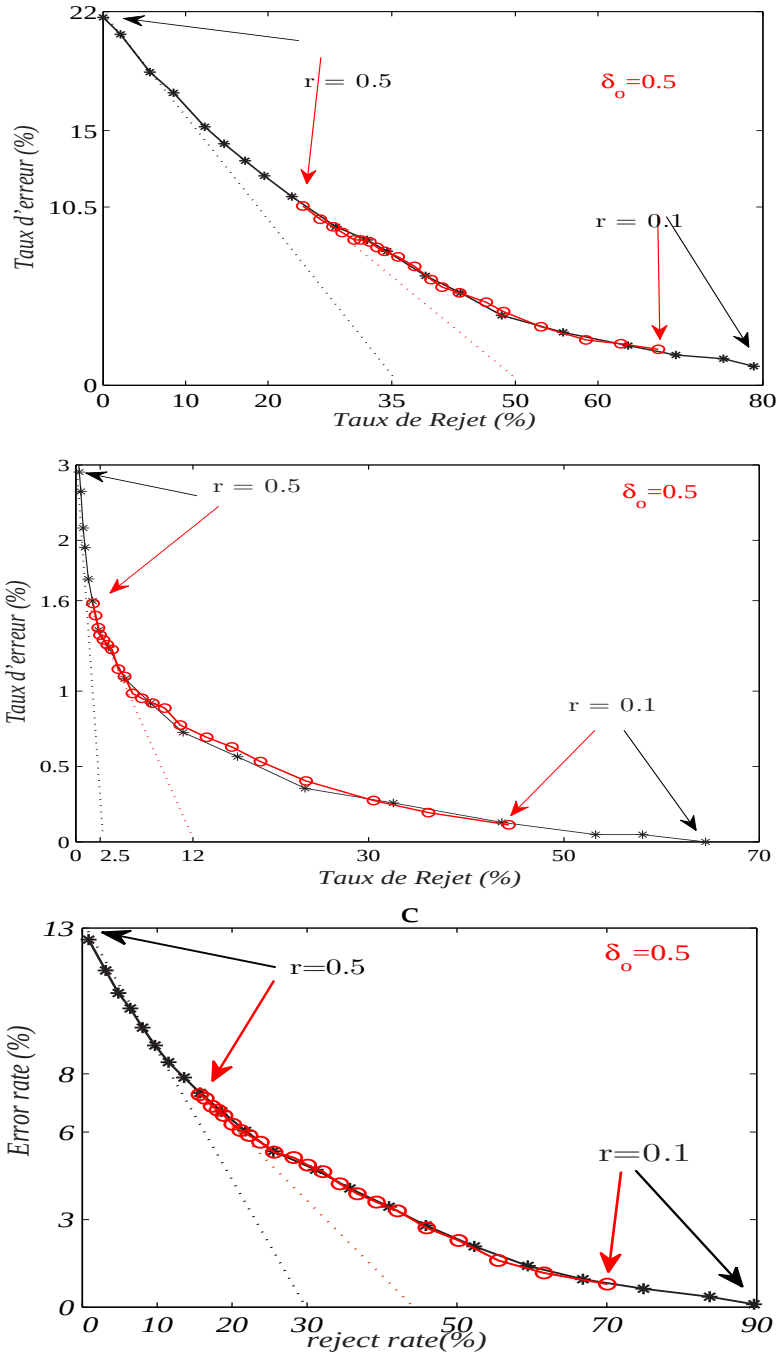


FIG. 5.6 – (a) Compromis erreur-rejet obtenu avec les données synthétiques, (b) données ECG, (c)avec les données forest ; en utilisant la méthode [Grandvalet 2009] (courbes noires) et en utilisant la méthode [Bartlett 2008, Herbei 2006, Wegkamp 2010] avec $\delta_o = 0.5$ (courbes rouges).

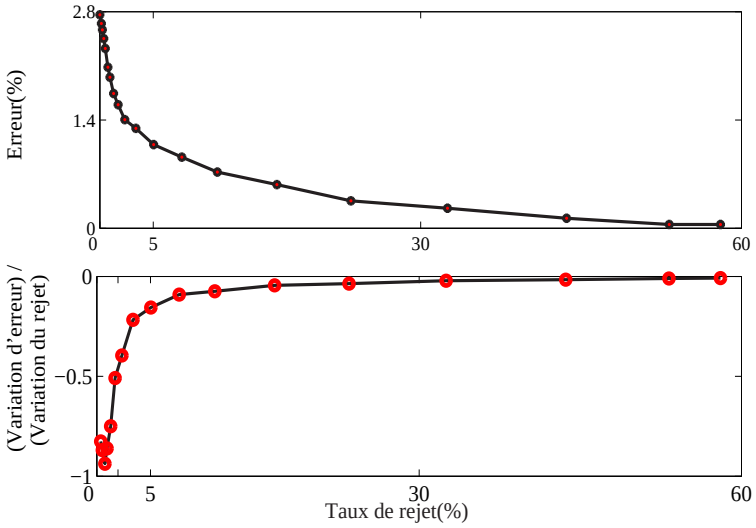


FIG. 5.7 – Haut : taux d'erreurs en fonction du taux de rejet. Bas : Variation du taux d'erreurs en fonction de la variation du taux de rejet

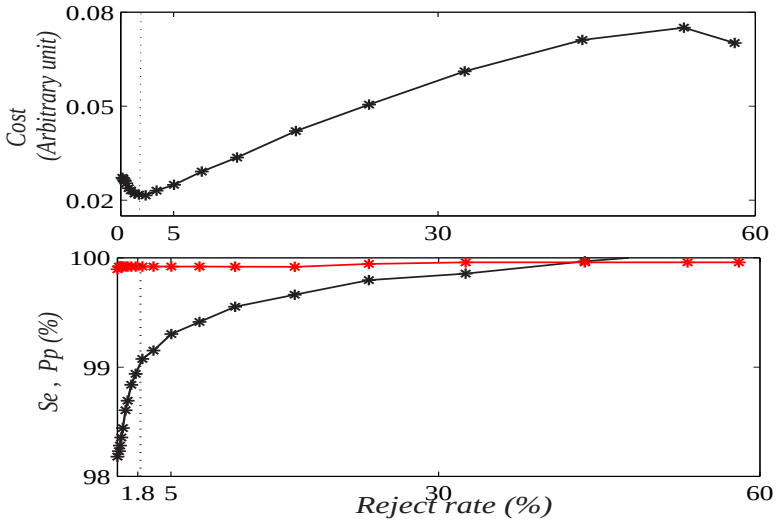


FIG. 5.8 – Haut : coût de la classification en fonction du taux de rejet. Bas : Sensitivité (courbe noire) et la prédictivité positive (courbe rouge) en fonction du taux de rejet.

and $C_n = 1$ et $\theta = 1.2$. Il est alors clair que le coût minimal n'a pas été obtenu pour r=0.5 (SVM standard) mais plutôt pour un taux de rejet de 1.8%. Ceci semble être un bon résultat car un système qui rejette trop n'est tout de même pas le plus désiré. Dans beaucoup d'applications, l'utilisateur doit opter pour un compromis erreur-rejet correspondant au coût minimal.

5.3 Localisation de perturbations électriques

Après avoir traité les signaux issus de l'activité électrique du coeur, nous nous intéressons dans ce qui suit aux signaux issus de sources électriques de puissance. Malgré la différence naturelle des origines de ces deux types de signaux, l'identification des évènements électriques et celle des ÉlectroCardioGrammes peut être réalisée par les mêmes outils. La différence se trouve dans l'interprétation physique des caractéristiques identifiées.

Au cours de ces dernières années, la Qualité de l'Energie est devenue une préoccupation majeure pour les fournisseurs et les consommateurs d'électricité. Les perturbations sont causées par la prolifération des charges non linéaires, telles que les redresseurs, le matériel informatique, les appareils de climatisation ou encore les éclairages à base de tubes fluorescents. Ces appareils absorbent des courants non sinusoïdaux et introduisent une distorsion de tension et de courant élevée avec des harmoniques excessives.

5.3.1 Construction d'une bases de données

Pour mener cette étude, nous avons créé deux bases de données contenant des signaux altérés par les principales perturbations :

- Les interruptions de tension.
- La présence d'harmoniques.
- Les houles de tension.
- Des affaissements de tension.

1. des données avec différentes perturbations ont été simulées selon la norme recommandée par IEEE-Std-1159 [IEEE 1995]. Nous considérons un signal $x(t)$ de fréquence $50Hz$ et de durée 0.2s (10 cycles) échantillonné avec une fréquence $f_s = 1000Hz$. Dans tous les cas, nous avons engendré des perturbation pendant trois cycles $[t_1, t_2] = [0.06s, 0.12s]$. Six perturbations ont été introduites : affaissement de tenstion, houle de tension, et interruption avec et sans harmonics. Dans chaque cas, nous avons considéré les trois premières harmoniques, c'est à dire $(150, 250and350Hz)$. Dans ce cas, la fréquence maximale du signal est de $350Hz$.

2. Un ensemble de données réelles a été généré au sein du laboratoire. Pour ce faire, nous avons considéré un variateur et une lampe halogène. Le courant a été varié de façon à créer différentes perturbations. Le signal enregistré sur une carte dSPACE DS1104 pendant $2s$ est discretisé avec une fréquence d'échantillonnage de $f_s = 10KHz$.

5.3.2 Méthode proposée

5.3.2.1 Transformée de Stockwel

la Transformée de Stockwell (ST) a été introduite par R.G.Stockwellen en 1996 [Stockwell 1996]. La transformée de Stockwell relève de la large gamme d'outils d'analyse multi-résolution. Elle peut être vue comme une Transformée de Fourier à Court Terme (TFCT) à fenêtre d'analyse variable ou comme une transformée en ondelette avec information de phase.

Vue comme une Transformée de Fourier à Court Terme, La ST d'un signal $x(t)$ est donnée par :

$$S(\tau, f) = \int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-i2\pi ft}dt \tag{5.2}$$

où $w(t)$ est une fenêtre temporelle centrée à $t = 0$ et utilisée pour extraire un segment du signal $x(t)$. La fenêtre de pondération $w(t)$ a été choisie pour être une gaussienne normalisée. Cette fenêtre a été choisie car c'est la plus compacte en temps et en fréquence. Elle est donnée par :

$$w(t) = \frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-t^2}{2\sigma^2}} \tag{5.3}$$

où σ characterise la largeur de la gaussienne. On associe à $w(t)$ une translation τ et une dilatation σ. Une contrainte a été ajoutée afin de rendre la largeur de la fenêtre, fonction de la fréquence :

$$\sigma(f) = \frac{1}{|f|} \tag{5.4}$$

Dans ce cas, l'equation 5.3 devient

$$w(t-\tau) = \frac{|f|}{\sqrt{2\pi}}e^{\frac{-(t-\tau)^2f^2}{2}} \tag{5.5}$$

La TS est alors réexprimée comme suit :

$$S(\tau, f) = \frac{|f|}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} x(t)e^{\frac{-(t-\tau)^2f^2}{2}}e^{-i2\pi ft}dt \tag{5.6}$$

où τ et f indiquent respectivement l'instant (où est localisé le spectre local) et la fréquence de Fourier. La largeur de la fenêtre σ variant inversement à la fréquence fait que la ST offre une multi-résolution au signal analysé. A chaque instant t_o, la fonction $S(t_o, f)$ à une dimension, représente un spectre local. De façon analogue, la fonction $S(\tau, f_o)$ à la seul dimension τ est appelée voie.

5.3.2.2 Energie de Shannon

L'énergie de Shannon [Shannon 1974] pour un signal dont la TS est donnée par la matrice S est.

$$SSE(j) = -\sum_{n=n_o}^{n_1} [S(j,n)]^2 \log[S(j,n)]^2 \tag{5.7}$$

où n_o et n_1 correspondent respectivement à $f_o = 350$ et $f_1 = 500Hz$. L'energie de Shannon normalisée est :

$$SSE_n = \frac{SSE}{|max(SSE|} \tag{5.8}$$

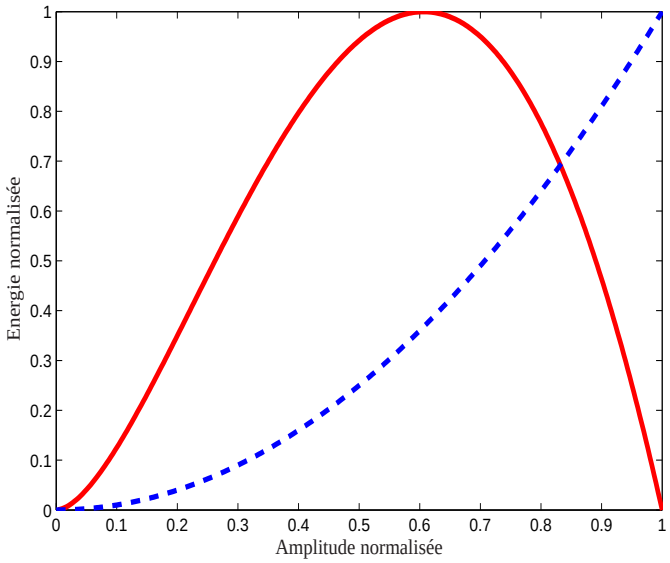


FIG. 5.9 – Comparaison de différentes energies obtenues avec différentes méthodes, energie de Shannon (trait continu) et enrgie quadratique(trait interrompu).

5.3.2.3 Application

Pour détecter les ruptures d'amplitude dans le signal, nous allons considéré, la gamme de fréquences où ces ruptures sont plus pertinentes que le signal utile lui-même. La figure 5.10 montre que les singularités dans le spectrogramme d'un signal contaminé par une interruption instantanée et par les harmoniques (150Hz, 250Hz et 350Hz), sont de haute fréquence $f > 350Hz$ (voir la zone encerclée). Sur cette

figure, on peut voir que dans la bande de fréquences $[f_o, f_1]$, les ruptures sont plus pertinentes.

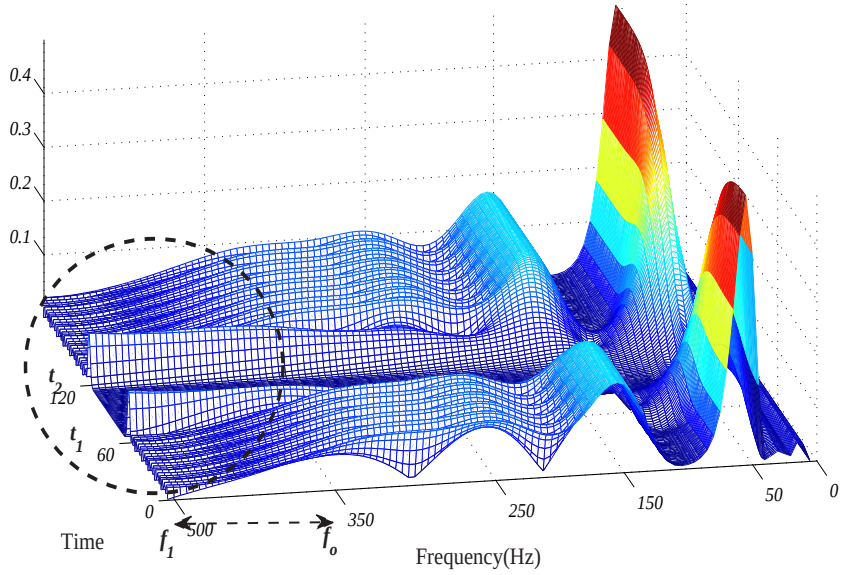


FIG. 5.10 – Localisation de singularité sur le plan temps-fréquence.

Interruption de tension

Nous avons simulé deux interruptions instantanées de tension, respectivement sur un signal sans harmoniques et sur un signal avec harmoniques. Les figures 5.11 et 5.12 montrent respectivement la détection du début et de la fin de l'interruption pour chaque cas. Les spectrogrammes respectifs présentent une réduction d'énergie au cours de la perturbation. Cette réduction est similaire à l'interruption visible dans le domaine temporel. En d'autres termes, la ST génère des contours appropriés pour le classement par simple inspection visuelle.

Le même test a été effectué en utilisant des données réelles bruitées avec de multiples interruptions. La figure 5.13 montre le signal original avec interruptions instantanées, le spectrogramme du signal et la detection des instants t_1 et t_2 sur l'enveloppe de Shannon. Dans chaque cas, on peut voir que l'enveloppe de Shannon conduit à une bonne détection du début et de la fin de toutes les ruptures du signal.

Affaissement de tension

Les figures 5.14, et 5.15 representent respectivement un cas d'affaissement instantané avec et sans harmoniques. Dans chaque cas, le spectrogramme affiché dans

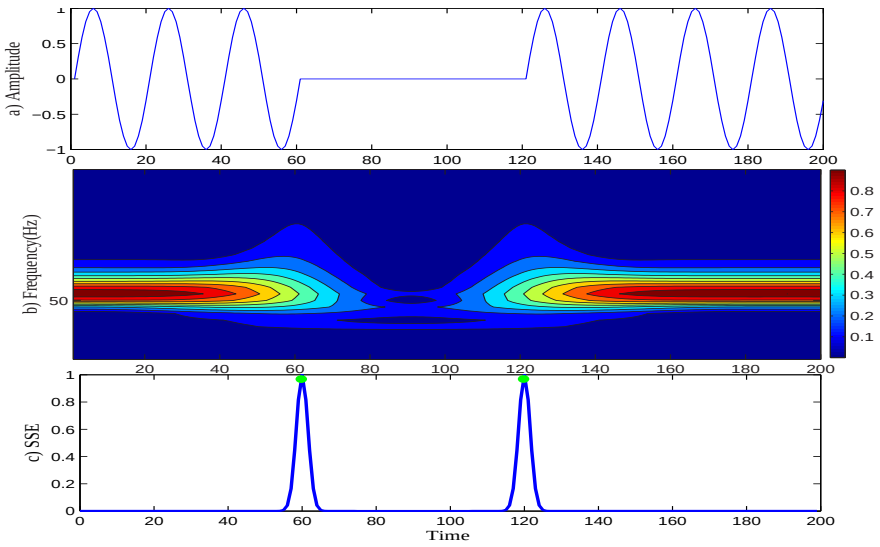


FIG. 5.11 – (a) Signal avec interruption instantanée ; (b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon.

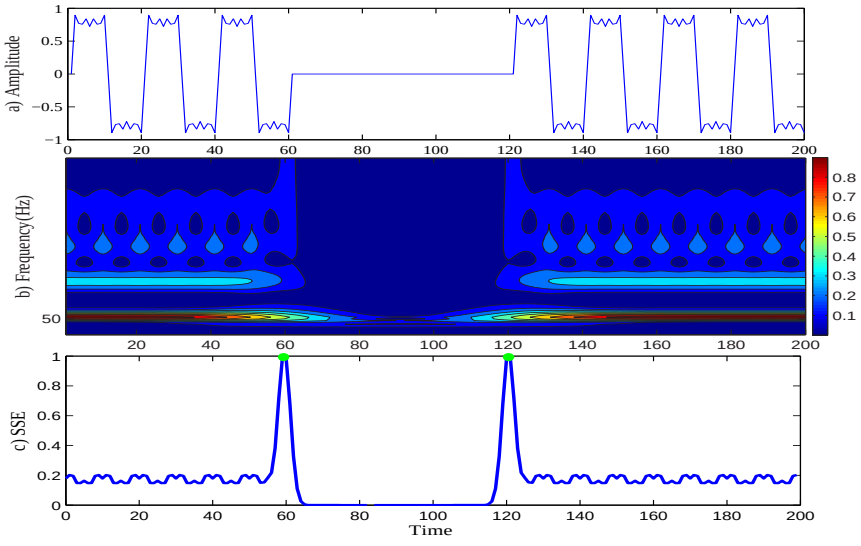


FIG. 5.12 – (a) Signal avec interruption instantanée en présence d'harmoniques ; (b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon.

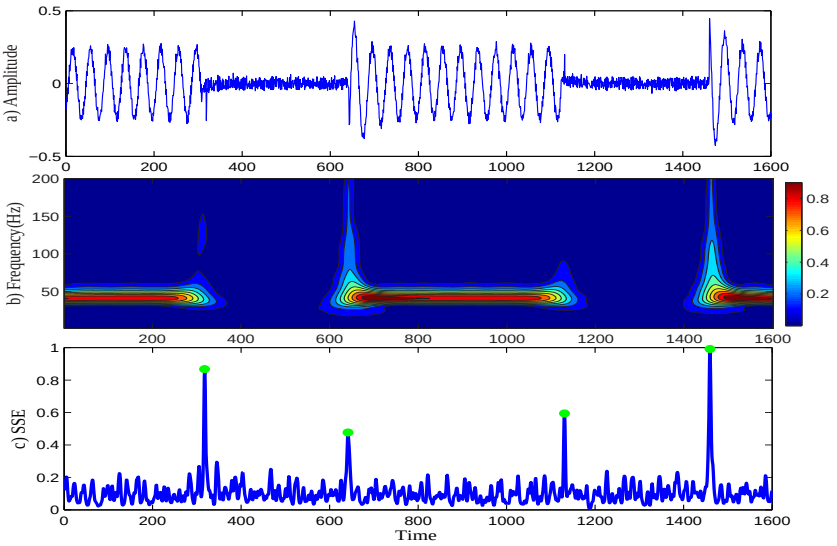


FIG. 5.13 – (a) Signal réel (données réelles bruitées) avec de multiples interruptions ; (b) Spectrogramme du signal ; (c) Detection des instants t_1, t_2 et t_3, t_4 sur l'enveloppe de Shannon.

5.14.b et 5.15.b présente une réduction d'amplitude au cours de la perturbation, similaire à celle observée sur le signal temporel. Sur la figure 5.14.c et 5.15.c, on peut voir que le début t_1 et la fin t_2 de la perturbation sont détectés avec précision.

Un test similaire a également été mené sur des données réelles avec affaissement. Les résultats sont présentés sur la figure 5.16. Ces résultat illustrent un signal original avec affaissement instantanée, le spectrogramme du signal ainsi que la détection des instants t_1 et t_2 sur l'enveloppe de Shannon.

Houle de tension

Une houle de tension (augmentation instantannée de tension) est souvent causée par une forte charge subite. Dans ce cas, la tenstion augmente de 10% à 90% de sa valeur initiale. Dans la présente application, nous avons considéré une augmentation de 33% de la valeur initiale du signal. Comme précédemment, nous avons réalisé des tests sur des signaux avec houle de tension sans harmoniques Fig.5.17 puis avec harmoniques Fig.5.18.

Le même test a égalememnt été mené sur des signaux réels que nous avons enregistrés au laboratoire. Sur la figure 5.19.a, sont indiqués, le signal original et le spectrogramme. Le début t_1 et la fin t_2 de la perturbation ont été correctement détectés sur l'enveloppe de Shannon.

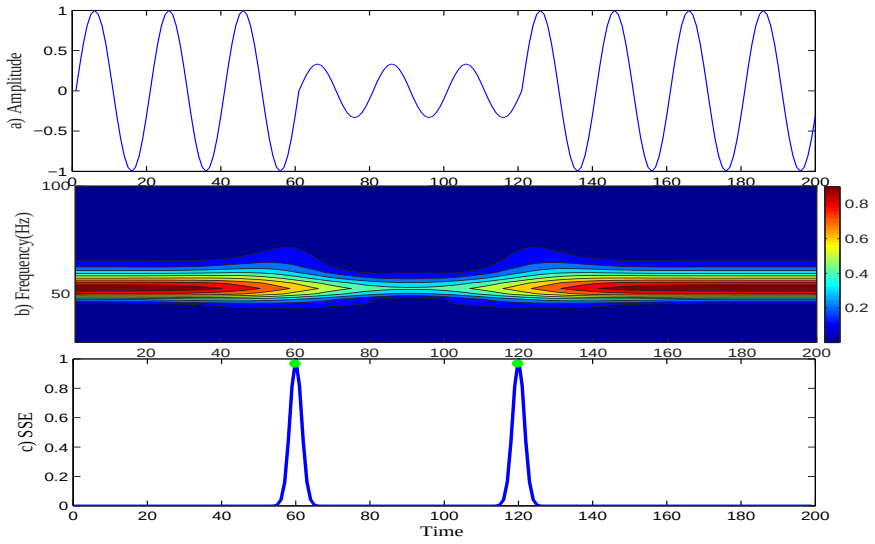


FIG. 5.14 – (a) Signal avec affaissement de tension ;(b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon.

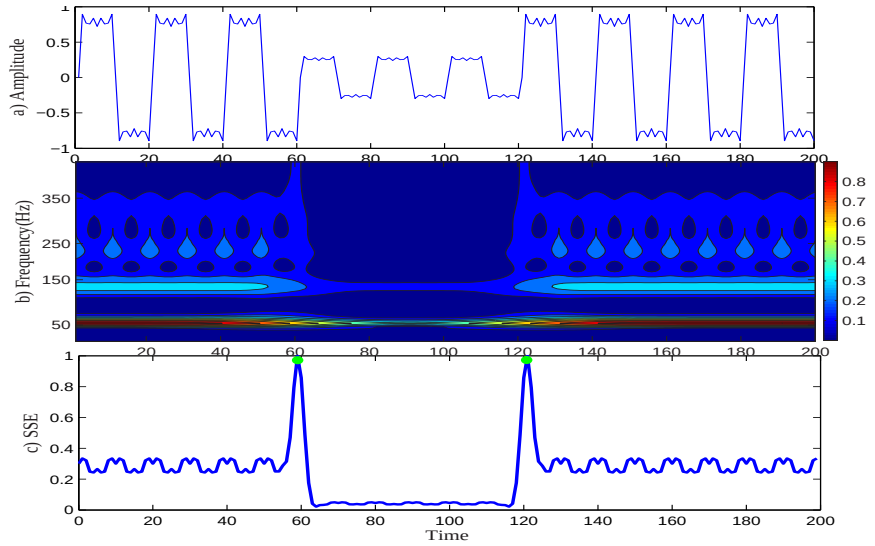


FIG. 5.15 – (a) Signal avec harmoniques et affaissement de tension ;(b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon.

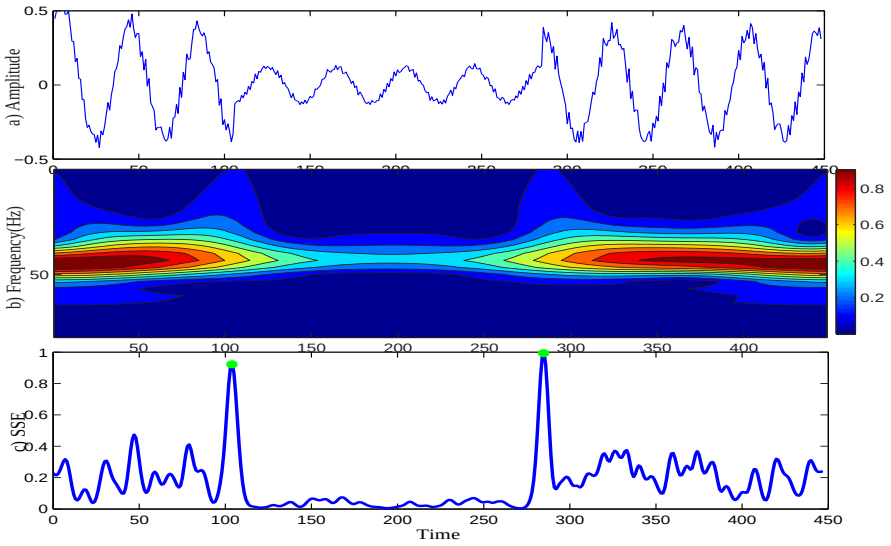


FIG. 5.16 – (a) Signal réel avec affaissement de tension ;(b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon.

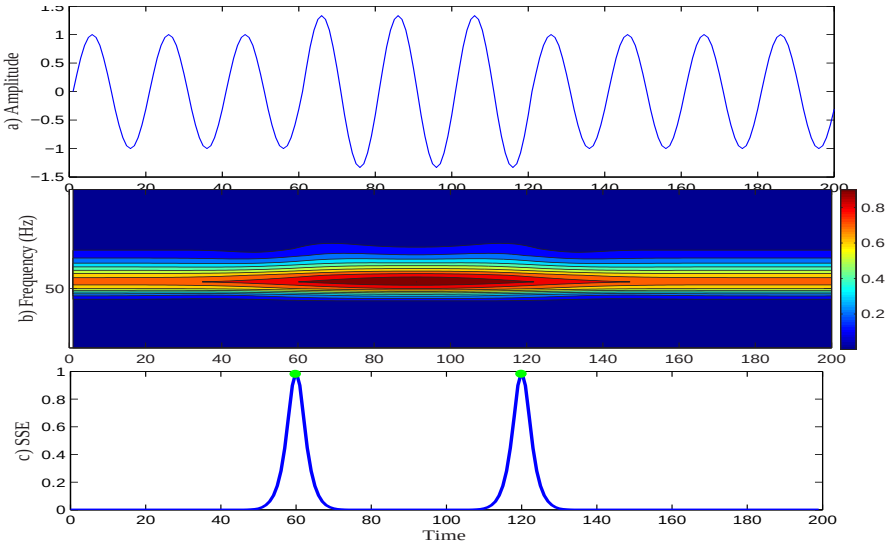


FIG. 5.17 – (a) Signal avec houle de tension ;(b) Spectrogramme du signal ; (c) Détection des instants t_1 et t_2 sur l'enveloppe de Shannon.

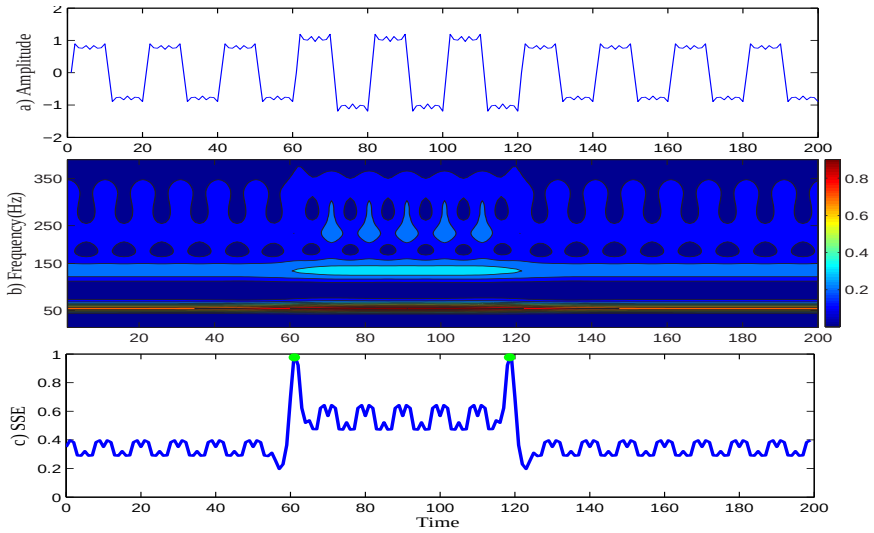


FIG. 5.18 – (a) Signal avec harmoniques et houle de tension ;(b) Spectrogramme du signal ; (c) Detection des instants t_1 et t_2 sur l'enveloppe de Shannon.

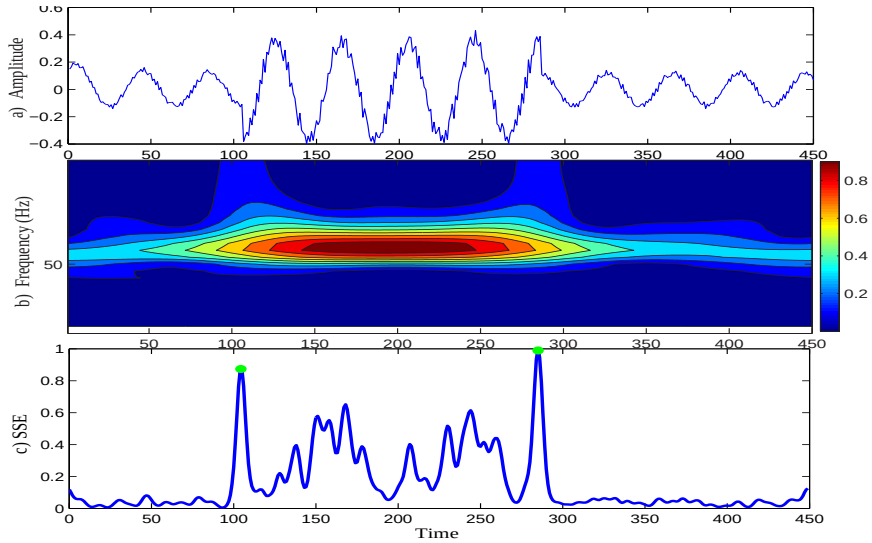


FIG. 5.19 – (Signal réel avec houle de tension ;(b) Spectrogramme du signal ; (c) Détection des instants t_1 et t_2 sur l'enveloppe de Shannon.

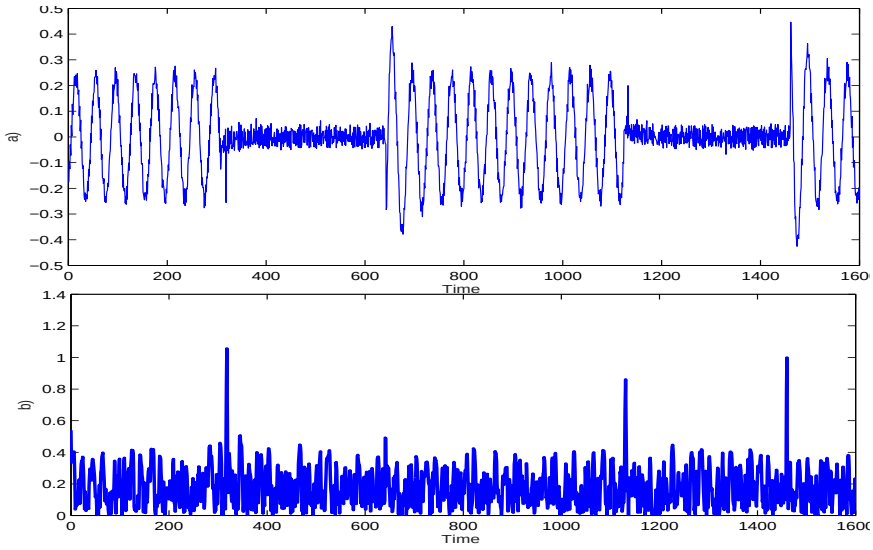


FIG. 5.20 – a) Signal avec interruption instantanée ; b) Coefficients d'ondelettes à l'échelle 1 " Morlet"

5.3.2.4 Comparaison avec la Transformée en ondelettes

La Transformée en ondelettes (CWT) est un outil mathématique qui permet de représenter les données sur le plan temps-echelle. Elle permet une analyse multi-résolution. Les avantages de cette transformée ont attiré l'attension des chercheurs dans plusieurs domaines, notamment lorsqu'il s'agit de détecter des évenements ou des singularités dans les signaux [Lin 2006, Kanitpanyacharoean 2004, Santoso 1996].

Malgré les différents avantages et la fléxibilité de cette transformée, le choix de l'ondelette mère et la détermination des niveaux de décomposition restent toujours un problème pour l'utilisateur. En guise de comparaison, nous avons effectué deux tests sur un signal réel sévèrement bruité en utilisant respectivement la méthode proposée autour de la ST et la CWT. les figures 5.13 et 5.20 montrent les résultats respectifs obtenus. La figure 5.20 montre clairememnt que la CWT est plus sensible aux bruits et quelle est moins efficace pour cette application.

5.4 Conclusion

Les résultats que nous avons exposés et commenté à travers ce chapitre paraissent convaiquants quant à la construction d'un classifieur visant à minimiser le coût de la classification. Nous avons obtenu un coût minimal pour un taux de rejet de 1.8%. Ceci est très satisfaisant car un classifieur qui rejette trop est loin d'être

le plus désiré.

Un autre travail basé sur des données industriels à été réalisé. Notre contribution [**Amirou** 2014a] réside dans l'introduction d'une nouvelle méthode de localisation des perturbations électriques. Elle est basé sur la ST et l'énergie de Shannon. Ceci sera suivi de la caractérisation de ces perturbation en vue de leur identification.

Chapitre 6

Conclusion et perspectives

Ce travail de thèse s'inscrit dans le domaine de la reconnaissance de formes. Il rentre dans le cadre d'un projet de recherche ayant abouti à plusieurs résultats déjà publiés dans des journaux. A ce projet de recherche, ont participé des enseignants chercheurs de l'université de Tizi-Ouzou (UMMTO) et des enseignants de l'université de Haute Alsace, Laboratoire MIPS où j'ai finalisé ma thèse.

L'objectif est d'optimiser le coût de la classification par SVMs afin d'améliorer le degré de confiance lors de la reconnaissance d'anomalies cardiaques d'une part et la détection de perturbations dans les réseaux électriques d'autre part.

Dans cette thèse, nous avons commencé par donner les principales notions d'apprentissage et d'aides à la décision. Parmi les algorithmes d'apprentissage, nous nous sommes particulièrement focalisés sur les SVMs, algorithme ayant fait ses preuves en termes de performances en discrimination et en pouvoir de généralisation. Ils représentent l'un des modèles de classification à avoir le plus marqué la reconnaissance de formes en fournissant un cadre théorique statistique. Ce classifieur est à même de prédire des frontières de décision complexes grâce à l'introduction de noyaux non linéaires.

Cette méthode de classification introduite par Vladimir Vapnik en 1995 est basée sur la recherche d'un hyperplan qui sépare au mieux les deux classes de données tout en maximisant la marge entre elles et en commettant un minimum d'erreurs. Ceci revient alors à résoudre un problème d'optimisation sous contraintes. Ce qui fait le point fort des SVMs est surtout l'introduction de l'astuce du noyau dans le cas de donnée non linéairement séparables. Ces dernières sont projetées dans un espace de plus grande dimension ou elles deviennent linéairement séparables. Ce processus de projection des données dans l'espace induit par le noyau est équivalent à une extraction de caractéristiques systématique sur les données d'entrée. Il permet d'augmenter ou de diminuer la capacité du classifieur. La dimension de cet espace augmenté n'est pas connue et est éventuellement infinie pour certains noyaux et certaines valeurs de leurs paramètres.

Différents algorithmes de résolution du problème d'optimisation d'un SVM ont été donnés. L'accent a cependant été mis sur l'algorithme simpleSVM. Ce dernier a été retenu et modifié pour y introduire l'option de rejet d'ambigüités.

En effet, malgré le pouvoir de généralisation pour lequel les SVMs sont connus, le taux d'erreur n'est jamais nul, surtout qu'un compromis entre le taux d'erreurs et la complexité du modèle doit toujours être fait. Par ailleurs, dans certaines applications biomédicales ou industrielles, certaines erreurs de classification doivent être sévèrement pénalisées.

Dans cet ordre d'idées, un classifieur ne devrait pas être évalué uniquement à travers son taux d'erreur mais aussi à travers son coût de classification. En cherchant à minimiser ce coût, la notion de rejet a été introduite pour la première fois par C.K.Chow en 1957 en faisant un compromis erreur-rejet.

En inscrivant nos travaux dans cet axe là, nous visons à construire une règle de décision qui rejette les exemples ambigus, c'est-à-dire ceux dont la probabilité d'appartenance à une classe est autour de 0.5. Nous sommes alors partis d'un SVM standard auquel nous avons modifié la fonction de coût structurel pour passer d'une fonction " hinge loss " vivement recommandée par V.Vapnick car elle conduit à une solution consistante, à une fonction "double hinge loss". Notons que la complexité algorithmique du nouveau problème est similaire à celle du SVM standard.

Dans cette fonction de perte, les seuils de probabilité sont flexibles. La règle de décision obtenue est alors sensible au coût des erreurs qui peuvent être symétriques ou asymétriques. Les seuils de rejet suivent alors le coût d'un exemple rejeté et le coût d'un exemple mal classé. L'objectif d'une telle approche est évidemment l'optimisation du coût global de la classification.

Des résultats très encourageants sont obtenus lorsque nous avons appliqué cette méthode à des domaines où l'option de rejet est très significative. Une première contribution concerne la classification de battements cardiaques par SVMs avec rejet d'ambiguïtés [Zidelmal 2009, Zidelmal 2012] puis un classifieur symétrique sensible au coût [Zidelmal 2013]. Pour reconnaitre des perturbation dans le réseau électrique, la première étape concernant leur localisation a été faite en utilisant la transformée de Stockwell [**Amirou** 2014a]. Quant à leur classification, elle est en cours de réalisation. Par ailleurs une étude comparative entre classifieurs SVMs avec rejet a été menée [**Amirou** 2014b]. Cette étude concerne les algorithmes basés sur une fonction de perte à double charnière "*double hinge loss*". Dans ce travail, nous avons détaillé les étapes de position du problème d'optimisation et de sa résolution. Dans chaque étape, sont également ressortis les points de comparaison. Ces différents classifieurs considérés sont testés sur plusieurs bases d'étude. Leurs performances respectives ont été particulièrement évaluées à travers le compromis erreur-rejet et le coût de la classification en fonction du rejet. Pour une meilleure lisibilité de nos résultats, nous avons choisi leur représentation graphique. La mise en évidence de l'option de rejet a été obtenue par la projection de l'espace d'entrée sur un espace à deux dimensions. Ceci a permis une visualisation effective de la classification et notamment de la zone de rejet.

Le deuxième volet consiste à localiser puis à caractériser des perturbations dans

le réseau électrique afin des les identifier Durant mon séjour au laboratoire MIPS, on a entamé l'autre volet de ce travail qui consiste à étudier les perturbation des signaux électrique.consiste à localiser puis à caractériser des perturbations dans le réseau électrique afin des les identifier

On a étudié en particulier les principales perturbations telles que : les variations de tension et les harmoniques et inter-harmoniques (de courant et de tension). On s'est intéressé également aux origines et aux symptômes ainsi qu'aux conséquences de ces perturbations. Dans la partie détection des événements, nous avons travaillé sur la Transformée de Stockwell qu'on a appliqué à des signaux réels issus d'une plateforme développée au laboratoire. De très bons résultats sont obtenus et comparés à ceux obtenus avec d'autres méthodes.

Perspectives : Bien entendu, les approches que nous avons proposées sont encore largement perfectibles. Ce travail s'ouvre sur plusieurs perspectives de travaux de recherche.

- Pour améliorer la qualité du classifieur, une perspective intéressante consisterait à revenir au niveau du processus d'extraction de caractéristiques. L'objectif serait alors de déterminer des ensembles de caractéristiques optimisés. L'introduction d'un nouveau noyau à support compact est également envisagée (travail en cours).
- Après la localisation automatique de la perturbation, une signature peut être attribuée à chaque appareil utilisé afin de construire les vecteurs caractéristiques. Pour ce faire, nous avons entammé la construction d'une base d'études au laboratoire MIPS. Cette base doit regrouper différents enregistrements de signaux perturbés par différents appareils à identifier. Il serait possible d'identifier des éléments polluants dans un réseau électrique domestique ou industriel.
- Suite à nos échanges avec le Laboratoire MIPS, un nouveau projet de recherche est né. Ce projet est financé par la région Alsace et quelques entreprises comme SOCOMEC. L'un de ses objectifs est la conception d'un prototype, implémenté sur un FPGA, pour l'identification et la classification des signaux électriques et biologiques. Ce projet démarrera en janvier 2015.

Annexe A

Différentes méthodes d'apprentissage d'un SVM

Ils existent plusieurs algorithmes de résolution du problème quadratique convexe. Parmis elles deux methodes sont présentées :

- Méthode de minimisation séquentielle (SMO) :[Platt 1998], résout à chaque itération un PQ à 2 variables dont la solution est calculée analytiquement.
- Méthode adaptée de résolution développée par R. Gabasov et al.[Gabasov 1995]

Nous presontons d'abord la formulationnn du prolème primal, la formulationnn du prolème dual et le lagrangien ansi que Les conditions d'optimalité de "KKT" du problème.

A.0.1 Formulation primale du problème

$$(Primal\ SVM)\ \begin{cases} \min\limits_{w,b,\xi} \dfrac{1}{2}||w||^2 + C\sum\limits_{i=1}^{n}\xi_i, \\ y_i[w'\phi(x_i)+b] \geq 1-\xi_i, i=1,\cdots,n \\ \quad \xi_i \geq 0, \quad i=1,\cdots,n, \end{cases} \tag{A.1}$$

Vue la possibilité d'une grande dimension du vecteur $w \in \mathbb{F}$, on résout généralement son problème dual associé.

A.0.2 Formulation duale du problème

$$(Dual\ SVM)\ \begin{cases} \min\limits_{\alpha\in\mathbb{R}^n} & \dfrac{1}{2}\alpha^T G\alpha - e^T\alpha \\ Sc & y^T\alpha = 0 \\ & 0 \leq \ \alpha_i \ \leq C, \qquad\qquad i=1,\ldots,n \end{cases} \tag{A.2}$$

le lagrangien s'écrit :

$$\mathcal{L} = \frac{1}{2}\|f\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(f(x_i)+b)-1+\xi_i) - \sum_{i=1}^{n}\beta_i\xi_i \tag{A.3}$$

$$\text{stationarité} \qquad \begin{aligned} f(x)^* - \sum_{i=1}^{n} \alpha_i^* y_i k(x_i, x) &= 0 \\ \sum_{i=1}^{n} \alpha_i^* y_i &= 0 \\ C - \alpha_i^* - \beta_i^* &= 0 \end{aligned} \tag{A.4}$$

$$\text{complementarité} \qquad \begin{aligned} \alpha_i^*(y_i(f^*(x_i) + b^*) - 1 + \xi_i^*) &= 0 \qquad i = 1, \ldots, n \\ \beta_i^* \xi_i^* &= 0 \qquad i = 1, \ldots, n \end{aligned} \tag{A.5}$$

$$\text{admissibilité primal} \qquad \begin{aligned} y_i(f^*(x_i) + b^*) + \xi_i^* &\geq 1 \qquad i = 1, \ldots, n \\ \xi_i^* &\geq 0 \qquad i = 1, \ldots, n \end{aligned} \tag{A.6}$$

$$\text{admissibilité dual} \qquad \begin{aligned} \alpha_i^* &\geq 0 \qquad i = 1, \ldots, n \\ \beta_i^* &\geq 0 \qquad i = 1, \ldots, n \end{aligned} \tag{A.7}$$

où les $\alpha_i \geq 0$ et les $\beta_i \geq 0$ sont les multiplicateurs de Lagrange associés aux contraintes. Les conditions d'optimalité de Karush, Kuhn et Tucker du problème permettent de caractériser la solution du problème.

A.0.3 L'Algorithme SMO

L'algorithme SMO a été proposé par Platt [Platt 1998] pour résoudre le problème de programmation quadratique des SVMs. C'est un algorithme rapide. Il décompose le problème quadratique puis résoud le plus petit sous ensemble possible. Il optimise la fonction objectif duale du problème global en opérant à chaque itération sur un ensemble réduit à deux multiplicateurs de Lagrange. La puissance de cette procédure réside dans le fait que le problème d'optimisation dépendant uniquement de deux variables, peut être résolu analytiquement.

La contrainte $\Sigma_{i=1}^{l} y_i \alpha_i$ qui doit être verifiée a chaque itération implique que le plus petit nombre de multiplicateurs à optimiser dans chaque étape est de deux. A chaque fois qu'un multiplicateur est mis a jour, un autre multiplicateur au moins doit être ajusté afin de maintenir la contrainte précédente satisfaite.

A chaque étape, l'algorithme SMO choisit deux élements α_i et α_j et les optimise conjointement. Il détermine les valeurs optimales de ces deux coefficients tout en gardant les autres multiplicateurs fixes puis il met à jour le vecteur solution $\alpha = (\alpha_1, \quad \alpha_2, \quad ..., \quad \alpha_l)$ correspondant. Le choix des deux points x_i et x_j est réalisé à l'aide d'une heuristique, alors que l'optimisation de leurs multiplicateurs correspondants se fait analytiquement. l'étape de résolution analytique évite pas mal d'itérations emboîtées. En plus de ses performances en terme de temps de convergence, l'algorithme SMO n'est pas gourmand en espace mémoire vu qu'il n'utilise pas d'opérations sur la totalité de la matrice de Gram. Le seul inconvénient de cette méthode est son critère d'arrêt basé sur les conditions de KKT et qui n'est

pas toujours facile à contrôler. À l'heure actuelle, cette méthode est la plus courante pour appliquer les SVM à des problèmes de grande taille.

Algorithme

- Si $\alpha_i = 0$ Alors $C - \alpha_i^* - \beta_i^* = 0 \Rightarrow \beta_i^* = C - \alpha_i^* = C$ on aura $\xi_i = 0$ donc $y_i(f^*(x_i) + b^*) \geq 1$

- Si $0 < \alpha_i < C$ Alors $C - \alpha_i^* - \beta_i^* > 0 \Rightarrow \beta_i^* = C - \alpha_i^* > 0$ on aura $\xi_i = 0$ donc $y_i(f^*(x_i) + b^*) = 0$

- Si $\alpha_i = C$ Alors $C - \alpha_i^* - \beta_i^* > 0 \Rightarrow \beta_i^* = C - \alpha_i^* = 0$ on aura $\xi_i \geq 0$ donc $y_i(f^*(x_i) + b^*) - 1 \leq 0$

La table (A.1) indique la répartition de l'ensemble d'apprentissage en trois sous ensembles définis par les contraintes du problème (A.3).

TAB. A.1 – Répartition des exemples d'apprentissage

I_0	$\alpha_i = 0$	$I_0 = i \vert y_i(f(x_i) + b) - 1 \geq 0$
I_α	$0 < \alpha_i < C$	$I_\alpha = i \vert y_i(f(x_i) + b) - 1 = 0$
I_C	$\alpha_i = C$	$I_C = i \vert y_i(f(x_i) + b) - 1 < 0$

On pose

$$\mathrm{R}_i = y_i(f(x_i) + b) - 1$$

$$\mathrm{R}_i = y_i(f(x_i) + b) - y_i^2$$

$$\mathrm{R}_i = y_i((f(x_i) + b) - y_i)$$

avec

$$\mathrm{E}_i = (f(x_i) + b) - y_i$$

$$\mathrm{E}_i = \mu_i - y_i$$

on aura

$$\mathrm{R}_i = y_i \mathrm{E}_i$$

Si les conditions de KKT (A.4, A.5) sont verifiées alors :

$$\begin{cases} \alpha_i = 0 & \mathrm{R}_i \geq 0 \\ 0 < \alpha_i < C & \mathrm{R}_i \simeq 0 \\ \alpha_i = C & \mathrm{R}_i \leq 0 \end{cases} \tag{A.8}$$

Les conditions (A.8) sont violées dans les deux cas suivants :

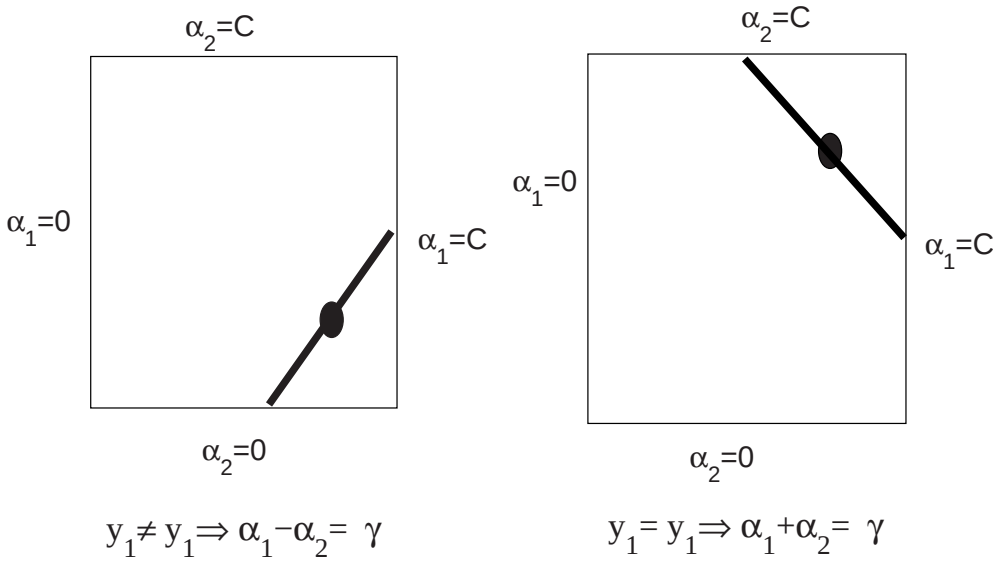
$$\begin{cases} \alpha_i < C & et \quad \mathrm{R}_i < 0 \\ \alpha_i > C & et \quad \mathrm{R}_i > 0 \end{cases} \tag{A.9}$$

Optimiser deux α_i

Afin de résoudre le problème A.10. L'algorithme SMO recherche dans la région réalisable du problème dual et optimise la fonction objectif

$$\begin{cases} \max\limits_{\alpha} \mathcal{L}_D = \sum_{i=1}^{n} \alpha_i - \dfrac{1}{2} \displaystyle\sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \\ \displaystyle\sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, n \end{cases} \tag{A.10}$$

Il calcule d'abord deux multiplicateurs de Lagrange à la fois, (les autres sont fixés). Il utilise des heuristiques pour choisir les deux α_i. Il optimise ensuite la fonction objectif à chaque étape. Les contraintes A.10, placent les α_i sur une ligne diagonale à l'intérieur du carré.



Par conséquent, une étape de l'algorithme SMO consiste à trouver un optimum sur le segment de droite délimité par le carré et qui est fonction de α_1 et α_2 avec :

$$\alpha_1 y_1 + \alpha_2 y_2 = \gamma$$

où

$$\gamma = -\sum_{i=3}^{n} \alpha_i y_i$$

$$\gamma = \alpha_1^{old} + s\alpha_2^{old}$$

$$s = y_1.y_2$$

L'algorithme détermine d'abord l'intervalle où peut varier α_2, puis calcule sa valeur. Nous avons deux cas d'optimisation selon que y_1 et y_2 soient de même signe ou non.

– Si $y_1 \neq y_2$ alors les limites de α_2 sont :

$$L = max(0, \alpha_2^{old} - s\alpha_1^{old})$$

$$H = min(C, C + \alpha_2^{old} + s\alpha_1^{old})$$

– Si $y_1 = y_2$ alors les limites de α_2 sont :

$$L = max(0, \alpha_1^{old} + s\alpha_2^{old} - C)$$

$$H = min(C, \alpha_1^{old} + s\alpha_2^{old})$$

En ne permettant qu'à deux multiplicateurs de Lagrange de varier, nous pouvons à present calculer en permettant uniquement le changement de deux multiplicateurs.

$$\mathcal{L}_D = \frac{1}{2}\eta\alpha_2^2 + (y_2(\mathrm{E}_1^{old} - \mathrm{E}_2^{old}) - \eta\alpha_2^{old})\alpha_2 + const \tag{A.11}$$

avec :

$$\eta = 2k(x_1, x_2) - k(x_1, x_1) - k(x_2, x_2)$$

On définie E_i comme étant l'erreur d'entrainement du i^{me} exemple d'apprentissage

$$\mathrm{E}_i = \mu_i - y_i$$

où μ_i est la sortie fournie par le SVM à l'exemple(x_i, y_i)

$$\mu_i = \sum_{j=1}^{n} \alpha_j y_j x_j x_i$$

Les dérivées première et seconde de la fonction objectif $\mathcal{L}_D$ définie dans (A.11) par rapport à α_2 peut s'exprimer de la foçon suivante :

$$\frac{\partial \mathcal{L}_D}{\partial \alpha_2} = \eta\alpha_2 + (y_2(\mathrm{E}_1^{old} - \mathrm{E}_2^{old}) - \eta\alpha_2^{old}) \tag{A.12}$$

$$\frac{\partial^2 \mathcal{L}_D}{\partial^2 \alpha_2} = \eta \tag{A.13}$$

Noter que

$$\eta = 2k_{12} - k_{11} - k_{22} \leq 0$$

Preuve : Soit

$$k_{12} = x_1^T x_2, \quad k_{11} = x_1^T x_1 \quad et \quad k_{22} = x_2^T x_2.$$

Alors

$$\eta = 2k_{12} - k_{11} - k_{22}$$
$$-(x_2 - x_1)^T (x_2 - x_1) = -\|x_2 - x_1\|^2 \leq 0$$

Soit

$$\frac{\partial \mathcal{L}_D}{\partial \alpha_2} = 0$$
$$\Rightarrow \eta\alpha_2 + y_2(\mathrm{E}_1^{old} - \mathrm{E}_2^{old}) - \eta\alpha_2^{old} = 0$$
$$\Rightarrow \alpha_2^{new} = -\frac{y_2(\mathrm{E}_1^{old} - \mathrm{E}_2^{old}) - \eta\alpha_2^{old}}{\eta}$$
$$\Rightarrow \alpha_2^{new} = \alpha_2^{old} + \frac{y_2(\mathrm{E}_1^{old} - \mathrm{E}_2^{old})}{\eta}$$

Une fois α_2^{new} est trouvé, il correspond au maximum sans contraintes. Comme il doit être limité par les points de la diagonale on utilise :

$$\alpha_2^{new,clipped} = \begin{cases} \mathrm{H} & Si \quad \alpha_2^{new} \geq \mathrm{H} \\ \alpha_2^{new} & Si \quad \mathrm{L} < \alpha_2^{new} < \mathrm{H} \\ \mathrm{L} & Si \quad \alpha_2^{new} \leq \mathrm{L} \end{cases} \tag{A.14}$$

Pour calculer α_1^{new}, nous utilisant l'équation suivante :

$$\alpha_1^{new} + s\alpha_2^{new,clipped} = \alpha_1^{old} + s\alpha_2^{old} \tag{A.15}$$

A présent, la valeur de α_1^{new} peut être calculée par :

$$\alpha_1^{new} = \alpha_1^{old} + s(\alpha_2^{old} - \alpha_2^{new,clipped}) \tag{A.16}$$

L'algorithme SMO optimise deux multiplicateurs de Lagrange à chaque étape de résolution. Le choix du premier multiplicateur se fait a traver la boucle externe de l'heuristique. Le choix du deuxième multiplicateur se fait à travers la boucle interne de l'heuristique en maximisant $|E_2 - E_1|$. On cherche séquentiellement un α_2 qui viole les conditions de KKT (A.9) en commencant par un indice pris au hasard. Après avoir trouvé les deux multilplicateur conjointement optimisés, on met à jour la valeur de la fonction objectif. L'algorithme SMO s'arrête lorsqu'il ne reste aucun multiplicateur de Lagrange qui viole les conditions de KKT.

Calcul du paramètre b

Le paramêtre b doit être calculé séparément à la fin. Soit μ_i la réponse du SVM avec les anciennes valeurs de α_1 et α_2.

$$\mu_1 = \alpha_1^{old} y_1^{old} k_{11} + \alpha_2^{old} y_2^{old} k_{12} + \sum_{j=3}^{n} \alpha_j y_j k_{1j} - b^{old}$$

Calcul de b_1 et b_2

$$b_1 = E_1 + y_1(\alpha_1^{new} - \alpha_1^{old})k_{11} + y_2(\alpha_2^{new,clipped} - \alpha_2^{old})k_{12} + b^{old}$$

$$b_2 = E_2 + y_1(\alpha_1^{new} - \alpha_1^{old})k_{12} + y_2(\alpha_2^{new,clipped} - \alpha_2^{old})k_{12} + b^{old}$$

La valeur de b^{new} est :

– Si $0 < \alpha_1 < C$

$$b^{new} = b_1$$

– Si $0 < \alpha_2 < C$

$$b^{new} = b_2$$

– Si α_1 et α_2 sont tous les deux égaux à C alors

$$b^{new} = \frac{b_1 + b_2}{2}$$

L'algorithme se déroule comme suit :

1 Trouver un multiplicateur de Lagrange α_1 qui viole les conditions Karush-Kuhn-Tucker (KKT) pour le problème d'optimisation.

2 Choisissez un deuxième multiplicateur α_2 et optimiser le couple (α_1,α_2).

3 Répétez les étapes 1 et 2 jusqu'à convergence.

Lorsque tous les multiplicateurs de Lagrange remplissent les conditions KKT (avec une tolérance définie par l'utilisateur), le problème est résolu. Bien que cet algorithme garantit la convergence, des heuristiques sont utilisées pour choisir la paire de multiplicateurs de façon à accélérer la vitesse de convergence.

A.0.4 La méthode adaptée

La méthode adaptée a été utilisée pour l'apprentissage des SVMs dans [Brahmi 2010]. Cette méthode est itérative, du type primal-dual. Elle est basée sur le principe du support et d'estimation de suboptimalité.

L'approche proposée dans [Brahmi 2010] est une extension de la méthode adaptée, développée par R. Gabasov et al.[Gabasov 1995] pour la résolution des problèmes de PL et PQ.

- Elle est formée d'une seule phase.
- Elle est itérative et génére une suite de points réalisables et dont la valeur de la fonction objectif croît à chaque itération.

Formulation primale des SVMs

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i, \tag{A.17a}$$

$$y_i[w'\phi(x_i) + b] \geq 1 - \xi_i, i = 1, \cdots, n \tag{A.17b}$$

$$\xi_i \geq 0, \quad i = 1, \cdots, n, \tag{A.17c}$$

Formulation duale

$$\max_{\alpha} -\frac{1}{2}\alpha' D\alpha + 1'\alpha, \tag{A.18a}$$

$$y'\alpha = 0, \tag{A.18b}$$

$$0 \leq \alpha \leq C, \tag{A.18c}$$

- $\alpha \in r^n$:multiplicateurs de Lagrange associés aux contraintes (A.17b) du problème primal ;
- y est n-vecteur des étiquettes, avec $y_i \in \{-1, +1\}$;
- 1 est n-vecteur dont chacune de ses composantes est égale à 1.

- $D \in r^{n\times n}$ et $D' = D \geq 0$, avec $D_{i,j} = y_i y_j k(x_i, x_j), \quad i, j = 1, .., n$ et $k(x_i, x_j) =< \phi(x_i)'\phi(x_j) >$, la fonction noyau.

Un ensemble d'indices $I_s \subset I = \{1, \cdots, n\}$ est appelé support du problème (A.18), si la matrice $D_{ss} = D(I_s, I_s)$ est non singulière.

Definition A.0.1
- *Le n-vecteur α vérifiant les contraintes (A.10) est dit solution réalisable (SR) du problème (A.10).*
- *Un point réalisable α^0 est dite solution optimale du problème si $f(\alpha^0) = \max f(\alpha)$, où α est pris parmi tous les points réalisables du problème (A.18).*
- *Un point réalisable α^ε est dit solution sub-optimale (ε-optimale) du problème (A.10) si $f(\alpha^0) - f(\alpha^\varepsilon) \leq \varepsilon$, où $\varepsilon \geq 0$ est précision choisie par l'utilisateur.*

- *Le n-vecteur* $\delta = D\alpha + by - 1$ *est appelé vecteur des coûts réduits associé à une SR du problème (A.10).*
- *La paire* $\{\alpha, I_s\}$ *est dite solution réalisable de support (SRS) si* α *vérifie les contraintes (A.18b)-(A.18c) et* $\delta_s = \delta(I_s) = 0$.
- *Une SRS* $\{\alpha, I_s\}$ *est dite non dégénérée, si elle vérifie :*

$$0 < \alpha_i < C, \forall i \in I_s. \tag{A.19}$$

Soit $I_n = I \backslash I_s$ *: L'ensemble d'indices hors-support. Alors, les différents vecteurs et la matrice D peuvent être partagés comme suit :*

$$\alpha = (\alpha_s, \alpha_n)', \ \alpha_s = \alpha(I_s), \ \alpha_n = \alpha(I_n); \quad y = (y_s, y_n)', \ y_s = y(I_s)\, y_n = y(I_n),$$

$$D = \begin{pmatrix} D_{ss} & D_{sn} \\ D_{ns} & D_{nn} \end{pmatrix}, \quad D_{sn} = D(I_s, I_n) = D(I_n, I_s)' = D'_{ns}.$$

A.0.4.1 Critères d'optimalité et de sub-optimalité

Theorem A.0.1 *(Critère d'optimalité) Une condition suffisante pour l'optimalité d'un point réalisable* α *est l'existence d'un support* I_s*, tels que :*

$$\begin{cases} \delta_i \geq 0, & si \quad \alpha_i = 0, \\ \delta_i \leq 0, & si \quad \alpha_i = C, \\ \delta_i = 0, & si \quad 0 \leq \alpha_i \leq C, \ i \in I_n. \end{cases} \tag{A.20}$$

Si de plus $\{\alpha, I_s\}$ *est non dégénérée, alors ses relations deviennent aussi nécessaire pour l'optimalité du point* α.

L'estimation de sub-optimalité :

$$\beta(\alpha, I_s) = f(\alpha^0) - f(\alpha) = \sum_{i \in I_n, \, \delta_i > 0} \delta_i \alpha_i + \sum_{i \in I_n, \, \delta_i < 0} \delta_i (\alpha_i - C) \tag{A.21}$$

mesure l'écart entre la valeur de la fonction objectif à l'optimum α^0 et une autre SRS $\{\alpha, I_s\}$. Le théorème suivant nous permet d'obtenir le critère de sub-optimalité :

Theorem A.0.2 *(critère de sub-optimalité) Soit* $\{\alpha, I_s\}$ *une SRS et* $\varepsilon \geq 0$ *choisi à l'avance. Si* $\beta(\alpha, I_s) \leq \varepsilon$*, alors* α *est une solution* ε*-optimale du problème (A.18).*

Cette méthode est basée sur le concept du support et utilise le critère de sub-optimalité comme critère d'arrêt. L'algorithme de la méthode proposée est fini, si durant le processus de résolution, le nombre de supports (bases) dégénérés est fini. A chaque itération, la diminution de l'estimation de sub-optimalité est assurée par la méthode proposée : [Brahmi 2010].

Algorithme 5 Algorithme méthode adaptée

(0) Initialisation : Soit $Is = \{1\}$ un support pour le problème (A.18).

– Démarrer avec une SRS triviale : $\alpha = 0, \quad b = y_1$.

– Calculer le vecteur des coûts réduits : $\delta = (\delta_s, \delta_n)'$:

$$\delta_s = 0, \quad \delta_n = D(I_n, I)\alpha + by_n - 1.$$

– Calculer l'estimation de sub-optimalité :

$$\beta(\alpha, I_s) = \sum_{i \in I_n,\, \delta_i > 0} \delta_i \alpha_i + \sum_{i \in I_n,\, \delta_i < 0} \delta_i (\alpha_i - C) \tag{A.22}$$

✓ Si $\beta(\alpha, I_s) \leq \varepsilon$ alors α est une solution ε-optimale pour le PQ (A.18). Arrêter l'algorithme.

✓ Sinon Aller à (1).

(1) Verification de l'optimalité de la SRS $\{\alpha, I_s\}$:

– Si les relations (A.20) sont vérifiées alors α est une solution optimale pour (A.18), on arrête l'algorithme ;

– Sinon, aller à (2).

(2) **Construction d'une nouvelle solution :**

$$\overline{\alpha} = \alpha + \theta l, \quad \overline{\delta} = \delta + \theta t, \quad \overline{b} = b + \theta r, \text{ avec } t = Dl + ry. \tag{A.23}$$

– Calculer les directions d'amélioration l et t :

$$\begin{cases} l_i = -\alpha_i, & \text{si } \delta_i > 0, \\ l_i = C - \alpha_i, & \text{si } \delta_i < 0, \\ l_i = 0, & \text{si } \delta_i = 0,\ i \in I_n, \\ l_s = -D_{ss}^{-1} \left[D_{sn} l_n + r y_s \right]. \end{cases} \tag{A.24}$$

$$\begin{cases} r = l_n'(y_n - D_{ns} D_{ss}^{-1} y_s) / y_s' D_{ss}^{-1} y_s, \\ t_s = 0, \\ t_n = D(I_n, I) l + r y_n. \end{cases} \tag{A.25}$$

Algorithme 6 Algorithme méthode adaptée (suite)

– Déterminer le pas optimal $\theta = min\{1, \theta_{i_1}, \sigma_{i_0}\}$: où σ_{i_0} est déterminé de manière à ce que δ_n et $\bar{\delta}_n$ ne changent pas de signe.

$$\sigma_{i_0} = \min_{i \in I_n} \sigma_i, \text{ avec } \sigma_i = \begin{cases} -\dfrac{\delta_i}{t_i}, & \text{if } \delta_i t_i < 0, \\ 0, & \text{si } \delta_i = 0 \text{ et } t_i \neq 0, \\ +\infty, & \text{sinon , } i \in I_n. \end{cases} \tag{A.26}$$

$$\theta_{i_1} = \min_{i \in I_s} \theta_i, \theta_i = \begin{cases} -\alpha_i / l_i, & \text{si } l_i < 0, \\ (C - \alpha_i)/l_i, & \text{si } l_i > 0, \\ +\infty, & \text{si } l_i = 0, \ i \in I_n. \end{cases} \tag{A.27}$$

– Calculer l'estimation de sub-optimalité de la nouvelle SRS $\{\bar{\alpha}, I_s\}$:

$$\beta(\bar{\alpha}, I_s) = (1 - \theta) \left(\beta(\alpha, I_s) - \theta l' D l \right). \tag{A.28}$$

✓ Si $\theta = 1$, alors $\bar{\alpha} = \alpha + l$ est optimal pour le problème (A.18). Arrêter l'algorithme.
✓ Si $\beta(\bar{\alpha}, I_s) \leq \varepsilon$, alors $\bar{\alpha}$ est une solution ε-optimale pour le problème. Arrêter l'algorithme.
✓ Si $\theta < 1 \vee \beta(\bar{\alpha}, I_s) > \varepsilon$, aller à (3).

(3) Changement de support :

1. Pour $\theta = \theta_{i_1}, i_1 \in I_s$, on a $\bar{\alpha}_{i_1} = 0 \vee C$:

$$\bar{I}_s = I_s \backslash i_1.$$

2. Pour $\theta = \sigma_{i_0}, i_0 \in I_n$, on a $\bar{\delta}_{i_0} = 0$:

$$\bar{I}_s = I_s \cup i_0.$$

Aller à **(1)**.

Annexe B

Bilan scientifique relatif à la thèse

---

1. **A.Amirou**, D.Ould-Abdeslam, Z.Zidelmal, M.Aidene and J.Merclé " S-transform and shannon energy for electrical disturbances detection", the 40th Annual Conference of the IEEE Industrial Electronics Society IECONŠ 2014, Dallas, TX-USA (2014).
2. **A.Amirou**, D.Ould-Abdeslam, Z.Zidelmal, M.Aidene and J.Merclé " Optimizing the classification cost using SVMs with a double hinge loss", Informatica, An international Journal of Computing and Informatics, Volume 38 Number 2, pp : 125-133, June 2014.
   http ://www.informatica.si/Vol38htm $\#No2$,
   ISSN : 0350-5596, Premier Volume :1977
   Facteur d'impact : 1.117
   Indéxé dans : Scopus, Zentralblatt Math, ACM Digital Library, Citeseer, COBISS, Compendex, Computer δ Information Systems Abstracts, Computer Database, Computer Science Index, MatSciNet, MatSci,
3. Z.Zidelmal, **A.Amirou**, D.Ould-Abdeslam, A.Moukadem δ A.Dieterlen "QRS detection using S-Transform and Shannon Energy", Elsevier, Computer Methods and Programs in Biomedicine Volume116, Issue 1, Pages 1-9, August 2014
   http ://www.cmpbjournal.com/article/S0169-2607(14)00147-3/
   ISSN : 0169-2607, Premier Volume :1985
   Facteur d'impact : 1.589 (Thomson reuters) Indéxé dans : Scopus, Embase, Elsevier BIOBASE, INSPEC, Science Search, Computer Science Index, Science Citation Index,...
4. **A.Amirou**, D.Ould-Abdeslam, M.Aidene , Z.Zidelmal and J.Mercle, "Using S-Transform and Shannon energy for electrical disturbances detection", Springer, Signal, Image and Video processing (SIVP) , Article en révision
   http ://www.springer.com/engineering/signals/journal/11760
   ISSN : 1863-1711 ; Premier volume : 2007 Facteur d'impact : 0.409
   Indexé dans : Science Citation Index Expanded (SciSearch), Journal Citation Reports/Science Edition, SCOPUS, INSPEC, Zentralblatt Math, Google Scholar, Academic OneFile, Current Contents/Engineering, Computing and Technology, DBLP, EI-Compendex, Expanded Academic, io-port.net, OCLC, SCImago, Summon by ProQuest

5. Z . Zidelmal, **A.Amirou**, M.Adnane and A.Belouchrani "QRS detection based on wavelet coefficients", Elsevier, Computer Methods and Programs in Biomedicine (CMPB), Vol 107, Issue 3 , pp : 490-496, 2012.
   http ://www.journals.elsevier.com/computer-methods-and-programs-in-biomedicine/
   ISSN : 0169-2607 Facteur d'impact : 1.589 (Thomson reuters)
   Indéxé dans : Scopus, Embase, Elsevier BIOBASE, INSPEC, Science Search, Computer Science Index, Science Citation Index,...
6. Z . Zidelmal, **A.Amirou** and A.Belouchrani "Heartbeat classification using Support Vector Machines (SVMs) with an embedded reject option", International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), Volume 26, Number 1, February 2012.
   http ://www.worldscientific.com/toc/ijprai/26/01
   Facteur d'impact : 0.669 (Thomson reuters)
   Indéxé dans plusieures bases : Scopus, INSPEC, Compendex, SciSearch, CompuMath, Computer Abstracts, DBLP Computer Science Bibliography.
7. Z.Zidelmal, **A.Amirou**, D.Ould-Abdeslam, J.Merckle, "ECG beat classification using a cost sensitive classifier", Elsevier, Computer Methods and Programs in Biomedicine (CMPB),, Vol. 111, Issue 3, Pages 570-577, (2013).
   http ://www.journals.elsevier.com/computer-methods-and-programs-in-biomedicine/
   ISSN : 0169-2607 , Premier Volume :1985
   Facteur d'impact : 1.589 (Thomson reuters) Indéxé dans : Scopus, Embase, Elsevier BIOBASE, INSPEC, Science Search, Computer Science Index, Science Citation Index,...

Articles dans des proceedings

8. Z.Zidelmal, **A. Amirou**, " Application des SVMs basés sur l'algorithme SMO pour la détection d'anomalies cardiaques ", 4rth International Conference : Sciences of Electronic,Technologies of Information and Telecommunications, 25-29 Mars, 2007 - TUNISIE.
9. **A.Amirou**, M.Aidene, and M.Ouanes " Implémentation de l'algorithme du simplexe dans le cas d'un problème d'optimisation semi-infini " COSI'08 Tizi Ouzou.
10. Z.Zidelmal , **A.Amirou** & A.Belouchrani, "Using Support Vector Machines (SVMS) With Reject Option For Heartbeat Classification", INSTIC, Biosignals, PORTO (Portugal), 2009.

Bibliographie

[Aha 1995] R. L. Aha D. W. et Bankert. *A comparative evaluation of sequential feature selection algorithms.* In 5th International Workshop on Artificial Intelligence and Statistics, pages 1–7, 1995. 12

[Aizermann 1964] E. et Rozoener L. Aizermann M. Bravermann. *Théoretical fondation of the potentiel function method pattern recongition learning.* automation and remote control, vol. 25, pages 821–837, 1964. 41

[Anquetil 1999] Éric Bertrand Couasnon Anquetil et Frédéric Dambreville. *A symbol classifier able to reject wrong shapes for document recognition systems.* In Proceedings of 3rd IAPR International Workshop on Graphics Recognition (GRECŠ99), pages 195–202, 1999. 75

[Argyriou 2006] Charles A. Micchelli Argyriou Andreas Raphael Hauser et Massimiliano Ponti. *A dc-programming algorithm for kernel selection.* ICML, 2006. 69

[Bach 2005] Francis David Heckerman Bach et Eric Horvitz. *In Robert G. Cowell and Zoubin Ghahramani.* Society for Artificial Intelligence and Statistics, pages 9–16, 2005. 69

[Bartlett 2008] P. L. Bartlett et M. H. Wegkamp. *Classification with a reject option using a hinge loss.* Department of Statistics, Florida State University, no. M980, pages 709–721, 2008. 4, 15, 74, 82, 83, 84, 85, 88, 98, 99, 100, 101, 102, 103, 104

[Bernhard 2002] Scholkopf Bernhard et Alexander J. Smola. *Learning with Kernels.* MIT Press, 2002. 17

[Boatas 2000] B. Dubuisson Boatas A. et M.A. Dillies-Peltier. *A new statistical pattern recognition distance rejection model : application to the monitoring of car catalytic converters.* In Systems, Man, and Cybernetics, IEEE International Conference on, vol. 4, page 2839Ũ2844, 2000. 74

[Boser 1992] B. E. I. Guyon Boser et V. N. Vapnik. *A training algorithm for optimal margin classifier.* Workshop of Computational Learning Therory, vol. 144-152, 1992. 15, 36, 41, 54

[Brahmi 2010] B. Brahmi et M. O. Bibi. *Dual support method for solving convex quadratic programs.* Optimization, bfseries, vol. 6, no. 59, pages 851–872, 2010. 130, 131

[Breiman 1984] L. et al. Breiman. *Classification and Regression Trees.* Chapman and Hall, New York, 1984. 30

[Broadbent 1988] J. Broadbent H. A. et Lucas. *patterns using and autoassociative neural network topology.* Neural network model of serial learning. SIGAPL APL Quote Quad, vol. 19, pages 54–61, 1988. 32

[Burges 1998] C. Burges et J. Christopher. *A tutorial on support vector machines for pattern recognition.* Data Mining and Knowledge Discovery, vol. 2, pages 121–167, 1998. 20

[Chen 2003] X.-w. Chen. *An improved branch and bound algorithm for feature selection.* Pattern Recognition Letters, IEEE Pattern Analysis and Machine Intelligence, vol. 12, no. 24, pages 1925–1933, 2003. 15

[Chow 1957] C. K. Chow. *An Optimum Character Recognition System Using Decision Function.* IRE Trans. Electronic Computers, vol. EC-6, no. 4, pages 247–254, 1957. 74

[Chow 1970] C. K. Chow. *On optimum recognition error and reject tradeoff.* IEEE Trans. on Information Theory, vol. 16, no. 41, pages 41–46, 1970. 15, 73, 74, 75, 77, 78, 79, 100, 103

[Christopher 1998] J. Christopher et C. Burges. *A tutorial on support vector machines for pattern recognition.* Data Mining and Knowledge Discovery, pages 121–167, 1998. 17

[Cortes 1982] C. Cortes et V. Vapnik. *Support-vector networks.* Machine Learning, vol. 3, no. 20, pages 273–297, 1982. 15, 41, 43, 50

[Cristianini 2000] N. Cristianini et J. Shawe-Taylor. *Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Pres, pages 121–167, 2000. 15

[Dietz 1989] Kiech E. L. et Ali M. Dietz W. E. *Classification of data patterns using and autoassociative neural network topology.* In Proceedings of the 2nd international conference on Industrial and engineering applications of artificial intelligence and expert systems , New York, NY, USA. ACM., vol. 2, pages 1028–1036, 1989. 32

[Dubuisson 1993] B. Dubuisson et M. Masson. *A statistical decision rule with incomplete knowledge about classes.* Pattern Recognition, vol. 26, no. 1, pages 155Ű–165, 1993. 75, 77

[Duda 2000] Peter E. Hart David G. Stork Duda Richard O. *Pattern Classification, 2nd Edition.* SVM Books, 2000. 41

[Frélicot 1995] C. M.H. Masson Frélicot et B. Dubuisson. *Reject options in fuzzy pattern classification rules.* In Proc. of 3rd European Congress on Intelligent Techniques and Soft Computing, 1995. 75

[Fumera 2000] Giorgio Fabio Roli Fumera et Giorgio Giacinto. *Reject option with multiple thresholds.* Pattern Recognition, vol. 33, no. 12, pages 2099–2101, 2000. 75, 78, 79

[Fumera 2002] G. Fumera et F. Roli. *Support Vector Machines with Embedded Reject Option.* Pattern Recognition with Support Vector Machines : First International Workshop, pages 68–82, 2002. 74

[Gabasov 1995] F. Gabasov et F. M. Kirillova. *Methods of Optimization (Translations Series in Mathematics and Engineering.* book, 1995. 123, 130

[Grandvalet 2006] Y. Grandvalet, J. Mariéthoz et S. Bengio. *A probabilistic interpretation of SVMs with an application to unbalanced classification.* Advances in Neural Information Processing Systems 18, pages 467–474, 2006. 74

[Grandvalet 2009] Y. Grandvalet, A. Rakotomamonjy, J. Keshet et S. Canu. *Support Vector Machines With a Reject Option.* Advances in Neural Information Processing Systems, no. 21, pages 537–544, 2009. 4, 84, 88, 98, 100, 101, 102, 103, 104

[Herbei 2006] R. Herbei et M. H. Wegkamp. *Classification with reject option.* The Canadian Journal of Statistics, vol. 34, no. 4, pages 709–721, 2006. 4, 74, 82, 83, 84, 85, 88, 98, 99, 100, 101, 102, 103, 104

[http ://perso.univ rennes1.fr/PolyF04cours.pdf] http ://perso.univ rennes1.fr/PolyF04cours.pdf. *Décomposition QR.* 71

[http ://www.seas.ucla.edu/chol.pdf] http ://www.seas.ucla.edu/chol.pdf. *Cholesky factorization.* 71

[IEEE 1995] IEEE. *Recommended Practices for Monitoring Electric Power Quality.* IEEE Std, pages 1159–1995, 1995. 107

[Indyk 1998] R. Indyk P. et Motwani. *Approximate nearest neighbors.* Towards removing the curse of dimensionality, vol. 604-613, 1998. 30

[Ishibuchi 2000] T. Ishibuchi H. et Nakashima. *Multi-objective pattern and feature selection by a genetic algorithm.* In Proc. of Genetic and Evolutionary Computation Conference GECCO, vol. 1069-1076, 2000. 15

[Jain 1997] D. Jain A. et Zongker. *Feature selection : Evaluation, application, and small sample performance.* Pattern Recognition Letters, IEEE Transaction on Pattern Analysis and Machine Intelligence, no. 19, pages 153–158, 1997. 15

[Jensen 2001] F. V. Jensen. *Bayesian Networks and Decision Graphs.* Springer-Verlag New York,, 2001. 31

[Jolliffe 1986] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag., 1986. 4

[Kanitpanyacharoean 2004] W. Kanitpanyacharoean et S. Premrudeepreechacharn. *Power Quality Problem Classification Using Wavelet Transformation and Artificial Neural Networks.* IEEE Xplore, pages 1106–1113, 2004. 116

[Kaufmann 2000] Guido Kaufmann et Horst Bunke. *Automated reading of cheque amounts.* Pattern Anal. Appl., vol. 3, no. 2, page 132Ű141, 2000. 74

[Kittler 1978] J. Kittler. *Feature set search algorithms. Pattern Recognition and Signal Processing.* Springer-Verlag., no. 20, pages 41–60, 1978. 12

[Knerr 1997] V. Anisimov O. Baret N. Gorski D. Price Knerr S. et J. C. Simon. *The a2ia intercheque system : Courtesy amount and legal amount recognition for french checks.* International Journal of Pattern Recognition and Artificial Intelligence, vol. 11, no. 4, page 505Ű548, 1997. 74

[Kuncheva 1999] L. C. Kuncheva L. I. et Jain. *Nearest neighbor classifier :.* Simultaneous editing and feature selection, 1999. 15

[Kwok 1999] J.T. Kwok. *Moderating the outputs of support vector machine classifiers.* IEEE Trans. on Neural Networks, vol. 10, no. 5, pages 1018–1031, 1999. 74

[Lethelier 1995] Edouard Manuel Leroux Lethelier et Michel Gilloux. *An automatic reading system for handwritten numeral amounts on french checks, In ICDAR.* International Journal of Pattern Recognition and Artificial Intelligence, page 92Ű97, 1995. 74

[Lin 2006] C.H. C.H.Wang Lin. *Adaptive wavelet networks for power-quality detection and discrimination in a power system.* IEEE Trans. Power. Delivery, pages 1106–1113, 2006. 116

[Liu 2004] Cheng-Lin H. Sako Liu et H. Fujisaw. *Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 11, pages 1395Ű–1407, 2004. 74

[Loosli 2005] G. Loosli, S. Canu, S. Vishwanathan et M. Chattopadhay. *Boite à outils SVM simple et rapide.* RIA - Revue d'Intelligence Artificielle, vol. 19, no. 4, pages 741–767, 2005. 90, 94

[Loosli 2006] G. Loosli. *Méthodes à noyaux pour la détection de contexte.* Thèse de doctorat de l'Institut National des Sciences Appliquées de Rouen, 2006. 57, 62

[Marill 1963] D. M. . Marill T. et Green. *On the effectiveness of receptors in recognition systems.* IEEE transactions on Information Theory., vol. 9, pages 11–17, 1963. 12

[Markou 2003] Markos Markou et Sameer Singh. *Novelty detection : a review - part 2 : neural network based approaches.* Signal Processing, vol. 83, no. 12, pages 2499Ű–2521, 2003. 75, 77

[Mercer 1909] J. Mercer. *Functions of positive and negative type and their connection with the theory of integral equations.* Philos. Trans. Roy. Soc, 1909. 41

[Milgram 2004] Jonathan Mohamed Cheriet Milgram et Robert Sabourin. *Two-stage classification system combining model-based and discriminative approaches.* In 17th International Conference on Pattern Recognition (ICPRŠ04), pages 152Ű–155, 2004. 75

[Minoux 1983] M. Minoux. *Programation mathématique. Théorie et algorithmes.* Tome 1. Dunod., 1983. 65

[Narendra 1977] K. Narendra P. M. et Fukunaga. *A branch and bound algorithm for feature subset selection.* IEEE Trans. Comput., no. 26, pages 917–922, 1977. 14

[Neapolitan 2003] R. E. Neapolitan. *Learning Bayesian Networks.* Prentice-Hall, Upper Saddle River, NJ, USA, 2003. 31

[Platt 1998] J. C. Platt. *Sequential Minimal Optimization : A Fast Algorithm for Training Support Vector Machines.* Microsoft Research, 1998. 123, 124

[Platt 1999] J. C. Platt. *Fast training of support vector machines using sequential minimal optimization.* In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods Support Vector Learning,, pages 185–208, 1999. 56

[Pudil 1994] Novovicova J. et Kittler J. Pudil P. *Floating search methods in feature selection.* Pattern Recognition and Signal Processing, no. 15, pages 1119–1125, 1994. 13

[Quinlan 1986] J. R. Quinlan. *Induction of decision trees.* Machine Learning, no. 1, pages 81–106, 1986. 30

[Quinlan 1993] J. R. Quinlan. *programs for Machine Learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, no. 1, 1993. 30

[Risi 2008] Kondor Risi. *Vapnik-Chervonenkis theory.* 2008. 41

[Santoso 1996] S. E. J. Power Santoso et P. Hofmann. *Power Quality Assessment via Wavelet Transform Analysis.* IEEE Trans. Power. Delivery, pages 924–930, 1996. 116

[Scholkopf 1999] C. Scholkopf B. Burges et A. Smola. *Introduction to support vector learning.* editors, Advances in Kernel Methods Support Vector Learning, 1999. 19

[Scholkopf 2002] B. Scholkopf et A. Smola. *Learning with Kernels. Support Vector Machines,.* Regularization, Optimization, and Beyond MIT Press, pages 185–208, 2002. 39

[Shannon 1974] C. E. Shannon. *A Mathematical Theory of Communication (Part 1).* Bell System Technical, Papers in the Development of Information Theory, Piscataway : IEEE Press,, pages 5–18, 1974. 109

[Smola 1998] A. Smola. *Estimation of Dependences Based on Empirical Data.* pringer Verlag, Berlin, 1982, 1998. 15, 17

[Somol 2004] Pudil P. et Kittler J. Somol P. *Fast branch and bound algorithms for optimal feature selection.* IEEE Pattern Analysis and Machine Intelligence, no. 26, pages 900–912, 2004. 15

[Steinwart 2003] Ingo Steinwart. *On the optimal parameter choice for support vector machines.* Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 25, no. 10, pages 1274–1284, 2003. 60

[Stockwell 1996] R.P. Lowe Stockwell R.G. L. Mansinha. *Localization of complex spectrum : the S transform.* IEEE Trans.Signal.Process, vol. 4, no. 44, pages 998–1001, 1996. 108

[**Amirou** 2014a] A. **Amirou**, Z. Zidelmal, D. Ould-Abdeslam, M. Aiden et J. Merckle. *S-Transform and Shannon energy for electrical disturbances detection,.* the 40th Annual Conference of the IEEE Industrial Electronics Society IECON'2014, Dallas, TX-USA, 2014. 16, 117, 120

[**Amirou** 2014b] A. **Amirou**, Zahia Zidelmal, Djaffar Ould-Abdeslam, Mohamed Aiden et Jean Merckle. *Optimizing the Classification Cost using SVMs with a Double Hinge Loss.* An International Journal of computing and informatics Informatica ISSN 0350–5596, vol. 38, no. 2, pages 125–133, 2014. 16, 120

[Tortorella 2004] F. Tortorella. *Reducing the classification cost of support vector classifiers through an ROC-based reject rule.* Pattern Analysis & Applications, vol. 7, no. 2, pages 128–143, 2004. 74

[Valiant 1984] L Valiant. *A Theory of the Learnable.* Communications of the ACM, vol. 27, pages 1134–1142, 1984. 41

[Vapnik 1963] V.N. Vapnik et A. Lerner. *Pattern Recognition using Generalized Portrait Method.* Automation and Remote Control, 1963. 32, 41

[Vapnik 1982] V.N. Vapnik. *Estimation of Dependences Based on Empirical Data.* pringer Verlag, Berlin, 1982. 17, 18, 41, 43

[Vapnik 1998] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer Series in Statistics, 1998. 15, 16, 17, 18, 33, 42, 43, 46, 50, 51, 84

[Vishwanathan 2003] S.V. Vishwanathan, Alexander J. Smola et M. Narasimha Murty. *SimpleSVM.* Proceedings of the Twentieth International Conference on Machine Learning, 2003. 57, 90

[Wahba 1998] G. Wahba. *Support vector machines, reproducing kernel hilbert spaces, and randomized.* gacv, 1998. 15, 69

[Wegkamp 2010] M. Yuan Wegkamp M. *em Classification methods with reject option based on convex risk min-imization.* Journal of Machine Learning Research, page 111Ű130, 2010. 4, 74, 82, 83, 84, 85, 88, 98, 99, 100, 101, 102, 103, 104

[Whitney 1971] A. W. Whitney. *A direct method of nonparametric measurement selection.* IEEE Trans. Comput, no. 20, pages 1100–1103, 1971. 12

[Zhu 2003] J. Zhu. *Flexible statistical modeling.* Ph.D. Thesis Stanford Univers, pages 9–16, 2003. 69

[Zidelmal 2009] Z. Zidelmal, **Amirou, A.** et A. Belouchrani. *Using Support Vector Machine SVMs with reject option for heatbeat classification.* 2nd International Conference on Bio-inspired Systems and Signal Processing, no. 2, pages 204–210, 2009. 16, 120

[Zidelmal 2012] Z. Zidelmal, **Amirou, A.** et A. Belouchrani. *Heartbeat classification using Support Vector Machines (SVMs) with an embedded reject option.* International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), no. DOI : 10.1142/S0218001412500012, 2012. 16, 96, 120

[Zidelmal 2013] Z. Zidelmal, **Ahmed Amirou,** , Djaffar Ould-Abdeslam et Jean Merckle. *ECG beat classification using a cost sensitive classifier.* Elsevier,Computer Methods and Programs in Biomedicine (CMPB) ISSN : 0169-2607, vol. 111, no. 3, pages 570–577, 2013. 16, 120

[Zoutendijk. 1960] G. Zoutendijk. *Methods of Feasible Directions.* Elsevier, 1960. 58