Discrimination robuste par méthode à noyaux

Antoine Lachaud

THÈSE

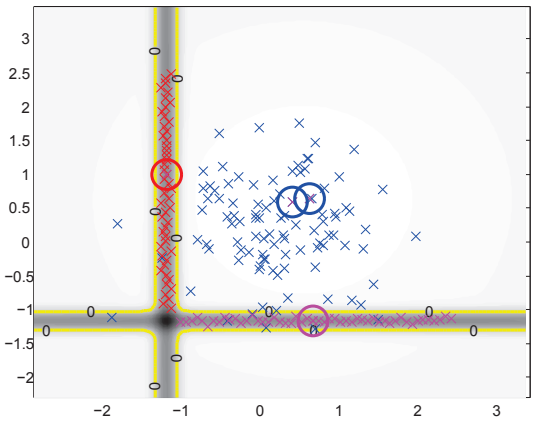
Présentée à :
L'Institut National des Sciences Appliquées de Rouen

En vue de l'obtention du titre de :
Docteur en Informatique

Par :
Antoine Lachaud

Intitulée :

Discrimination robuste par méthode à noyaux



soutenue le 17 décembre 2015 à l'INSA de Rouen

Devant le jury composé de :

Rapporteurs :	Massih-Reza AMINI	-	Université Joseph Fourier
	Frédéric PRÉCIOSO	-	Université de Nice
Examinateurs :	Antoine CORNUÉJOLS	-	AgroParisTech
	Laurent HEUTTE	-	Université de Rouen
Directeur de thèse :	Stéphane CANU	-	INSA de Rouen
Encadrant :	Frédéric SUARD	-	CEA
	David MERCIER	-	CEA

Résumé

L'objectif principal de cette thèse porte sur la réalisation simultanée de deux problématiques classiques en apprentissage machine à savoir la classification et l'interprétation. Ces dernières pouvant être potentiellement antinomiques, il est nécessaire d'introduire un compromis afin de pouvoir les réaliser conjointement. Plus précisément, nous recherchons une fonction de coût ainsi qu'une pénalisation appropriée afin de générer un modèle capable d'opérer une tâche de classification et qui intègre en son sein des informations explicatives dont l'analyse permet de fournir de l'interprétabilité sur les données de traitement.

Dans un premier temps tous resituons les problématiques de classification et d'interprétation, afin de déterminer un choix pertinent pour la fonction de coût et la régularisation. Nous avons opté respectivement pour la fonction charnière pour sa simplicité car elle n'implique pas de paramètre supplémentaire, et pour une pénalité *elastic-net* car elle permet de faire de la sélection de variables tout en corrigeant certains défauts inhérents au LASSO.

Cela nous a conduit à nous intéresser à un problème de classification appelé DRSVM, et plus particulièrement à un algorithme construisant un chemin de régularisation à partir d'un problème équivalent. Dans certaines configurations de données, nous avons constaté une certaine instabilité sur le chemin dont l'analyse des causes nous a révélé que la résolution du chemin de régularisation est inadaptée. Nous avons alors étudié le problème initial non-différentiable grâce à la théorie de la sous-différentielle afin de dériver les équations d'optimalité. Cette analyse nous a permis de comprendre quel paramètre choisir afin de construire le chemin de régularisation. Nous avons alors proposé un chemin alternatif par rapport à lambda1 le paramètre régularisation en norme 1 et montré qu'il est linéaire par morceaux.

Cependant le problème DRSVM est dans sa formulation initiale un problème linéaire, nous sommes limités à un unique choix dans la représentation des observations. Or, il est possible que le format des données d'origine ne prenne pas en compte la richesse de leur topologie et de leur structure. Dans une perspective d'interprétabilité, il nous a paru pertinent de générer un modèle qui englobe ces informations. Aussi, l'approche par noyaux nous a semblé particulièrement adaptée car elle introduit la notion de mesure de similarité et permet ainsi de changer l'espace de représentation. Plus précisément, l'étude des noyaux, nous a mené à nous intéresser à une forme de modèle appelé *kernel basis* qui permet, de par sa structure, de générer des modèles où l'influence des noyaux est pondérée localement. Dans un premier temps nous avons mené une analyse via la théorie des RKHS afin de déterminer la pénalisation appropriée pour ce modèle. La conclusion de cette étude, nous a montré que la pénalisation uniquement en norme 2 n'est pas pertinente pour le *kernel basis*.

Nous avons alors recherché comment intégrer le modèle *kernel basis* au sein du problème linéaire DRSVM. L'introduction de la kernelisation via le *kernel trick* n'est pas adaptée dans cette situation en raison de la présence du terme de régularisation en norme 1. Aussi, nous avons opté pour une approche par dictionnaire que nous avons construit de manière à générer la forme *kernel basis* au sein du DRSVM. Puis nous avons validé notre fusion du DRSVM à travers plusieurs ensembles de données jouets avec pour chaque simulation une double exigence : nous avons vérifié que notre modèle effectue la tâche de classification de façon satisfaisante et nous avons analysé la forme de solution pour montrer qu'elle génère au son sein des éléments interprétables permettant d'expliquer les observations. Enfin, nous avons choisi de traiter la base d'images MNIST afin de valider notre approche sur des données réelles. La visualisation des images, nous a permis d'une part, de jouer le rôle d'expert et d'introduire un a priori dans le choix des noyaux à utiliser, et d'autre part, d'analyser la forme de la solution et d'illustrer l'interprétabilité de notre modèle.

Remerciements

Je ne saurais en ces quelques lignes, citer de manière exhaustive, toutes les personnes qui ont contribué directement ou indirectement au succès de ma thèse.

Je voudrais tout d'abord remercier mes différents encadrants ainsi que mon directeur de thèse, pour leur aide précieuse tout au long de la thèse. David m'a appris à prendre du recul vis à vis des équations afin de déceler l'intuition physique sous-jacente aux concepts. Frédéric m'a beaucoup apporté au point de vue méthodologique. Enfin Stéphane m'a transmis son amour de la rigueur scientifique.

Massih-Reza AMINI et Frédéric PRÉCIOSO m'ont fait l'honneur d'être rapporteurs sur ma thèse et les remarques qu'ils ont formulées lors de ma soutenance m'ont permis d'envisager d'autres pistes de réflexion pour continuer mon travail de recherche. Je veux également remercier Antoine CORNUÉJOLS et Laurent HEUTTE d'avoir participé à mon jury.

Au cours de la thèse, j'ai intégré le laboratoire LADIS (Laboratoire d'Analyse de Données et Intelligence des Systèmes), j'ai apprécié la bonne ambiance au sein de l'équipe lors des différents échanges que j'ai pu avoir. En outre, je n'oublierai jamais les nombreuses et passionnantes discussions scientifiques (ou pas) dans le bureau des thésards avec Credo, Nicolas, Maxime et Néhémy.

Enfin je remercie ma famille qui m'a soutenu tout au long de la thèse et qui m' aidé lors de la relecture du manuscrit.

Table des matières

Notations **1**

Introduction **3**

1 Apprendre pour expliquer ou pour prédire ? **9**

1.1 Contexte : de la donnée à la connaissance 10

1.2 Datamining : synthétiser automatiquement les données 14

1.3 Apprendre un modèle 18

1.4 Régularisation des modèles d'apprentissage 29

1.5 Formulation retenue et orientation des travaux 33

2 Le DRSVM un modèle interprétable ? **35**

2.1 Le DRSVM, un problème de classification avec sélection de variables intelligente 36

2.2 DRSVM et chemin de régularisation 49

2.3 Proposition d'un chemin pour le DRSVM, via l'analyse de la sous-différentielle 58

2.4 Conclusion 78

3 Kernelisation DRSVM **81**

3.1 Machines à noyaux 82

3.2 Le modèle *kernel basis*, une approche multi-noyau 90

3.3 Formalisation du problème *kernel basis* via les RKHS 94

3.4 Kernelisation du DRSVM 101

3.5 Expérimentations pour le *kernel basis* 105

3.6 Conclusion 121

Conclusion **123**

Annexe **127**

A Analyse du SVM MCP via la sous-différentielle 127

Notations

Acronymes d'algorithmes et de méthodes

SVM ..... Support Vector Machine
L_1 ..... Norme 1
L_2 ..... Norme 2
CVX ..... Solveur d'optimisation convexe [1]
DRSVM ..... Doubly Regularized Support Vector Machine
KB_DRSVM ..... Kernel based Doubly Regularized Support Vector Machine

Ensembles

$\mathscr{E}$ ..... Points dans la marge
$\mathscr{R}$ ..... Points bien classés
$\mathscr{L}$ ..... Points non classés
$\mathscr{V}_\beta$ ..... Variables actives
$\mathscr{V}_0$ ..... Variables non actives
I^+ ..... Points ayant un label positif
I^- ..... Points ayant un label négatif
$\mathbb{R}$ ..... Valeurs réelles
H ..... Espace de Hilbert
$\mathscr{H}$ ..... Espace RKHS

Fonctions et opérateur

H ..... Fonction de coût charnière (*hinge loss*)
f ..... Fonction de décision
∂ ..... Sous-différentielle
v ..... Sous-gradient
L ..... Lagrangien

1. `http://cvxr.com/cvx/`

$arg\,max$ ..... Valeur maximale d'un ensemble
$\|.\|$ ..... Norme
$\langle .,. \rangle$ ..... Produit scalaire
$.^T$ ..... Transposée

Variables

n ..... Nombre d'exemples
p ..... Dimension des variables explicatives
$\mathscr{X}$ ..... Domaine des variables explicatives (entrées)
X ..... Variable aléatoire associée aux entrées
x ..... Réalisation de X (entrées)
$\mathbf{X}$ ..... Matrice $n \times p$ des entrées
$\mathscr{Y}$ ..... Ensemble des étiquettes possible (co domaine)
Y ..... Variable aléatoire associée aux étiquettes
y ..... Réalisation de Y. Etiquette ou classe associée à une donnée x
$\mathbf{Y}$ ..... Vecteur de n étiquettes
S_n ..... Ensemble d'apprentissage. Echantillon de n couples $(x_i, y_i)_{1 \leq i \leq n}$
β ..... Coefficient de pondération de la fonction de décision
β_0 ..... Biais
λ_1 ..... Coefficient associé à la régularisation L_1
λ_2 ..... Coefficient associé à la régularisation L_2

ε ..... Variables d'écart
r ..... Résidu
c ..... Valeur de corrélation
α ..... Coefficient de Lagrange associé à la contrainte sur la marge
η ..... Coefficient de Lagrange associé à la contrainte sur la norme 1
s ..... Paramètre de régularisation associé à la contrainte sur la norme 1

α ..... Sous-différentielle associée à la fonction *hinge*
γ ..... Sous-différentielle associé à la norme 1
S , S' ..... Systèmes d'équations
J ..... Fonction de coût DRSVM
δ ..... Variable intermédiaire utilisée pour le calcul de la sous-différentielle du DRSVM

Introduction

« L'éternel mystère du monde est son intelligibilité »

Albert Einstein

Contexte général, motivations

Afin de décrire la complexité du monde, la démarche scientifique consiste à en rechercher des approximations cohérentes avec la réalité appelées *modèles*. Un modèle peut être défini de différentes manières, soit par des connaissances, soit par des observations. Dans le premier cas, la formulation du modèle est établie par des a priori physiques, chimiques ou encore électriques très précis capables de représenter les phénomènes à plusieurs échelles depuis l'élément le plus petit jusqu'au système complet. Cette famille de modèles, appelée *boites blanches* est définie par sa capacité à caractériser au plus juste l'ensemble des phénomènes rencontrés, quelque soit l'état du système dans son ensemble. Cependant, une telle capacité implique de formuler dans le moindre détail les réponses possibles du système et de ses composants à un stimulus. Dans le deuxième cas, la définition du modèle repose sur des données d'observation et la capacité du modèle à copier ces observations pour représenter de manière fidèle les phénomènes mesurés. Ces modèles-ci sont appelés *boites noires*, car l'utilisation intrinsèque des observations n'est pas intelligible par l'homme, quelque soit son domaine d'expertise. L'intervention humaine est ici nécessaire pour choisir l'architecture du modèle qui permettra de poser le problème mathématiquement afin de mettre en correspondance les observations avec le modèle. De manière générale, un modèle peut donc se définir comme un ensemble d'hypothèses initiales qui sont validées et configurées pour être adaptées aux observations recueillies.

Quelque soit la nature du modèle construit, il est nécessaire de s'interroger sur sa pertinence. Au-delà de la précision apportée à la formulation du modèle, c'est essentiellement l'usage de ce

modèle qui permettra de le juger et de le valider. Une approche pour juger la qualité du modèle est d'évaluer son pouvoir de *prédiction* sur de nouvelles données d'observation entraînant sa validation ou sa réfutation. Dans une recherche d'*explication* des phénomènes physiques, il est souhaitable de construire le modèle de manière à ce qu'il fasse sens pour l'homme, afin qu'indépendamment des données observées, on puisse comprendre son fonctionnement intrinsèque.

La dualité entre l'explication et la prédiction est fondamentale en science car ces problématiques sont toutes deux nécessaires pour générer des connaissances fiables et comprendre les phénomènes physiques de notre environnement. Nous pouvons néanmoins formuler la question suivante : faut-il expliquer pour prédire ou prédire pour expliquer ? Cette vaste question est intrinsèquement liée à la notion de *connaissance* qui est en elle-même sujette à interprétation. Elle constitue, selon la théorie de la connaissance, une croyance vraie mais qui, de plus, est justifiée. En d'autres termes, il ne suffit pas de savoir, mais de savoir pourquoi l'on sait.

Aussi, il est tout à fait possible de prédire avec justesse l'évolution d'un phénomène sans en avoir une vraie connaissance ni de pouvoir l'expliquer. C'est notamment un des usages les plus fréquents des modèles *boites noires*, et, si la finalité d'utilisation se limite à l'exécution d'une tâche, ils sont pleinement satisfaisants. En revanche, si l'on cherche à utiliser le modèle également en vue d'expliquer les données d'observations, il est nécessaire de s'assurer qu'il soit interprétable, c'est à dire telle que sa forme intègre des informations intelligibles. Les modèles statistiques s'inscrivent naturellement dans cette définition. La formulation du modèle n'utilise que les données d'observations, l'homme devant au préalable définir l'architecture du modèle, c'est à dire dans ce cas, la manière d'intégrer et de combiner au sein du modèle les données. C'est une approche de modélisation très aboutie pour l'automatisation de tâches, en particulier pour la prédiction. Cependant, il est plus délicat de concevoir un modèle statistique dans un objectif d'interprétation. Le fonctionnement de ces modèles peut se résumer par "qui se ressemble s'assemble". Il en ressort que la qualité du modèle est donc fortement lié à la qualité de la métrique. La capacité d'interprétation doit donc être focalisée sur la définition d'une métrique propice à comparer l'information contenue dans les données, ainsi que la prise en compte de la métrique dans la formulation selon un schéma lisible. Ainsi dans le cadre de cette thèse, nous nous sommes intéressés à des possibilités d'intégration d'éléments explicatifs au sein du modèle de prédiction dans une perspective d'interprétation des données.

Organisation du manuscrit

Le manuscrit est décomposé en trois parties principales correspondant respectivement aux chapitres 1, 2 et 3. Le premier chapitre est un état de l'art sur l'analyse de données, le deuxième présente l'algorithme DRSVM par le biais des chemins de régularisation et le dernier introduit l'approche noyau du DRSVM.

L'analyse de données pour l'extraction de la connaissance

Nous avons avons commencé par nous interroger dans le chapitre 1 sur la dualité entre les problématiques d'explication et de prédiction. Ces problématiques ne sont pas nécessairement antinomiques et nous nous sommes intéressé à la possibilité de construction de modèles de prédiction intégrant au sein leur formulation des informations susceptibles d'être interprétées en vue de comprendre les données. Expliquer la nature d'un phénomène à partir de l'analyse des observations recueillies est un enjeu phare de l'analyse de données et de nombreuses approches on été proposées afin d'extraire de l'information pertinente. Ce chapitre présente ainsi la démarche d'analyse de données impliquant notamment des méthodes statistiques de fouille de données.

L'enjeu de l'apprentissage statistique est bien différent. Il consiste à inférer à partir de données collectées des modèles de prédictions capables de réaliser des tâches de manière autonome. Lorsque l'on cherche à construire une machine d'apprentissage dont la fonction est de répartir les données en différents groupes cohérents, il s'agit d'une tâche de classification. Dans le cadre de notre problématique de thèse, nous avons recherché plus particulièrement des manières de définir et d'apprendre des classifieurs interprétables. C'est à dire que nous avons recherché à générer des modèles dont la forme permet d'extraire des informations pouvant être exploitées afin d'expliquer les données. Cependant, cette capacité d'interprétation impose quelques contraintes sur la structure même du modèle appris. Cela nous a conduit à nous interroger sur le choix de la pénalisation du modèle afin d'induire de l'interprétabilité au sein de la forme de la solution.

Le modèle DRSVM est-il un modèle interprétable

Dans l'optique d'induire de l'interprétabilité au sein du classifieur, nous avons commencé par nous intéresser au processus de sélection de variables. Cette étape est essentielle car elle permet d'extraire les variables les plus discriminantes et de les intégrer au sein du modèle, facilitant ainsi l'analyse du modèle et donc son interprétation. Cela nous a conduit à étudier des pénalités de type LASSO, dont la singularité en zéro induit une certaine parcimonie et un processus de sélection de variables. Néanmoins la pénalité LASSO souffre d'un certain nombre de limitations. Notamment une incapacité à sélectionner plus de variables que de points, ce qui peut être problématique dans le cas où les données sont de hautes dimensions. De plus, cette pénalité possède un caractère propre à rejeter des variables, indépendamment de leur caractère discriminant, si elles sont fortement corrélées avec des variables déjà sélectionnées. Pour remédier à ces limitations, l'approche de régularisation *elastic-net* introduite initialement dans un contexte de régression a été étendue dans le cadre de la classification à travers un problème appelé DRSVM.

Dans le deuxième chapitre, l'étude de ce problème montre que c'est un problème paramétrique et que par conséquent il est possible de construire un chemin de régularisation (voir section 2.1), une technique particulièrement intéressante afin de suivre l'évolution du modèle et d'analyser les dynamiques de synergie entre les variables discriminantes. Un algorithme existant propose de résoudre un problème équivalent via la technique des chemins pour résoudre le DRSVM. Nous en avons réalisé une implémentation sous MATLAB et avons constaté l'apparition de problèmes récurrents liés à la nature du problème (voir section 2.2). L'analyse des causes des difficultés rencontrées nous a conduit à repartir de la formulation initiale du DRSVM afin de construire une formulation robuste du chemin de la régularisation en norme 1. Mais ce problème n'étant pas différentiable, nous avons utilisé la théorie de la sous-différentielle et proposé un chemin. Puis nous avons illustré la robustesse de notre approche par une validation théorique et expérimentale (voir section 2.3). Ensuite nous avons exploré des pistes afin de « kerneliser » le DRSVM.

Comment inclure des noyaux dans la formulation DRSVM

Le DRSVM est un problème qui inclut un processus de sélection de variables intelligent afin de construire un modèle interprétable. Cependant le DRSVM est dans sa formulation initiale un problème linéaire et nous sommes donc limités dans le choix de la représentation des données pendant l'apprentissage. Or, dans une optique d'interprétabilité, il peut être justement intéressant de sortir du cadre linéaire et d'essayer de construire le modèle d'apprentissage, non pas à partir des variables d'origines, mais à partir de variables davantage explicatives appelées *prototypes*. Plus précisément nous avons recherché des formes de prototypes de manière à retenir dans le modèle final des variables symbolisant des relations de dépendance locale entre les points d'observation et les variables. Cela nous a conduit à nous intéresser aux noyaux (voir section 3.1) en raison de leur capacité à changer l'espace de représentation des données et d'englober dans leur formulation des informations relatives à la topologie et à la structure des données.

Une forme de modèle de type kernel appelée *kernel basis* a particulièrement retenu notre attention car elle permet d'associer des noyaux différents, ce qui est en adéquation avec notre recherche de prototypes spécifiques (voir section 3.2). Nous avons mené une étude théorique via la théorie des RKHS dont les conclusions nous indiquent qu'il est pertinent de nous orienter vers une pénalisation de type $L_1 - L_2$ (voir section 3.3). Ensuite nous avons réfléchi sur la manière de combiner le DRSVM avec le modèle *kernel basis*. La présence du terme de régularisation en norme 1 rend cependant difficile l'introduction de la kernelisation via les RKHS (voir section 3.4). A la place, nous avons choisi d'adopter une approche dictionnaire afin de coupler le problème DRSVM au modèle *kernel basis* (voir section 3.4). Enfin nous avons réalisé une série d'expérimentations sur des données synthétiques et des données réelles afin de valider la capacité de notre modèle à réaliser une tâche de classification tout en générant simultanément de l'interprétabilité (voir section 3.5).

Contributions

Les apports principaux de cette thèse se composent d'une robustification d'un algorithme de chemin de régularisation appelé DRSVM et d'une proposition d'extension dans le cadre *kernel basis* dans une perspective d'interprétation de modèle.

Formulation du chemin λ_1 DRSVM

Nous avons rencontré des problèmes structurels liés à la nature du chemin proposé pour l'algorithme DRSVM. En outre nous avons conjecturé que le paramètre par rapport auquel est construit le chemin n'est pas approprié. Nous avons alors réalisé une étude du problème DRSVM sous sa forme initiale. Mais ce dernier n'étant pas différentiable, il a été nécessaire de mener l'analyse via le prisme de la théorie de la sous-différentielle. Cela nous a conduit à proposer la construction d'un chemin linéaire par morceau par rapport au paramètre de régularisation λ_1. Nous avons présenté cette approche lors de la conférence ESANN de 2014 [Lachaud et al., 2014].

Reformulation du problème DRSVM par les noyaux

Le DRSVM est, dans sa formulation initiale, un algorithme linéaire. Afin de l'adapter en modèle *kernel basis*, nous nous sommes interrogés sur la manière d'introduire les noyaux au sein de ce modèle. Le problème faisant intervenir dans sa structure un terme en norme 1, il est difficile d'induire la kernelisation de manière directe par le *kernel trick*. Aussi, nous avons opté pour une approche par dictionnaire afin d'introduire les noyaux au sein du DRSVM. Plus précisément nous avons explicité la forme du dictionnaire approprié afin d'obtenir une solution de la forme *kernel basis*. Une fois le dictionnaire construit nous avons résolu le problème DRSVM à l'aide de la résolution du chemin en norme 1 développée ci-dessus. Ensuite nous avons proposé pour une application de reconnaissance de formes dans des images un protocole afin d'illustrer l'aptitude de ce modèle à proposer des modèles pertinents et interprétables. Ce deuxième apport fait l'objet d'une publication en cours de rédaction d'un article journal portant sur la fusion du DRSVM et le *kernel basis*, dont la soumission est prévue début 2016.

1

Apprendre pour expliquer ou pour prédire ?

Sommaire

1.1 Contexte : de la donnée à la connaissance 10
1.1.1 Définition des informations et données 10
1.1.2 Les outils et méthodes statistiques 12
1.2 Datamining : synthétiser automatiquement les données 14
1.2.1 Statistique descriptive 14
1.2.2 Estimer les relations entre les variables 15
1.2.3 Localisation les données 17
1.2.4 Limites de l'analyse de données pour l'interprétation 18
1.3 Apprendre un modèle 18
1.3.1 Définition de l'apprentissage 19
1.3.2 Apprentissage supervisé 20
1.3.3 Compréhension des modèles construits 24
1.4 Régularisation des modèles d'apprentissage 29
1.4.1 Phénomène de sur-apprentissage et de sous-apprentissage 30
1.4.2 Les problèmes régularisés 31
1.4.3 Choix de la régularisation et interprétabilité 31
1.5 Formulation retenue et orientation des travaux 33

1.1 Contexte : de la donnée à la connaissance

Depuis quelques années, la forte croissance du monde numérique a été poussée par un domaine particulier : les données. En effet, la multiplication des sources de données telles les mesures physiques à partir des capteurs, les images vidéos, les données audio, les informations utilisateurs ou bien d'autres deviennent de plus en plus présentes.

Ces données sont non seulement générées et transférées, mais elles sont la plupart du temps stockées, afin d'être exploitées ultérieurement. Cette exploitation peut être automatisée au sein d'un procédé ou d'un traitement, ou bien formatée pour une analyse par un expert de l'application considérée. Dans ce dernier cas, il est important non seulement d'extraire une synthèse qui soit la plus fiable possible au regard de critères statistiques, mais qui doit conserver une lisibilité propice à l'interprétation humaine.

1.1.1 Définition des informations et données

La notion de données, selon le domaine ou l'interlocuteur peut prendre différentes significations. Dans un premier temps, il est donc nécessaire de définir le concept de données et la notion d'information qui peut en être extraite. De manière générale, les données et informations sont des valeurs numériques caractéristiques de l'état d'un système, d'un composant, d'une personne, etc. La distinction entre l'information et la donnée réside dans le niveau d'abstraction définissant cette valeur, l'information étant généralement une valeur d'un niveau plus élevé, contenu partiellement ou intégralement dans l'ensemble des données dont elle est issue. La finalité revient donc à extraire des informations qui soient lisibles et compréhensibles à partir d'un ensemble de données.

1.1.1.1 Sources et formats de données

Différentes applications telles que le diagnostic médical, la surveillance de systèmes industriels, la prospection géophysique, le calcul de risque financier ou encore la prévision météorologique sont fondées généralement sur l'expérience acquise au fil des ans à partir de l'historique des comportements des différents acteurs évoqués. Afin de mieux comprendre et anticiper ces comportements, il est nécessaire non seulement de collecter des mesures ou des informations mais également de pouvoir les exploiter par la suite. Ainsi, la première démarche consiste à instrumenter les systèmes pour mesurer de manière continue les phénomènes observés et collecter de manière systématique ces mesures au sein d'un système numérique. Sans cette mémorisation et cette procédure de mesure systématique il est difficilement envisageable d'analyser et de comparer des exemples nombreux relatifs aux applications envisagées.

Parmi les différents champs de données et mesures disponibles, les données générées seront des images, du texte, des séries temporelles, des signaux, ou encore des valeurs binaires. La figure 1.1 affiche quelques exemples de données selon ces différents formats. Dans la plupart des cas, les données peuvent être définies à l'aide de tableaux de valeurs qui sont bien adaptés à des traitements numériques. Ainsi, dans la suite, les travaux seront focalisés principalement sur des données matricielles ou vectorielles.

(a) (b) (c)

FIGURE 1.1 : Exemples de données : signal temporel (a), image (b) et impulsions (c).

Une donnée individuelle, ou observation, sera intitulée x. Il s'agit d'un vecteur réel de dimension p, soit $x \in \mathbb{R}^p$. Un ensemble de n données sera une matrice $\mathbf{X}$ de $\mathbb{R}^{n \times p}$.

Il faut noter qu'il s'agit ici de données brutes. Selon l'application envisagée, il est parfois souhaitable d'ajouter également une étape intermédiaire entre l'acquisition et l'analyse afin de caractériser les données. L'objectif est ici de trouver un nouveau codage, c'est à dire une nouvelle représentation des données. L'intérêt de ce traitement peut être la réduction du bruit, la synchronisation des séries temporelles, la limitation du volume. Selon la méthode appliquée, la nouvelle base considérée contiendra des données vectorielles de même dimension.

Lorsque seulement les données x sont disponibles, les méthodes d'analyse sont dites *non supervisées* car aucune description adjacente n'est associée à celles-ci. Dans le cas contraire, si une étiquette y est associée à chaque individu x, alors les méthodes sont *supervisées*. Ici l'étiquette est un scalaire, c'est à dire une valeur réelle et l'ensemble de n étiquettes est noté $\mathbf{Y} \in \mathbb{R}^n$.

1.1.1.2 Des données à l'information

Individuellement, chaque observation est définie par un ensemble de valeurs. Leur signification peut avoir une réalité physique et donc porteuse de sens en elle-même. La valeur ajoutée de l'analyse d'un ensemble de données revient dans la capacité à extraire de l'information, c'est à dire des conclusions permettant de comprendre l'organisation et les phénomènes de génération de ces données. Une information est extraite de l'intégralité ou d'une partie des données, elle reste donc en cohérence avec la nature des observations. C'est donc une donnée de plus haut niveau, qui intègre une synthèse de la base étudiée.

L'information peut permettre d'identifier des synergies entre des variables, la redondance de valeurs, des individus représentatifs ou au contraire des individus aberrants et plus généralement une relation sous-jacente entre des individus et des variables descriptives.

La notion de connaissance s'applique lorsque ces informations sont analysées par un expert du domaine d'application considéré qui en extrait une information intelligible. La connaissance n'est pas nécessairement nouvelle, elle peut au contraire être conforme à des hypothèses connues mais non appliquées encore sur les données observées. Le processus décrit par Fayyad et al. [1996] résume les étapes préliminaires de l'extraction de connaissance à partir de données. Il met notamment l'accent sur le fait que c'est un processus visant à impliquer un humain dans une chaîne de traitement. A ce titre, les informations produites doivent pouvoir être visualisées.

1.1.2 Les outils et méthodes statistiques

Le cœur de l'extraction de connaissance réside dans l'application de méthodes pour automatiser au maximum le traitement de volumes de données afin de faciliter l'interprétation par un analyste ou un expert.

Au fil des ans, les techniques d'analyse ont été appliquées sur de nombreuses catégories de données. La généricité de la méthodologie d'analyse a en effet été éprouvée pour différents usages à partir de données vectorielles, temporelles, matricielles, Les domaines d'application de ces méthodes d'analyse de données sont très variés.

On peut notamment citer quelques exemples en :
- Santé : quantification de risques selon des symptômes et des données cliniques,
- Industrie : détection d'anomalies dans un système,
- Finances : compréhension du marché et prédiction de l'évolution des prix,
- Marketing : analyse du comportement des acheteurs.

Ces méthodes, appelées aujourd'hui *analytics*, sont à l'interface des statistiques, de l'informatique et de la visualisation. Comme l'illustre la figure 1.2, il s'agit d'un schéma exhaustif du traitement itératif de l'information [Padhy et al., 2012].

Initialement focalisée sur la description synthétique des données pour la génération de tableaux de bord, les résultats nécessitaient une forte implication de l'expert qui devait maitriser l'intégralité des outils statistiques pour extraire efficacement des résultats pertinents et cohérents.

Par la suite, des composantes d'aide à l'analyse ont permis d'approfondir ces résultats, en particulier la visualisation et la recherche interactive. Une nouvelle dimension permettait ainsi à l'expert de comprendre non seulement la tendance des données, mais également de pouvoir accorder de l'importance à des cas spécifiques. Ces deux premières étapes se focalisent essen-

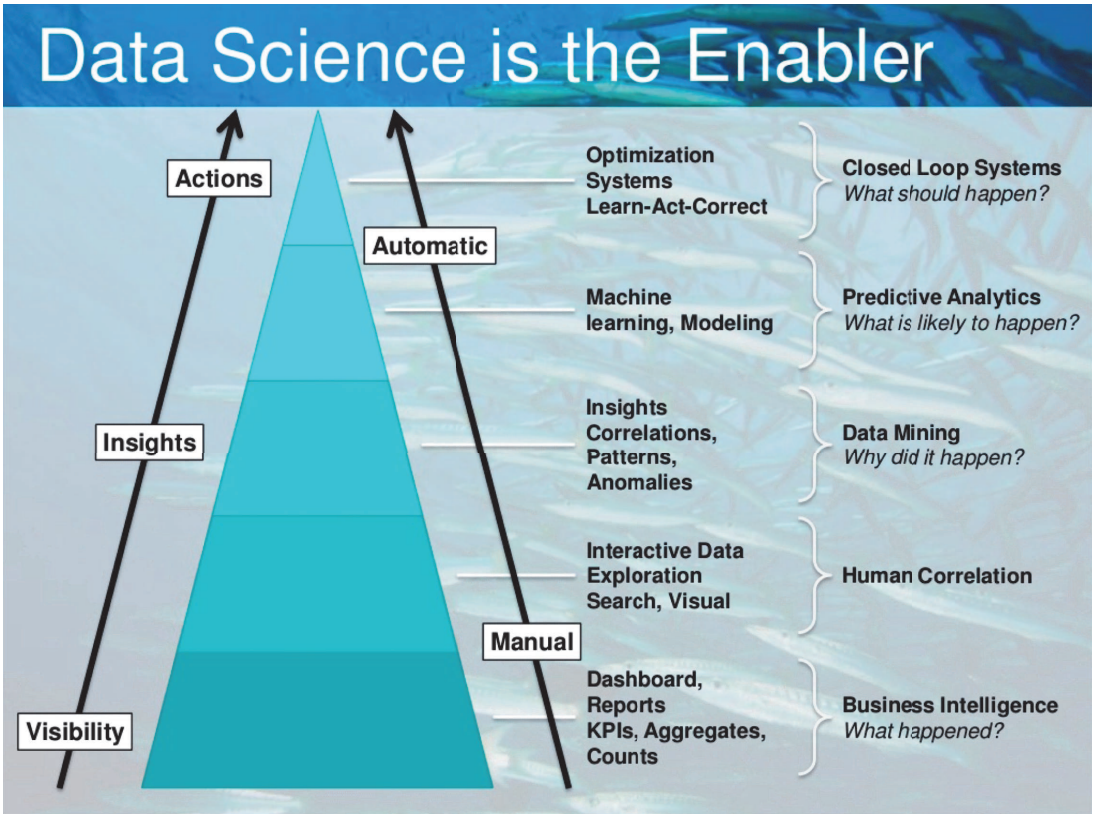


FIGURE 1.2 : Schéma d'exploitation depuis les données brutes jusqu'à la décision (Source : http ://hivedata.com)

tiellement sur l'historique d'observations et visent donc à extraire de cet ensemble de données des conclusions relatives au comportement passé, afin de pouvoir décrire et voir les tendances obtenues.

La fouille de données, ou *Datamining* peut être vue comme une étape complémentaire pour comprendre, c'est à dire d'établir des liens entre les données afin de justifier les tendances et les comportements décrits précédemment [Wu et al., 2008]. Les méthodes de corrélation ou de recherche de motifs sont ainsi capables de fournir des explications par la recherche de liens, mais également la quantification de l'impact d'un sous-ensemble de données.

Comme les tâches précédentes, une des difficultés réside dans la capacité à appliquer de telles méthodes sur des volumes conséquents, notamment par la combinatoire causée par la recherche exhaustive de relations. L'autre problématique relative à la fouille de données est la validité des analyses effectuées. La cohérence des résultats peut être partiellement validée par des connaissances expertes, mais doit pouvoir aussi proposer des résultats inconnus afin de générer de la connaissance. La fouille de données est un moyen de valider des hypothèses relatives à l'existence d'un modèle sous-jacent de génération de données. Elle ouvre la voie à l'apprentissage statistique.

Les étapes ultimes des *analytics* sont en effet l'aide à la décision. Une dimension de prédiction est ainsi nécessaire pour prolonger la validité des analyses fournies en appliquant une stratégie d'apprentissage applicable sur des données inconnues. La plupart des modèles d'apprentissage s'appuient sur des concepts statistiques existant en fouille de données en posant un cadre d'optimisation et des méthodes de résolution en vue de généraliser facilement la procédure de génération d'un modèle prédictif. Un tel modèle, selon son architecture, peut être appliqué pour estimer les valeurs de nouvelles données, ou bien être intégré dans un schéma d'optimisation de comportement afin de fournir des préconisations sur un usage plus efficace pour un scénario donné.

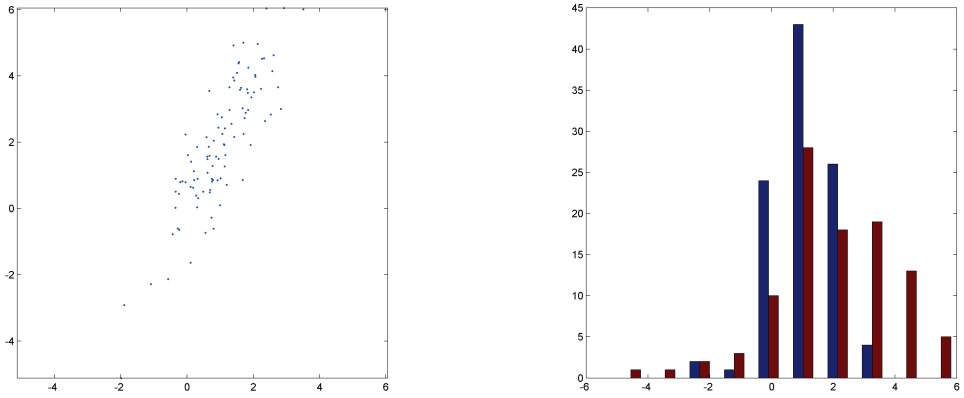
1.2 Datamining : synthétiser automatiquement les données

Un premier moyen pour extraire de la connaissance à partir de données consiste à les explorer. Cela constitue ce que l'on appelle fouille de donnée ou *Datamining*. L'objectif consiste à appliquer des méthodes statistiques pour non seulement décrire les valeurs mesurées, mais également identifier les variables d'intérêt contenant le cœur de l'information portée par l'ensemble de la base analysée.

1.2.1 Statistique descriptive

La statistique descriptive consiste à résumer les données d'observations, à partir d'informations de nature statistique [Weibull, 1951]. L'objectif est d'obtenir des informations qui constituent une synthèse efficace, permettant de comprendre et d'analyser le comportement des données. Pour avoir un aperçu de ce comportement, on peut commencer par calculer et afficher sous la forme d'un tableau de synthèse, des caractéristiques élémentaires telles que la moyenne, les valeurs extrémales, la médiane, l'écart type (voir figure 1.3). Notons qu'à certains égards, la statistique descriptive est proche de la représentation graphique de données statistiques ou *Statistical graphics* qui possèdent en commun certaines techniques. Notamment la méthode des histogrammes, l'un des moyens les plus classiques de représentation graphique de la répartition des données selon une variable [Silverman, 1986, Sheather, 2004].

Ainsi, la statistique descriptive constitue une étape préliminaire essentielle afin d'évaluer la localisation et la dispersion des données, d'identifier éventuellement des variables élémentaires permettant de caractériser l'évolution des données et de réaliser certains prétraitements tels que la normalisation ou le centrage. Mais pour des données à hautes dimensions faisant intervenir des mécanismes complexes entre les variables, la statistique descriptive ne suffit plus et il est alors nécessaire d'utiliser d'autres méthodes, capables d'exploiter et de faire émerger les relations pertinentes sur les données et sur les variables.



variables	μ	m	min	max	σ
x	1.0058	0.9741	-1.9852	3.5093	0.9958
y	1.9329	1.8892	-5.1184	6.0550	1.9763

FIGURE 1.3 : Nous avons affiché un jeu de données composé de deux variables (x,y) (figure de gauche). Le tableau affiche des informations élémentaires sur la répartition des données : la moyenne μ, la médiane m, les valeurs extrémales min/max et l'écart-type σ. La figure de droite représente la superposition des histogrammes de chaque variable (en bleu l'histogramme associé à la variable x, en rouge l'histogramme associé à la variable y).

1.2.2 Estimer les relations entre les variables

Une autre manière de décrire les données, consiste à étudier les relations, possiblement complexes, entre les différentes variables. Cette problématique constitue un enjeu phare du domaine de l'analyse de données et a conduit au développement de nombreuses méthodes afin d'extraire de telles relations.

1.2.2.1 Détecter l'existence d'une relation de dépendance entre les variables

On peut commencer par tester naïvement tout les couples de variables et observer s'il existe ou non une relation de dépendance. Pour ce faire on supposera que les observations sont des réalisations i.i.d. d'un couple de variables aléatoires (X,Y) de loi jointe de densité $\mathbb{P}(x,y)$ inconnue. L'information mutuelle est une mesure basée sur le concept d'entropie [Ihara, 1993] qui permet de détecter des relations de dépendance :

$$I(X,Y) = \int_{\mathbb{R}} \int_{\mathbb{R}} \mathbb{P}(x,y) \log \Big(\frac{\mathbb{P}(x,y)}{\mathbb{P}(x)\mathbb{P}(y)} \Big) dxdy,$$

avec $\mathbb{P}(x)$ et $\mathbb{P}(y)$ les densités des lois de X et Y.

Notons qu'elle n'indique aucune précision sur la forme de la dépendance. L'information mutuelle a l'avantage de détecter simplement l'existence des dépendances entre les variables [Batina et al., 2011].

1.2.2.2 Estimation de la forme d'une dépendance

Corrélation : dans certaines situations nous pouvons avoir des a priori sur la nature des relations entre les variables. Aussi, il peut être intéressant de définir des critères d'évaluation spécifiques à la forme des dépendances. Le coefficient de corrélation permet par exemple de quantifier le degré de dépendance linéaire entre deux variables [Joseph Lee Rodgers, 1988] :

$$\rho(X,Y) = \frac{E\big((X-E(X))(Y-E(Y))\big)}{\sqrt{E((X-E(X))^2)}\sqrt{E((Y-E(Y))^2)}}.$$

Contrairement à l'information mutuelle, il ne permet pas a priori de repérer d'autres formes de dépendance. Une haute valeur en valeur absolue du coefficient de corrélation indique une forte dépendance linéaire et son signe indique si les variables sont corrélées ou anti-corrélées.

Causalité : si le coefficient de corrélation est simple à calculer, sa réalité physique est plus difficile à interpréter. Une corrélation élevée entre deux variables ne permet pas nécessairement d'expliquer une variable en fonction de l'autre. Il est d'ailleurs possible de générer de nombreux paradoxes en partant du postulat inverse. Afin de qualifier l'explication d'une variable par rapport à une autre, on parle plutôt de causalité. Cette notion repose sur l'hypothèse implicite que l'une des variables est antérieure à la seconde (la cause précède l'effet) et qu'elle est cause unique de l'effet. Ce paradigme a conduit à la définition du test de Granger [Granger, 1969] qui permet de quantifier le degré de causalité entre deux variables. Notons que la notion de causalité est particulièrement pertinente dans le cadre des série temporelles où la notion d'antériorité est intrinsèquement présente [Eichler, 2012]. Néanmoins la notion de causalité, de part son aspect sémantique étendu demeure une notion ambiguë et de plus l'approche du test de Granger est restreinte au cas mono-causale [Guyon et al., 2007].

L'approche bi-variée à partir de mesures telles que l'information mutuelle, le coefficient de corrélation ou de causalité permet d'identifier simplement d'éventuelles interactions entre les variables des données d'étude. Mais elle peut devenir assez vite laborieuse quand le nombre de variables p devient important car on a $p(p-1)$ relations à analyser. Aussi, il peut être intéressant d'extraire plutôt des relations globales entre les variables afin d'expliquer les données.

1.2.2.3 Extraction d'un sous-ensemble de variables explicatives

Représentation des données : pour les problèmes de hautes dimensions, l'extraction de variables pertinentes en vue de leur description, dans l'espace des données d'origine est une tâche ardue. Aussi, il peut être intéressant d'essayer de rechercher par le biais de transformations, d'autres espaces où la représentation des données est plus adaptée. La célèbre méthode d'analyse en composantes principales (*Principal Component Analysis*) [Jolliffe, 2005] propose de décomposer les données dans des sous espaces orthogonaux ordonnés de manière à ce que les premiers espaces représentent au mieux la variabilité des données.

Sélection de variables : de manière générale toutes les variables d'un problème ne sont a priori pas explicatives. Aussi il peut être intéressant de réduire leur nombre afin de simplifier l'analyse des données en réalisant ce que l'on appelle *une sélection de variable* [Avrim Blum, 1997, Guyon and Elisseeff, 2003]. Par exemple, si on utilise l'analyse en composantes principales pour décrire les données, on peut effectuer une sélection de variables pertinentes en limitant la projection des données sur les premiers sous-espaces qui maximisent l'information. On obtient alors un ensemble limité de variables discriminantes permettant de décrire simplement les données. Notons que dans sa forme initiale, le problème de sélection de variable est combinatoire car il consiste à rechercher des sous-ensembles de variables informatifs et c'est donc a priori un problème extrêmement complexe.

1.2.3 Localisation les données

Plutôt que d'expliquer les observations à partir de variables explicatives, une approche duale consiste à rechercher des points afin de décrire les observations. Afin de se représenter comment sont localisées les données, on peut utiliser aux méthodes de *clustering* dont objectif consiste en la recherche de partitions des données en groupes appelés *classes*, où les observations ont un comportement homogène. Cette problématique est difficile, dans la mesure où nous ne savons pas a priori si les données d'étude ont effectivement une tendance à s'agréger en groupes et à supposer même qu'un tel phénomène existe, il reste à déterminer leur nombre.

1.2.3.1 Déterminer les points centraux

La description des données d'observation dans l'espace, peut être envisagée en tentant de les comparer relativement à des points, possiblement virtuels, qui représentent un caractère local des données. Si nous avons la connaissance d'une partition pertinente des données, un choix naturel est de s'intéresser au centre de gravité de chaque classe pour localiser les observations. Mais dans la pratique les classes sont a priori inconnues et il est nécessaire de se donner un

critère afin de pouvoir évaluer la pertinence d'une partition. Un choix standard est de prendre la somme des variances intra-classe $V(S)$ (voir par exemple Cornuéjols and Miclet [2011]) d'une partition $S = (s_j)_{1 \leq j \leq m}$ de $\mathbf{X}$:

$$V(S) = \frac{1}{n} \sum_{j=1}^{m} \sum_{x \in s_j} \left\| x - \mu_j \right\|_2^2, \tag{1.1}$$

avec m le nombre de classe, $(\mu_j)_{1 \leq j \leq m}$ les centres de gravité des classes et $|s_j|$ le cardinal de s_j.

Dans l'hypothèse où le nombre de classes est fixe, ce critère est particulièrement pertinent car plus le regroupement est adapté aux données et plus la variance intra-classe est faible. La méthode *k-means* introduite par MacQueen et al. [1967], consiste pour un jeu de données $(x_i)_{1 \leq i \leq n}$, à trouver une partition $S = (s_j)_{1 \leq j \leq m}$ de $\mathbf{X}$ minimisant la variance intra-classe (1.1).

Notons que la méthode est sous-optimale et il est possible de tomber sur un optimum local.

1.2.4 Limites de l'analyse de données pour l'interprétation

La fouille de données exploratoire permet d'extraire des informations pertinentes sur les données et constitue une approche essentielle dans le but d'expliquer les observations. Cependant elle possède plusieurs limites inhérentes à la tâche qu'elle tente de résoudre. Par hypothèse, ne sachant pas a priori ce que l'on cherche, il devient rapidement impossible quand la dimension du problème augmente, d'explorer toutes les relations possibles (combinaisons de variables ou de points explicatifs) et il est alors nécessaire d'avoir recours à des heuristiques afin d'axer la recherche pour extraire de l'information pertinente. Nous pouvons nous demander, dans l'hypothèse où de telles relations existent, s'il est possible lors d'un processus d'apprentissage de les faire ressortir à travers la forme du modèle afin de pouvoir expliquer les données d'observation.

1.3 Apprendre un modèle

A partir des observations retenues durant l'analyse préliminaire des données qui permet de faire émerger des schémas de dépendances au sein des données, la problématique de prédiction surgit naturellement. Elle consiste en effet à s'interroger sur la capacité à générer une fonction de prédiction, c'est à dire à généraliser les observations effectuées. Le modèle construit dans cette optique n'étant fondé que sur des données, il s'agit de modèles dits statistiques.

1.3.1 Définition de l'apprentissage

Le concept d'apprentissage machine ou *Machine Learning* a émergé historiquement avec le développement de l'informatique dans les année 50. La machine (automate, robot, ordinateur) peut être assimilée à un système abstrait dont la fonction est de réaliser une tâche. En général, elle reçoit des données dites d'*entrée* afin d'être fonctionnelle et renvoie des données dites de *sortie*, utilisées en vue de la réalisation de la tâche considérée. Arthur Samuel donne en 1959 la définition suivante du Machine Learning : "*Machine Learning : Field of study that gives computers the ability to learn without being explicitly programmed*". Bien que le terme *apprentissage*, ne soit pas explicitement défini, cette définition suggère la motivation de rendre les machines autonomes et capables d'évoluer par elles même, sans l'intervention directe de l'homme.

Une définition plus récente de Cros et Gardin donne un sens plus précis à l'apprentissage : "*Processus permettant à la machine d'améliorer ses performances, pour l'analyse et/ou pour la recherche automatique des informations en fonction de ses expériences propres*". La machine est appréhendée comme un objet sensible, qui, de par sa confrontation avec le réel, développe une sorte d'individualité et une forme de connaissance lui permettant d'effectuer plus efficacement une tâche donnée. Afin de juger du pouvoir de notre machine, il est nécessaire d'évaluer la qualité de l'apprentissage. La définition proposé par Tom Mitchell introduit la notion de mesure de performance afin d'évaluer l'apprentissage : "*A computer program CP is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.* Ainsi on compare l'état de la machine avant et après apprentissage, en évaluant sa capacité à exécuter une tâche, relativement à une mesure de performance que l'on a définie au préalable.

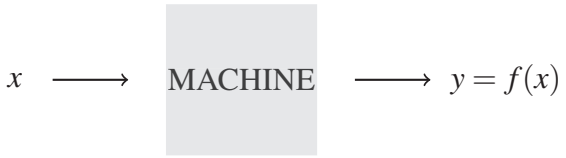


FIGURE 1.4 : Représentation d'une machine comme une fonction.

Avant de réaliser la phase d'apprentissage, il faut définir la structure de la machine en choisissant un *modèle d'apprentissage*. Dans le contexte des travaux présentés dans ce manuscrit, un modèle est défini à partir des données fournies en exemple du phénomène à apprendre sans a priori sur la forme de la fonction. Ainsi pour associer efficacement les données, les modèles d'apprentissage font intervenir dans leur formulation des coefficients, dont les valeurs nécessitent d'être réglées afin de rendre la machine opérationnelle. L'adaptation des coefficients du modèle de la machine est effectuée relativement à des données observées dites *données d'apprentissage* et constitue la phase d'apprentissage. Autrement dit, la machine, indépendamment

de son architecture, n'est pas directement utilisable, il faut préalablement l'adapter aux données et c'est en ce sens que nous parlons d'apprentissage machine. Le réglage des coefficients se fait par le biais d'une résolution par optimisation sur les données d'apprentissage. Le but en fin d'apprentissage est d'avoir, si possible, une machine capable de réaliser efficacement la tâche sur de nouvelles données de même nature que les données d'apprentissage. De façon très informelle, nous pouvons considérer que *réaliser une tâche T*, revient à prendre une action A en réponse à un stimulus S. Nous n'abordons pas ici la problématique d'apprentissage de préférence (*ranking*) ou de classifieurs multi-labels. Aussi, nous supposons que l'action est représentée par une variable aléatoire Y à valeur dans $\mathscr{Y} \subseteq \mathbb{R}$ et le stimulus par une variable aléatoire X à valeur dans $\mathscr{X} \subseteq \mathbb{R}^p$. De plus, nous assimilons la machine d'apprentissage à une fonction $f : \mathscr{X} \to \mathscr{Y}$ (voir figure 1.4).

1.3.2 Apprentissage supervisé

Dans la suite, nous nous intéresserons spécifiquement à l'apprentissage dit supervisé, c'est à dire que pour chaque donnée x une étiquette (*label*) y est associée.

1.3.2.1 Principe

Objectif : l'apprentissage dit *supervisé* consiste à exploiter l'information issue de couples d'observations (x,y) afin de construire une machine f capable de prédire pour un nouvel exemple x' via sa réponse $f(x')$ une valeur appropriée de y', c'est-à-dire tel que $f(x')$ et y soient proches, dans un certain sens qu'il est nécessaire de préciser.

Modélisation des observations : nous supposons que nous avons à disposition un ensemble de données d'observation $S_n = (x_i, y_i)_{1 \leq i \leq n} \subset (\mathscr{X} \times \mathscr{Y})^n$, réalisations d'un couple de variables aléatoires (X,Y) de loi parente de densité $\mathbb{P}_{X,Y} \in \mathscr{P}$, où $\mathscr{P}$ est l'ensemble des densités de probabilité sur $\mathscr{X} \times \mathscr{Y}$. De plus, l'échantillon est supposé vérifier la propriété d'indépendance (i.i.d.). Cette hypothèse est fondamentale en apprentissage statistique, dans la mesure où elle garantit que les observations récoltées sont bien issues d'une même source et que chaque exemple est informatif [Amini, 2015]. Sous ces hypothèses, l'apprentissage supervisé peut être aussi interprété comme un processus d'induction, où on cherche à estimer une fonction f à partir d'ensembles d'observations [Cornuéjols and Miclet, 2011].

Classification et régression : l'apprentissage peut se scinder en deux catégories : la régression et la classification. La régression consiste à partir d'un échantillon d'estimer une fonction de la densité de la loi des sorties Y par rapport à la loi des entrées X. La classification, bien qu'elle puisse se concevoir comme un cas particulier de régression où la variable prédite prend des valeurs discrètes, consiste aussi à séparer les données en différents groupes appelés *classes*.

Paradigme de l'apprentissage statistique : la modélisation du stimulus x en tant que réalisation dune variable aléatoire X de loi de densité $\mathbb{P}_X$ permet de rendre compte l'incertitude liée à la représentativité des données d'apprentissage dont on dispose. Plus la dimension des données d'entrée est grande, plus il est nécessaire d'avoir à disposition un nombre de données important pour pouvoir apprendre une représentation correspond à la réalité. Ce phénomène est connu sous le nom de « malédiction de la dimensionnalité » [Friedman, 1997]. La modélisation de l'action y en tant que réalisation d'une variable aléatoire suivant une loi marginale de densité conditionnelle sachant $X = x$ est intéressante dans la mesure où elle traduit l'incertitude sur la possibilité d'évaluer y connaissant x. Par exemple, si les variables (X,Y) sont indépendantes, construire une machine pour évaluer y à l'aide de x, est intrinsèquement voué à l'échec. Le paradigme de l'apprentissage statistique est fondamentalement différent des modèles paramétriques dans la mesure où aucune hypothèse sur la forme de la fonction n'est posée. Ces derniers font l'hypothèse que, connaissant la variable x, la valeur de y est déterminée par une fonction $f^\star$ appartenant à un ensemble connu et fini de fonctions.

Réaliser une tâche d'apprentissage consiste à construire $f^\star$ à partir d'un échantillon S_n de façon à ce qu'ayant observé $x \in \mathscr{X}$, la machine renvoie une réponse $y \in \mathscr{Y}$ qui permette d'effectuer la tâche le plus efficacement possible.

1.3.2.2 Un critère de performance : le risque

Fonction de coût : construire une machine consiste à trouver une fonction $f : \mathscr{X} \to \mathscr{Y}$ qui exploite au maximum la relation entre les variables $x \in \mathscr{X}$ et $y \in \mathscr{Y}$. Afin d'apprécier la qualité de la machine, il est nécessaire d'introduire $L : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}^+$ appelée fonction de coût qui quantifie l'erreur de prédiction. C'est à dire que pour une observation (x,y), plus la valeur prédite $f(x)$ est différente de y, plus la valeur de L est grande, afin de pénaliser l'erreur. Il existe plusieurs choix standards pour la fonction de coût (voir figure 1.5) qui pénalisent de manière différente les écarts entre prédictions et valeurs observées selon la tâche considérée. Ces fonctions coût les plus usuelles sont :

$$\begin{array}{ll} L_{0/1} & = \mathbb{1}_{y \neq f(x)} \\ L_{hinge} & = max[0, 1 - yf(x)] \\ L_{hinge^2} & = max[0, (1 - yf(x))^2] \\ L_{logistic} & = \frac{1}{log(2)} log\big(1 + exp^{-yf(x)}\big) \\ L_{exponential} & = exp^{-yf(x)} \\ L_{sigmoid} & = \frac{1}{2}(1 - tan(yf(x)). \end{array} \tag{1.2}$$

Dans le cadre de la classification, la fonction de coût la plus naturelle est la fonction binaire $L_{0/1}$ qui associe zéro si la prédiction est égale à l'étiquette et 1 sinon. Cependant, elle n'est ni continue ni convexe et par conséquent peu compatible avec une résolution efficace. Aussi,

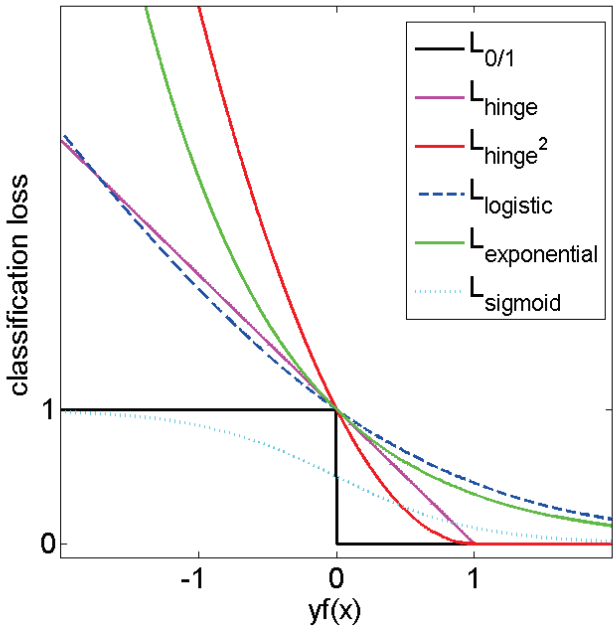


FIGURE 1.5 : Nous avons affiché différentes fonctions de coût utilisées dans le cadre de la classification. La fonction de coût évalue la disparité entre l'étiquette y et la valeur de prédiction de la machine $f(x)$.

pendant la phase d'apprentissage, on préfère souvent la substituer par d'autres fonctions de coût convexes et continues, telles que la fonction « charnière » ou *hinge*.

Risque : une fois que l'on a défini la fonction de coût L, il reste à choisir une mesure afin de quantifier la capacité de la machine. Un des choix les plus répandu est l'espérance de L relativement à x appelée risque structurel R :

$$R(f) = \int_{\mathscr{X}\times\mathscr{Y}} L(y, f(x)) d\mathbb{P}_{X,Y}. \tag{1.3}$$

Plus $R(f)$ est faible plus la machine f est globalement apte à réaliser la tâche. Nous appelons oracle global la fonction $f^\star : \mathscr{X} \to \mathscr{Y}$ tel que le risque soit minimal :

$$f^\star = \underset{f:\mathscr{X}\to\mathscr{Y}}{\arg\min}\, R(f). \tag{1.4}$$

Dans la pratique nous ne connaissons pas $\mathbb{P}_{X,Y}$ et nous ne pouvons calculer ni R ni $f^\star$. Il est classique d'utiliser à la place une approximation appelée risque empirique R_{emp} qui est défini de la façon suivante :

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)), \tag{1.5}$$

avec n le nombre d'individus x.

Construire une machine qui apprend à partir de la minimisation de R_{emp} est appelée stratégie MRE (Minimisation du Risque Empirique, Vapnik [1995]).

Pertinence de la stratégie MRE : afin de pouvoir évaluer le pouvoir d'une machine f à généraliser sur de nouvelles données, il faut que l'approximation $R(f)$ par $R_{emp}(f)$ ne soit pas trop mauvaise. En outre on souhaite que lorsque la taille de l'ensemble d'observation augmente, l'approximation devienne de plus en plus faible. Autrement dit, on aimerait obtenir une convergence en probabilité de $R_{emp}(f)$ vers $R(f)$:

$$\forall \varepsilon, \lim_{n\to\infty} \mathbb{P}(|R_{emp}(f) - R(f)| > \varepsilon) = 0. \tag{1.6}$$

Malheureusement, on peut remarquer que tenter d'appliquer la stratégie MRE, sans introduire le moindre a priori sur la forme de la machine f, est généralement voué à l'échec. Pour en comprendre la raison, il suffit de considérer pour une tâche de régression, la fonction d'interpolation f^{int} définie de la façon suivante :

$$f^{int}(x) = \begin{cases} y_i & \text{si } x = x_i \\ 0 & \text{sinon.} \end{cases} \tag{1.7}$$

On peut constater que la fonction f^{int} minimise $R_{emp}(f)$ mais présente a priori un risque élevé $R(f)$, ainsi choisir f^{int} est un très mauvais candidat. Aussi il est nécessaire de limiter la rechercher de la machine f à un sous-ensemble de fonctions restreint $\mathscr{F}$, afin de vérifier 1.6. Sous cette hypothèse, on espère que plus le nombre d'exemples est important plus l'erreur empirique se rapproche du minimum du risque sur l'ensemble $\mathscr{F}$, c'est-à-dire telle que :

$$\forall \varepsilon, \lim_{n\to\infty} \mathbb{P}(|R_{emp}(f) - R(\tilde{f})| > \varepsilon) = 0 \quad \text{avec} \quad \tilde{f} = \arg\min_{f\in\mathscr{F}} R(f). \tag{1.8}$$

Si les conditions 1.6 et 1.8 sont respectées et que l'ensemble d'observations est suffisamment grand, alors la stratégie MRE est pertinente dans la mesure où 1.6 évite de sélectionner un candidat qui généralise mal et 1.8 assure que f est voisine de $\tilde{f}$. Remarquons cependant que choisir de minimiser $f \in \mathscr{F}$, conduit à exclure des candidats potentiellement intéressants, notamment l'oracle global. Mais on espère néanmoins que l'espace de fonctions $\mathscr{F}$ est suffisamment riche afin que la différence entre le risque de l'oracle global et l'oracle de la classe ne soit pas trop importante, c'est-à-dire :

$$|R(\tilde{f}) - R(f^\star)| \ll 1. \tag{1.9}$$

1.3.3 Compréhension des modèles construits

Un bonne machine d'apprentissage, sous-entend généralement que le modèle appris atteint une bonne performance afin d'accomplir la tâche fixée. Mais on peut aussi désirer avoir un certain contrôle des mécanismes de fonctionnement de la machine afin d'apporter des connaissances supplémentaires sur les données modélisées selon Shmueli [2010]. Pour analyser les causes des erreurs de la machine, par exemple, ou pour savoir s'il est possible d'extraire des informations utiles sur les données par l'étude de l'architecture du modèle. Ainsi il peut être aussi intéressant d'essayer de construire des modèles interprétables. Un tel caractère doit respecter certains types de propriétés afin de faciliter son analyse et d'extraire une certaine intelligibilité de la solution :

- lisibilité : la solution doit être simple, et donc parcimonieuse,
- réalité : le modèle contient un lien réel avec les données,
- interactivité : le modèle évolue en fonction des paramètres.

Nous allons à travers ce prisme étudier quelques algorithmes d'apprentissages classiques et nous interroger sur leur structure vis-à-vis de cette problématique d'interprétabilité de modèle.

1.3.3.1 Interprétation du modèle par les variables

La méthode des arbres de classification introduite par Breiman et al. [1984] consiste à construire un ensemble de règles hiérarchiques sur les variables afin de séparer les données. On classe un nouvel exemple, en parcourant l'arbre par un test mono-variable. Ainsi le modèle est représenté par un certain nombre de règles dont leur hiérarchie met en évidence les variables discriminantes.

L'exemple de classification de la figure 1.6 est issu d'un arbre à deux niveaux. Cette profondeur est ici suffisante pour classer parfaitement les exemples d'apprentissage. Le premier niveau applique une règle de décision $R1$ par seuillage de la variable x_2. Le deuxième niveau de l'arbre applique une deuxième règle $R2$ pour les données prédites dans la classe positive (bleu) au premier niveau par un seuillage sur la variable x_1.

Du point de vue de l'interprétabilité, les arbres apportent une solution intéressante car chaque règle ne considère qu'une seule variable discrète ou continue et le modèle, qui peut être affiché, est lisible par un expert. Cependant, la discrimination n'est opérée que par rapport aux variables et non par rapport aux données.

1.3.3.2 Interprétation du modèle par les prototypes de données

La technique des *k-means*, présentée dans la section 1.2.3.1, est un exemple intéressant de méthode d'apprentissage non supervisé où il est possible d'établir une relation entre le modèle

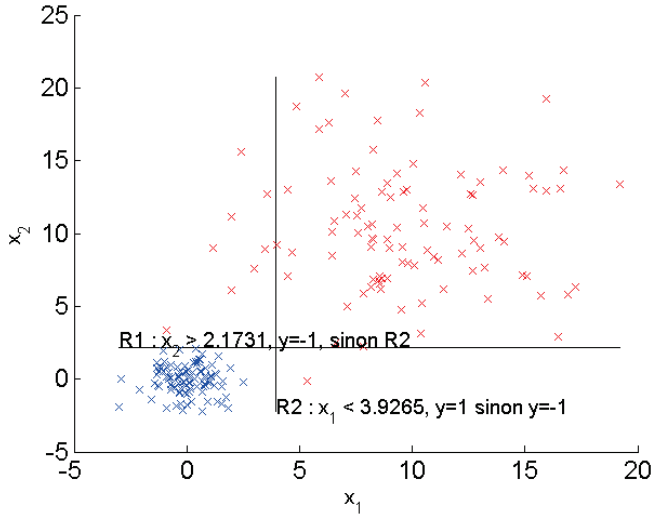


FIGURE 1.6 : La distribution des données est composée de deux classes générées à partir de lois normales, peut être décrite par un arbre à deux niveaux. Le modèle met en évidence la variable la plus pertinente du jeu de donnée.

et les données d'apprentissage. La résolution de ce problème d'optimisation permet de séparer les points en différents clusters (groupes) s_j, dont les moyennes respectives μ_j sont appelées centroïdes. Ainsi le modèle final peut être représenté par la famille de points μ et la classe (numéro de cluster) d'un nouveau point x_{i_0} à classer, est déterminée en choisissant le cluster s_{j_0} dont le centroïde μ_{j_0} associé est le plus proche de x_{i_0}. Il est possible de donner une interprétation des centroïdes μ en les considérant comme des représentants locaux de la distribution de points. Un nouveau point est alors classé selon le représentant local qui lui ressemble le plus.

Nous avons simulé un problème bidimensionnel à deux classes, où les données sont générées à partir de lois normales de variances différentes. Le modèle final, pour $k = 2$, construit 1 centroïde pour chacune des deux classes (voir figure 1.7). Ainsi la distribution des données est décrite à l'aide de 2 points.

Il faut cependant noter que le représentant est un point virtuel, il est donc nécessaire de pouvoir définir ce centroïde quelque soit la nature de la donnée. De plus, le modèle des *k-means* ne renvoyant que des centroïdes les frontières ne tiennent pas compte de la distribution statistique des données, en particulier de leur densité. Cette problématique peut être abordée via les mixtures de gaussiennes. L'approche consiste à faire émerger des clusters tout en indiquant la probabilité d'appartenir à l'ensemble des clusters, dans l'hypothèse où les données sont générées par des lois gaussiennes. Ainsi on peut obtenir des frontières tenant compte de la dispersion des données qui peut ne pas être isotrope (voir figure 1.8).

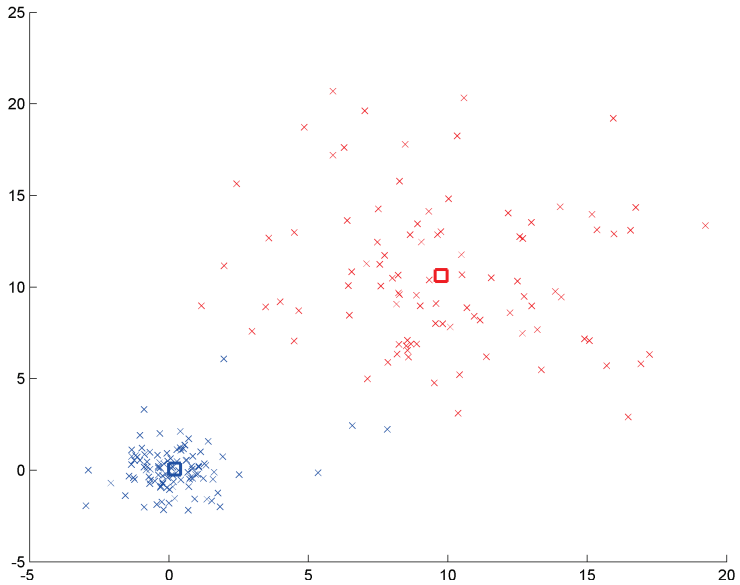


FIGURE 1.7 : La distribution des données est composée de deux classes générées à partir de lois normales, peut être décrite par deux centroïdes construits à l'aide des *k-means*. Les centroïdes sont représentés par des carrés

1.3.3.3 interprétation du modèle par les données

Notons que les modèles des *k-means* et des mixtures de gausiennes construisent des prototypes au sens où les centroïdes ne représentent pas des données réelles mais des moyennes de points. Ces derniers peuvent ne plus avoir de signification physique notamment dans le cas de données discrètes. En ce cas il peut être intéressant d'essayer de générer les centroïdes directement à partir de données réelles. C'est l'approche d'une variante des *k-means* appelée *k-medoid*. Cette méthode introduite par Kaufman and Rousseeuw [2009] permet de lutter plus efficacement contre les données aberrantes en construisant des objets appelés médoides qui sont des points de la distribution dont la distance intra-cluster est la plus faible. Nous avons repris le même exemple que précédemment et nous pouvons constater que cette fois-ci le modèle décrit la distribution de données à l'aide de 2 de ces points sur la figure 1.9.

Les algorithmes présentés ci-dessus sont donc capables d'estimer un modèle à partir de données non-supervisées. Ils n'exploitent cependant que l'information fournie par les individus, sans tenir compte de la véritable étiquette associées à ceux-ci. Afin de pouvoir bénéficier de cette information supplémentaire, il faut appliquer des algorithmes du domaine supervisé.

Le modèle SVM (*Support Vector Machine*, Vapnik [1995]) est l'une des méthodes les plus populaires en classification supervisée. Les bonnes performances de ce modèle trouvent une explication statistique dans la notion de *marge* que le SVM se propose de maximiser [Burges, 1998]. Pour un jeu de données $S_n = (x_i, y_i)_{1 \leq i \leq n}$ de dimension p composé de deux classes

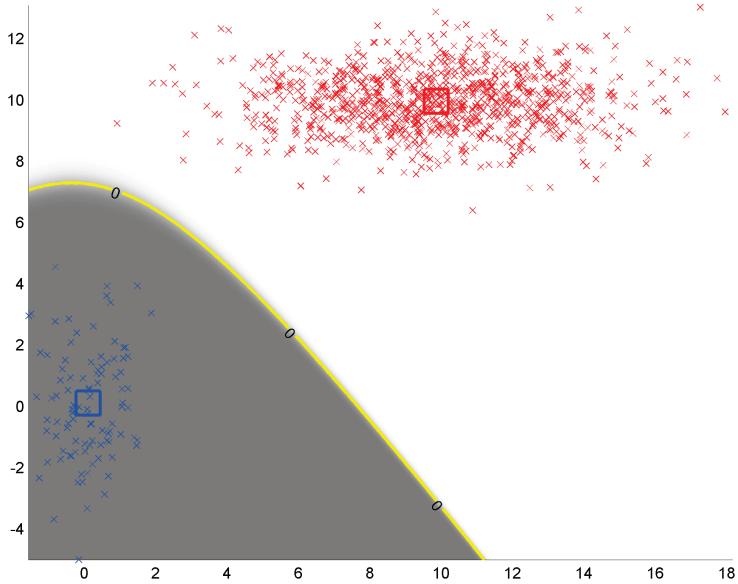


FIGURE 1.8 : La distribution des données est composée de deux classe générées à partir de lois normales, peut être décrite par deux mixtures de gaussiennes définies à partir de deux centroïdes. La frontière de séparation (équiprobabilité) entre les classes est tracée en jaune.

linéairement séparables, la marge m relativement à S_n se définit comme la distance minimale par rapport à un hyperplan $H = \{x \in \mathbb{R}^p, \quad \beta_0 + \beta^T x = 0\}$, avec β les coefficients de l'hyperplan (voir figure 1.10).

Dans le cas où les classes sont linéairement séparables, le modèle SVM est la solution du problème d'optimisation suivant :

$$\begin{cases} (a) & \arg\min_{\beta,\beta_0} \frac{1}{2} \|\beta\|_2^2 \\ (b) & \forall i \in [1,n], \quad 1 - y_i(\beta_0 + \beta^T x_i) \leq 0. \end{cases} \tag{1.10}$$

L'équation (a) est une reformulation équivalente à la maximisation de la marge et l'équation (b) correspond aux contraintes de séparation des deux classes que doit respecter le classifieur. Une propriété intéressante du SVM pour l'interprétabilité est la forme de sa solution $f^*(x) = \beta_0^* + \beta^{*T} x$, qui peut se ré-exprimer directement en fonction des données d'origine :

$$f^*(x) = \beta_0^* + \sum_{i \in VS} \alpha_i \, x_i^T x \quad \text{avec} \quad 0 \leq \alpha_i \leq 1, \tag{1.11}$$

où VS désigne l'ensemble des points supports, c'est à dire aux points qui sont sur la marge (voir figure 1.11). Nous n'explicitons pas ici la signification des poids α de l'équation 1.11 qui seront abordés dans le chapitre 2, mais insistons simplement sur la représentation du modèle qui s'exprime à partir d'un sous-ensemble de points. Les vecteurs supports sont situés à la frontière entre les classes.

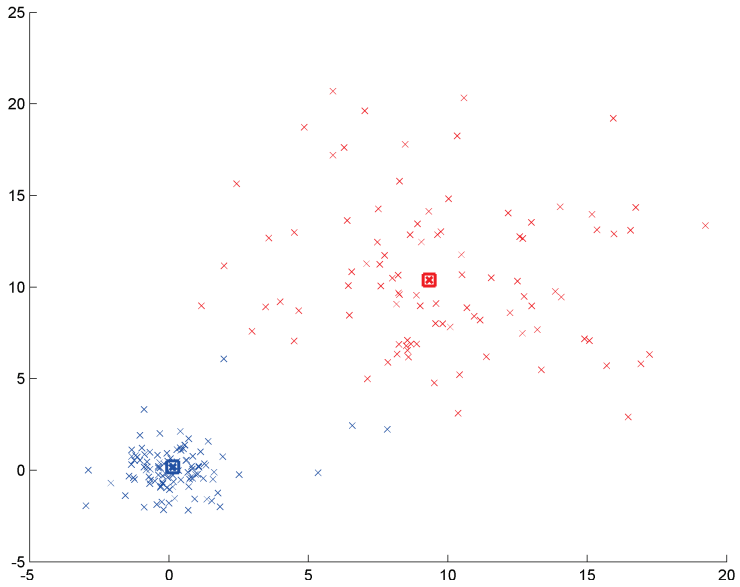


FIGURE 1.9 : La distribution des données est composée de deux classes générées à partir de lois normales, peut être décrite par deux médoides construits à l'aide des *k-medoid*. Les medoides sont représentés par des carrés. Contrairement aux centroïdes générés par la méthode des *k-means*, les médoides sont issus des points de la distribution initiale.

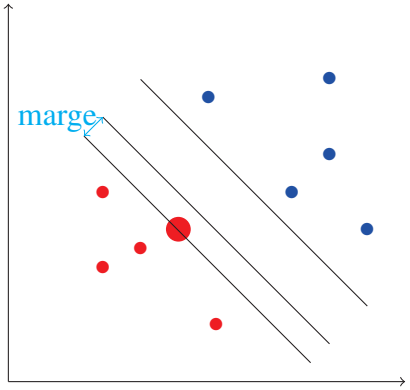


FIGURE 1.10 : Représentation de la marge séparant deux classes.

Dans le cas où les classes ne sont plus séparables, le problème 1.10 n'a pas de solution. L'introduction d'un terme de pénalisation pour les points ne respectant pas la contrainte de séparation conduit à la définition du problème suivant :

$$\begin{cases} (a) & \arg\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ (b) & \forall i \in [1,n], \quad 1 - y_i(\beta_0 + \beta^T x_i) \leq \xi_i. \end{cases} \tag{1.12}$$

L'introduction des variables d'écarts ξ permet de relaxer la contrainte sur la marge. L'influence de la pénalisation sur les points mal classés, c'est à dire enfreignant la contrainte sur la marge est contrôlée par le paramètre C.

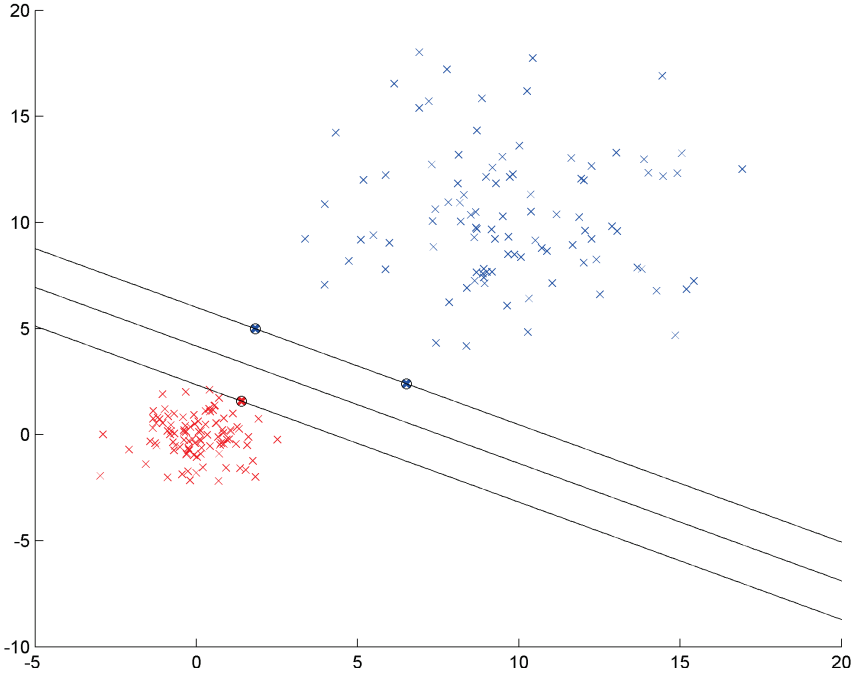


FIGURE 1.11 : La distribution de donnée peut être décrite par trois vecteurs supports situés sur la marge de la frontière de décision du classifieur. Le modèle SVM est représenté par des points de la distribution initiale.

La qualité du modèle et en particulier la forme de la solution construite est fortement dépendante de ce paramètre. Plus généralement, les algorithmes d'apprentissage nécessitent le réglage d'un paramètre : le nombre de clusters, le nombre de mixtures, la profondeur de l'arbre ou encore le poids des points mal classés, respectivement pour les *k-means*, mélanges, arbres de décision et SVM. La valeur imposée à chaque paramètre influe donc non seulement sur la robustesse du modèle, mais également sur son potentiel d'interprétation puisqu'un modèle non pertinent de présentera pas de conclusion satisfaisante.

1.4 Régularisation des modèles d'apprentissage

Nous venons de voir que la qualité d'un modèle n'est pas uniquement liée à la performance atteinte sur les données exemples de la base d'apprentissage (le risque empirique), mais dépend du risque (equation 1.3), la capacité d'un modèle à généraliser la qualité de sa prédiction à l'ensemble des données possibles. L'écart entre le risque et le risque empirique dépend de la complexité de $\mathscr{F}$, l'ensemble des hypothèses considérées. Un des problème de l'apprentissage statistique est d'adapter la complexité de de $\mathscr{F}$ au problème posé. Une mauvaise adaptation conduit aux phénomènes de sur ou sous-apprentissage.

1.4.1 Phénomène de sur-apprentissage et de sous-apprentissage

Le sur-apprentissage est une phénomène qui apparait lors de la construction de la machine. Quand nous adaptons trop la machine aux données d'apprentissage, on perd la capacité de généralisation (voir figure de droite 1.12). Dans ce cas, la stratégie MRE est inefficace car l'ensemble $\mathscr{F}$ étant trop grand, il inclue des fonctions ne respectant pas (1.6) comme par exemple la fonction d'interpolation (voir 1.7). Une façon de réduire le sur-apprentissage consiste à un ajouter un terme dit *régularisant* lors de la minimisation du risque empirique afin de réduire la complexité de $\mathscr{F}$. Cependant si la régularisation est trop importante, l'espace $\mathscr{F}$ devient trop pauvre et cela conduit au phénomène appelé sous-apprentissage (voir figure de gauche 1.12). Dans cette situation, la différence entre le risque de l'oracle globale et l'oracle de $\mathscr{F}$ est trop importante et la condition (1.9) n'est pas respectée.

En termes d'interprétation, le sous-apprentissage ne mettra pas en évidence les spécificités locales permettant de distinguer chaque classe et aboutira donc à une analyse juste globalement, mais inachevée. Inversement, le sur-apprentissage présentera beaucoup trop de cas particuliers sans être capable de définir des comportements globaux sur les critères de distinction.

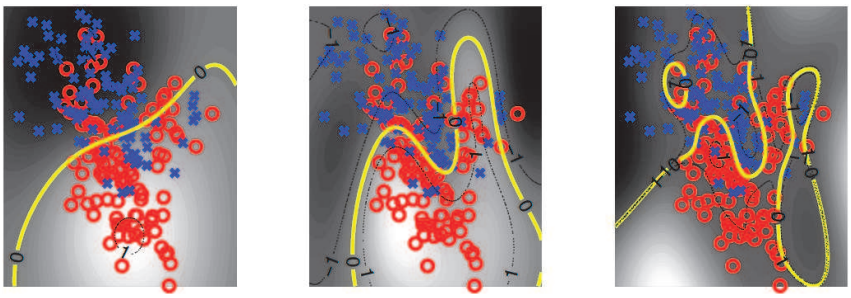


FIGURE 1.12 : Si le modèle s'adapte trop aux données, on peut obtenir une frontière de séparation qui sépare parfaitement les données d'apprentissage mais donc la forme trop complexe entraîne une mauvaise capacité de généralisation (droite). Inversement, une frontière trop simple ne captera pas la topologie locale des données (gauche). Il convient donc de trouver un compromis pour couvrir suffisamment mais sans excès la frontière discriminante (milieu).

Une approche simple pour résoudre cette problématique de sélection de paramètres réside dans l'application d'une stratégie de validation. Pour cela, une deuxième base de données, distincte de celle utilisée pour l'apprentissage et nommée base de test, permet de comparer les performances obtenues pour chaque valeur de paramètre. Par exemple, la validation croisée ou le *boostrapping* permettent de vérifier de manière statistique la fiabilité du modèle. On parle ainsi d'erreur de généralisation qui est étroitement liée à la notion de compromis biais-variance [pour plus de détails voir Hastie et al., 2005, Cornuéjols and Miclet, 2011]. Pour la plupart des modèles ce compromis évolue en fonction de la complexité du modèle. Limiter cette complexité nécessite cependant de pouvoir tout d'abord la quantifier, on parle alors de problème régularisé.

1.4.2 Les problèmes régularisés

Souvent le régularisateur s'identifie à la norme d'un des paramètres de la solution (par exemple les coefficients d'un polynôme), il existe plusieurs choix possibles pour la norme. Selon la norme choisie la solution sera plus ou moins parcimonieuse, c'est à dire que le nombre de paramètres nuls sera plus ou moins grand. Par exemple la norme 1 est appréciée pour sa capacité à obtenir des solutions parcimonieuse, c'est à dire comportant peu de coefficients non nuls. Au final le choix du régularisateur Ω dépend de l'application considérée et surtout de son adéquation mathématique avec la résolution du problème posé.

En effet nous imposons notre a priori que la fonction présente une certaine régularité par le biais d'un terme de lissage. Nous ajoutons au risque empirique un terme de régularisation Ω afin d'éviter le sur-apprentissage, le problème de minimisation devient :

$$\min_{f \in \mathscr{F}} R_{emp}(f) + \lambda \Omega(f). \tag{1.13}$$

Le terme λ est un coefficient de régularisation, il contrôle l'influence du régularisateur sur le risque empirique. Si λ est très faible, Ω n'a plus d'influence sur la solution, on revient à un problème de minimisation non régularisé. Si λ est très grand, Ω lisse fortement la solution. Cela peut entrainer un R_{emp} important car la solution n'est plus assez complexe pour décrire les données (sous-apprentissage). Il y a donc un compromis à trouver sur la valeur de λ.

Un autre manière d'interpréter ce terme et de le voir dans le cadre de la statistique bayésienne comme un maximum a posteriori.

1.4.3 Choix de la régularisation et interprétabilité

Un des modèles les plus standard utilisé en régression, appelé régression *ridge*, intègre au risque empirique des moindres carrés un terme en régularisation en norme 2 sur les coefficients du modèle :

$$\min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_2^2, \tag{1.14}$$

avec β le vecteur des paramètres, $\mathbf{Y}$ le vecteur des étiquettes et $\mathbf{X}$ la matrice des observations (chaque ligne représente un exemple). Cela permet de contrebalancer le comportement des moindres carrés initial qui possède une forte variance et donc est particulièrement sujet au sur-apprentissage. Cependant la norme 2 possède un certain nombre de limitations, notamment une sensibilité au bruit. De plus, elle affecte un poids non nul pour tous les coefficients, ce qui peut gêner la lisibilité du modèle. D'autres pénalisations ont été envisagées pour les moindres carrés notamment la norme 1 qui conduit au problème du LASSO [Tibshirani, 1996] :

$$\min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1. \tag{1.15}$$

Une caractéristique de la pénalisation en norme 1 est d'annuler exactement des coefficients du modèle, ainsi les coefficients présents dans la solution sont discriminants (voir figure 1.13). En comparaison, la norme 2 a davantage tendance à intégrer partiellement des variables dans le modèle, même avec un coefficient très faible, n'ayant pas cette capacité d'annuler complètement un coefficient. Ainsi la pénalisation LASSO est particulièrement intéressante dans le cadre de la problématique d'interprétation de modèle dans la mesure où, tout en réduisant la variance inhérente au modèle des moindres carrés, elle induit une certaine parcimonie dans la solution, ce qui facilite l'analyse de la solution.

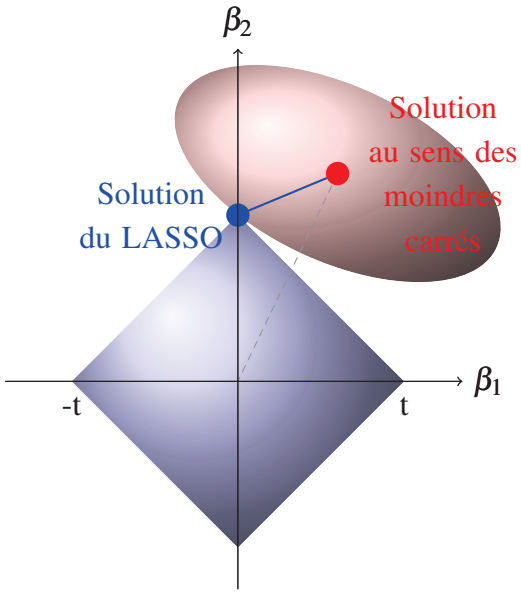


FIGURE 1.13 : La figure illustre le comportement de la norme 1 (LASSO) à induire de la parcimonie au sein du modèle.

Cependant la norme 1 présente aussi certaines limitations dont deux principales : d'une part, le nombre de variables sélectionnables est borné, et d'autre part, des variables discriminantes peuvent être écartées au profit d'une seule si elles sont corrélées entre elles. Pour remédier à ces limitations une norme hybride $L_1 - L_2$ appelée *elastic net* a été proposée et intégrée au sein des moindres carrées [Zou and Hastie, 2005] :

$$\min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2, \tag{1.16}$$

avec λ_1 et λ_2 les coefficients pondérant respectivement l'influence de la norme 1 et de la norme 2 pour la résolution.

1.5 Formulation retenue et orientation des travaux

Dans le cadre de ces travaux, nous nous sommes intéressés à la génération de modèles interprétables dans le cadre de la classification. Nous nous sommes tout d'abord interrogés sur la fonction de coût à utiliser pour la construction du classifieur. La fonction coût de Hinge, choix standard pour la classification, demeure un choix pertinent dans la mesure où, d'une part, elle introduit une pénalisation convexe, contrairement coût 0/1 par exemple. D'autre part il existe une relation entre le risque structurel relativement au Hinge et le risque relativement à l'erreur de classification (voir le théorème 2.31 Steinwart and Christmann [2008])). De plus, la non-différentiabilité génère une certaine parcimonie et conduit à la construction comme dans le cas des SVM à la définition d'ensembles, ce qui permet de donner un sens au modèle et donne des informations sur la localisation des points d'apprentissage dans l'ensemble initial.

Une fois la fonction de coût choisie, il est nécessaire, comme indiqué dans la section précédente, d'introduire un terme de régularisation afin contrôler la complexité du modèle. Dans une optique d'interprétabilité, la norme LASSO, apparaît comme un choix séduisant dans la mesure où elle permet de générer des modèles avec un nombre faible de coefficients exactement non-nuls, ce qui est conforme avec un objectif de simplicité et donc de lisibilité. Cependant elle souffre de plusieurs inconvénients dont notamment une limitation des variables du modèle par le nombre de données ainsi qu'une tendance à retenir de façon relativement arbitraire les variables quand plusieurs sont corrélées entre elles. L'interprétation est dans ce cas faussée par la suppression de variables pertinentes. La pénalité *elastic-net* introduite pour pallier à ces difficultés est en adéquation avec notre motivation d'interprétabilité dans le sens où elle permet de construire des modèles relativement parcimonieux (voir très parcimonieux dans le cas où $\lambda_2 \ll \lambda_1$, ce qui revient à une approximation de la norme LASSO) tout en ayant la possibilité de résoudre le défaut inhérent de la norme 1, qui est susceptible d'exclure du modèle des variables explicatives car corrélées avec une autre variable déjà intégrée au modèle. La pénalité *elastic-net* est capable de fournir des modèles plus riches en interprétation, tout en conservant une bonne capacité de prédiction, gage de cohérence des résultats.

Enfin, il est intéressant de construire un modèle sous forme de prototypes à partir des données d'apprentissage (cette approche sera détaillée dans le chapitre 3) dont le nombre d'éléments peut potentiellement dépasser le nombre de données. Cette situation aboutit au paradigme *Large p Small n* [De Mol et al., 2009], où le problème est a priori mal posé. L'ensemble des prototypes envisagés sont en fait des éléments d'un dictionnaire utilisé à la place des données. Cette définition pose un verrou quant à la définition d'une régularisation adaptée [Donoho and Elad, 2003].

Ces différentes contraintes nous ont conduit à nous intéresser à un problème de classification appelé DRSVM (Doubly Regularized Support Vector Machine) qui intègre la fonction de Hinge ainsi qu'une pénalisation *elastic-net* des coefficients de la solution. Le chapitre suivant se focalise sur cette formulation.

2

Le DRSVM un modèle interprétable ?

Sommaire

2.1 Le DRSVM, un problème de classification avec sélection de variables intelligente 36
2.1.1 Introduction au problème DRSVM 37
2.1.2 Calcul du dual DRSVM 38
2.1.3 Chemins de régularisation 42
2.1.4 Conclusion 49
2.2 DRSVM et chemin de régularisation 49
2.2.1 L'algorithme DRSVM construit un chemin de régularisation 49
2.2.2 Conditions d'optimalité du DRSVM 50
2.2.3 Initialisation 53
2.2.4 Fonctionnement général 53
2.2.5 Simulations et problèmes rencontrés 55
2.3 Proposition d'un chemin pour le DRSVM, via l'analyse de la sous-différentielle 58
2.3.1 Théorie de la sous-différentielle 59
2.3.2 DRSVM et formalisation du chemin en λ_1 63
2.3.3 Initialisation 67
2.3.4 Fonctionnement de l'algorithme après la phase d'initialisation 71
2.3.5 Influence du paramètre λ_2 sur l'évolution de α,γ, β_0 et β 74
2.3.6 Robustesse du λ_1 chemin 75
2.3.7 Simulation de chemin de régularisation λ_1 76
2.4 Conclusion 78

Dans une optique de construction de classifieurs interprétables, le choix de la régularisation est primordiale car elle contrôle la complexité du modèle. L'utilisation de la pénalité LASSO est particulièrement intéressante pour l'interprétabilité, dans la mesure où elle intègre au sein de la solution, les variables les plus discriminantes. De plus elle induit une certaine forme de parcimonie et rend le modèle plus facile à analyser. Mais elle souffre de deux limitations majeures : le nombre de variables sectionnables est borné par le nombre de donnée et elle a tendance à rejeter les variables, même discriminantes, en cas de corrélation entre les variables. Pour pallier à ces défauts, une norme hybride L_1-L_2 appelé *elastic-net* a été proposée.

Dans ce chapitre nous présentons un problème de classification intégrant la norme *elastic net*, appelé DRSVM (paragraphe 2.1). L'analyse montre que c'est un programme quadratique bi-paramétrique. Dans son article séminal sur le sujet, Wang et al. [2006] ont proposé un algorithme de résolution du DRSVM par une technique de chemin de régularisation que nous rappelons paragraphe 2.2. Lors de la mise en œuvre de cette méthode, nous avons rencontré des difficultés liées à la structure de la formulation proposée par Wang. L'analyse de ces problèmes, nous a démontré la nécessité de construire un autre chemin, à partir de la formulation initiale du DRSVM. Ce problème étant non différentiable, nous l'avons étudié à travers le prisme de la théorie de la sous-différentielle, ce qui nous a conduit à proposer un nouvel algorithme de chemin de régularisation plus robuste (paragraphe 2.3), qui est la principale contribution de ce chapitre.

2.1 Le DRSVM, un problème de classification qui inclut une sélection de variables intelligente

Bien que la pénalisation *elastic net* ait été introduite dans un contexte de régression par Zou and Hastie [2005], les auteurs insistent sur la pertinence de l'appliquer également dans le cadre de la classification. Cette motivation repose d'une part, sur une limitation de la norme 1, qui, dans le cas de problèmes de classification où la dimension des données est supérieure au nombre de données, ne peut sélectionner au sein du modèle qu'un nombre borné de variables. D'autre part, la pénalisation elastic-net, contrairement à la norme 1, permet de conserver toutes les variables pertinentes au sein du modèle, même si ces dernières sont fortement corrélées entre elles. Cette dernière propriété étant en adéquation avec notre problématique de recherche de modèle de classification interprétable, nous nous sommes intéressés aux problèmes de classification intégrant une pénalisation de type *elastic-net*.

2.1.1 Introduction au problème DRSVM

Le désir de combiner à une tâche d'apprentissage un objectif de sélection de variable, s'est étendu au domaine de la classification et certains algorithmes de classification standard on été adaptés. Une version alternative aux SVM appelée *1-norm SVM* a été proposée par Zhu et al. [2004] comme la solution du problème suivant pour un échantillon $S_n = (x_i, y_i)_{1 \leq i \leq n} \in (\mathbb{R}^p, \mathbb{R})^n$ donné et un paramètre $\lambda_1 \geq 0$ fixé :

$$\min_{\beta_0 \in \mathbf{R}, \beta \in \mathbf{R}^p} \quad \sum_{i=1}^{n} \max[0, 1 - y_i(\beta_0 + \langle \beta, x_i \rangle)] + \lambda_1 \|\beta\|_1 . \tag{2.1}$$

Le nom *1-norm SVM* se justifie par la forme du problème (2.1) qui évoque celle du SVM standard mais qui se différencie grâce au remplacement de la pénalisation classique L_2 par une pénalisation L_1. Cette modification permet de résoudre des problèmes de classification de façon satisfaisante tout en induisant de la parcimonie dans la solution. Cependant il souffre des limites intrinsèques de la norme L_1 évoquées ci-dessus. Pour cette raison nous nous sommes intéressés à une variante de ce problème appelé DRSVM (*Doubly Regularized Support Vector Machine*) qui intègre une pénalité de type *elastic-net* au sein du problème :

$$\min_{\beta_0, \beta} \quad \sum_{i=1}^{n} \max[0, 1 - y_i(\beta_0 + \langle \beta, x_i \rangle)] + \frac{\lambda_2}{2} \|\beta\|_2 + \lambda_1 \|\beta\|_1 , \tag{2.2}$$

pour $\lambda_1 \geq 0$ et $\lambda_2 \geq 0$ fixés.

Ce problème doublement régularisé peut s'interpréter comme un modèle hybride entre le SVM classique et le SVM L_1. En effet, lorsque $\lambda_1 = 0$, le DRSVM est équivalent au SVM classique. Lorsque $\lambda_2 = 0$, il est équivalent au SVM L_1. Si les problèmes SVM L_1 et DRSVM semblent similaires au SVM, il existe cependant une différence profondes liés à l'introduction de la contrainte sur la norme L_1. La présence de cette dernière perturbe l'interprétation géométrique de la solution des problèmes SVM L_1 et DRSVM , en tant que vecteur support. Cela s'explique par la nature non euclidienne de la norme L_1, c'est-à-dire qu'on ne peut pas lui associer de produit scalaire. Notons qu'il existe une variante du problème DRSVM , appelée *hybrid huberized support vector machines* de Wang et al. [2008]. Dans ce modèle, le terme d'attache aux données (la fonction « charnière » ou *hinge*) est ici remplacée par la fonction de Hubert (voir figure 2.2 qui relaxe la fonction *hinge* en la rendant différentiable ce qui permet de réduire les temps de calcul. Le problème de minimisation se formalise alors de la manière suivante :

$$\min_{\beta_0, \beta} \quad \sum_{i=1}^{n} h(y_i(\beta_0 + \langle \beta, x_i \rangle)) + \frac{\lambda_2}{2} \|\beta\|_2 + \lambda_1 \|\beta\|_1 ,$$

avec h la fonction de Hubert, définie de la façon suivante :

$$h(t) = \begin{cases} 0 & \text{si} \quad t > 1 \\ \dfrac{(1-t)^2}{2\delta} & \text{si} \quad 1-\delta < t \leq 1 \\ 1-t-\frac{\delta}{2} & \text{si} \quad t \leq 1-\delta. \end{cases}$$

Nous pouvons cependant, observer que la fonction de Hubert h fait intervenir un paramètre supplémentaire δ qui nécessite d'être réglé. Or, dans notre optique de construction de classifieur interprétable, nous avons décidé de limiter au maximum les paramètres à régler et donc nous n'avons pas retenu cette méthode. Notons aussi qu'il existe une chemin de régularisation linéaire par morceau en fonction de δ, car comme nous le verrons ce problème est toujours un programme quadratique paramétrique.

Le problème DRSVM (voir figure 2.1) fait intervenir la fonction *hinge* qui est linéaire par morceaux ainsi que 2 termes de pénalisation respectivement linéaire et quadratique, c'est donc un problème d'optimisation quadratique. Afin de nous familiariser avec ce problème, nous allons commencer par dériver son dual dans la section suivante.

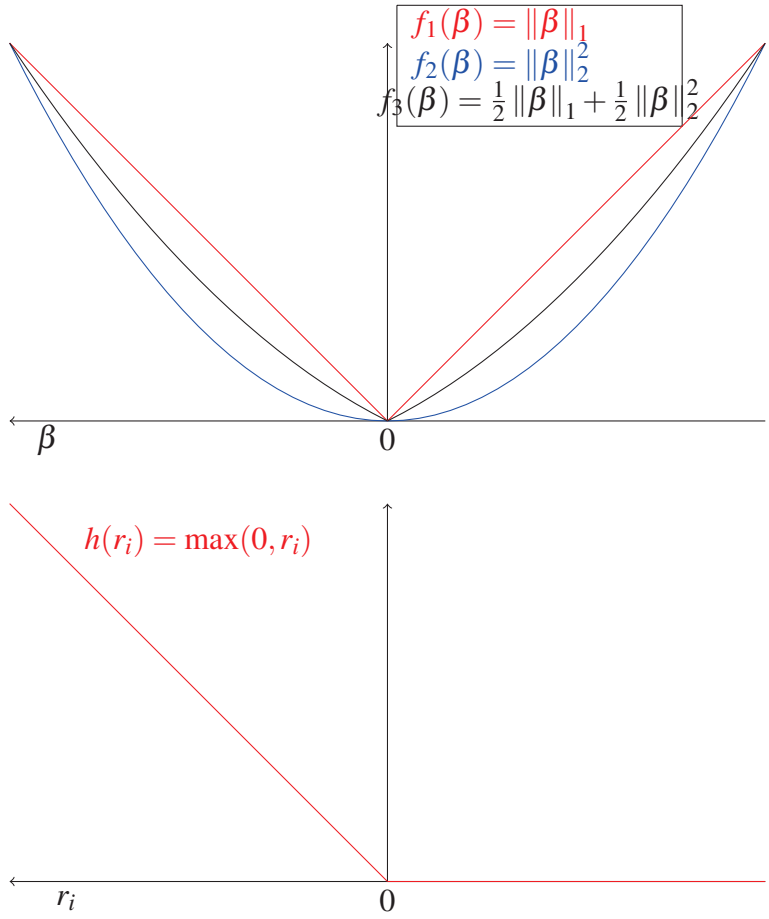
2.1.2 Calcul du dual DRSVM

Afin d'analyser la structure du problème DRSVM et de motiver le choix de la méthode de résolution, nous nous proposons de calculer son problème dual. Le terme non-différentiable du problème DRSVM associé à la fonction *hinge*, est transformé en contraintes différentiables, via l'introduction des variables d'écart ξ. Le problème (2.2) peut alors être reformulé comme le problème d'optimisation sous contraintes suivant :

$$\begin{cases} J(\beta_0,\beta,\xi) = \xi^T \mathbf{1} + \dfrac{\lambda_2}{2}\|\beta\|_2^2 + \lambda_1 \|\beta\|_1 \\ \text{tel que} \quad \forall i \in [1,n]\ ,\ \xi_i \geq 0 & (a) \\ \text{et} \quad \forall i \in [1,n]\ ,\ 1-y_i(\beta_0+\langle\beta,x_i\rangle) \leq \xi_i & (b). \end{cases} \tag{2.3}$$

Notons que le problème (2.3) n'est toujurs pas différentiable à cause de la présence de la norme 1 et cela complique l'étape de différentiation du gradient, nécessaire pour le calcul du dual. Pour contourner cette difficulté nous transformons le système (2.3) en introduisant la variable δ :

$$\begin{cases} J(\beta_0,\beta,\xi,\delta) = \xi^T \mathbf{1} + \dfrac{\lambda_2}{2}\|\beta\|_2^2 + \lambda_1 \|\delta\|_1 \\ \forall i \in [1,n]\ ,\ \xi_i \geq 0 & (a) \\ \forall i \in [1,n]\ ,\ 1-y_i(\beta_0+\langle\beta,x_i\rangle) \leq \xi_i & (b) \\ \delta = \beta & (c). \end{cases} \tag{2.4}$$



avec $r_i(x) = 1 - y_i f(x_i)$

FIGURE 2.1 : Représentation des termes qui composent la fonction de coût associée au DRSVM en deux dimensions. La fonction *hinge* (figure du bas) a un point de singularité en zéro et possède 2 types de comportement : nul et linéaire. La pénalisation elastic-net peut se décrire comme une combinaison de la norme 1 et 2 (figure du haut).

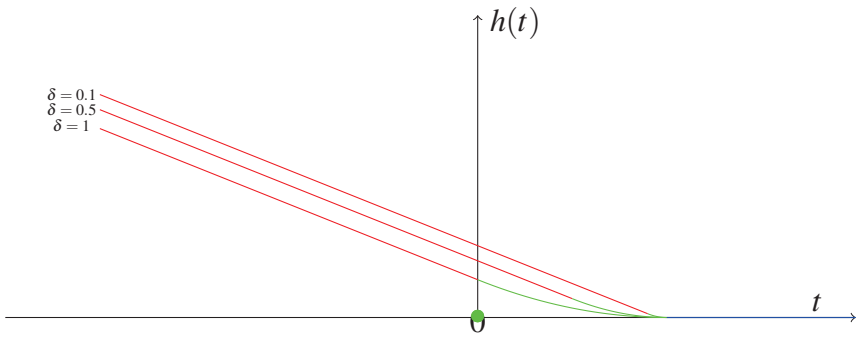


FIGURE 2.2 : Représentation de la fonction de Hubert. Cette fonction est différentiable grâce à l'introduction d'une partie quadratique autour de zéro, dont la longueur de la portion est paramétrée par δ.

Notons que le système (2.4) n'est toujours pas différentiable. Mais l'introduction de la variable δ induit un découplage entre certaines variables et facilite le calcul du gradient du Lagrangien du problème (2.4). Le problème se compose de la fonction de coût J et de $(2n+1)$ contraintes, où n représente le nombre de points dans la base de données. Le Lagrangien associé au problème (2.4) s'écrit de la façon suivante :

$$\begin{cases} L(\beta_0,\beta,\xi,\delta,\alpha,\mu,\psi) = \xi^T\mathbf{1} + \dfrac{\lambda_2}{2}\|\beta\|_2^2 + \lambda_1\|\delta\|_1 - \mu^T\xi \\ \qquad\qquad + \displaystyle\sum_{i=1}^{n}\alpha_i(1-y_i(\beta_0+\langle\beta,x_i\rangle)-\xi_i) + \psi^T(\beta-\delta) \\ \forall i\in[1,n]\ ,\ \mu_i\geq 0 \\ \forall i\in[1,n]\ ,\ \alpha_i\geq 0, \end{cases} \tag{2.5}$$

avec μ, α et ψ les coefficients de Lagrange respectivement associés aux contraintes d'inégalités (a), (b) et à la contrainte d'égalité (c) du problème (2.4).

Une fois le Lagrangien formé, le dual du DRSVM est par définition :

$$\begin{cases} \max\limits_{\mu,\alpha,\psi}\inf\limits_{\beta_0,\beta,\xi,\delta} L(\beta_0,\beta,\xi,\delta,\alpha,\mu,\psi) \\ \forall i\in[1,n]\ ,\ \mu_i\geq 0 \\ \forall i\in[1,n]\ ,\ \alpha_i\geq 0. \end{cases} \tag{2.6}$$

Notons que, selon l'opérateur inf, il y a un découplage possible entre les variables (β_0,β,ξ) et δ. Ainsi il est possible de mener une optimisation indépendante par rapport à (β_0,β,ξ) et δ, en scindant le Lagrangien L en deux fonctions g et h définies comme suit :

$$L(\mu,\alpha,\psi,\beta_0,\beta,\xi,\delta) = g(\mu,\alpha,\psi,\beta_0,\beta,\xi) + h(\psi,\delta)$$

$$\text{avec}\ \begin{cases} g(\mu,\alpha,\psi,\beta_0,\beta,\xi) = \xi^T\mathbf{1} + \dfrac{\lambda_2}{2}\|\beta\|_2^2 - \mu^T\xi + \displaystyle\sum_{i=1}^{n}\alpha_i(1-y_i(\beta_0+\langle\beta,x_i\rangle)-\xi_i) + \psi^T\beta \\ h(\psi,\delta) = \lambda_1\|\delta\|_1 - \psi^T\delta. \end{cases}$$

La fonction g étant différentiable, on peut calculer son gradient :

$$\begin{cases} \nabla_{\beta_0} g(\mu,\alpha,\psi,\beta_0,\beta,\xi) = -\alpha^T y \\ \nabla_{\xi} g(\mu,\alpha,\psi,\beta_0,\beta,\xi) = \mathbf{1}-\alpha-\mu \\ \nabla_{\beta} g(\mu,\alpha,\psi,\beta_0,\beta,\xi) = \lambda_2\beta - \displaystyle\sum_{i=1}^{n}\alpha_i y_i x_i + \psi. \end{cases} \tag{2.7}$$

Les paramètres $(\beta_0^*,\beta^*,\xi^*)$ sont les solutions du problème si $\nabla_{\beta_0,\beta,\xi} g(\mu,\alpha,\psi,\beta_0^*,\beta^*,\xi^*) =$

0, soit :

$$\left\{\begin{array}{lllll} \nabla_{\beta_0} g(\mu,\alpha,\psi,\beta_0^*,\beta^*,\xi^*) & =0 & \Leftrightarrow & \sum_{i=1}^{n}\alpha_i y_i & = 0 \\ \nabla_{\xi} g(\mu,\alpha,\psi,\beta_0^*,\beta^*,\xi^*) & =0 & \Leftrightarrow & 1-\alpha-\mu & = 0 \\ \nabla_{\beta} g(\mu,\alpha,\psi,\beta_0^*,\beta^*,\xi^*) & =0 & \Leftrightarrow & \beta^* & = \frac{1}{\lambda_2}\Big(\sum_{i=1}^{n}\alpha_i y_i x_i - \psi\Big). \end{array}\right. \tag{2.8}$$

Ensuite les conditions sur les paramètres optimaux $(\beta_0^*,\beta^*,\xi^*)$, sont intégrées dans la fonction g :

$$\begin{aligned} g(\mu,\alpha,\psi,\beta_0^*,\beta^*,\xi^*) = {\xi^*}^T\mathbf{1} + \frac{\lambda_2}{2}\left\|\frac{1}{\lambda_2}\Big(\sum_{i=1}^{n}\alpha_i y_i x_i - \psi\Big)\right\|_2^2 - {\xi^*}^T\mu + \\ \sum_{i=1}^{n}\alpha_i\Big[1-y_i(\beta_0^* + \frac{1}{\lambda_2}\Big(\sum_{j=1}^{n}\alpha_j y_j x_j - \psi\Big)\Big]^T x_i - \xi_i^*\Big) + \\ \psi^T\Big[\frac{1}{\lambda_2}\Big(\sum_{i=1}^{n}\alpha_i y_i x_i - \psi\Big)\Big]. \end{aligned}$$

Ensuite le développement des termes donne :

$$\begin{aligned} g(\mu,\alpha,\psi,\beta_0^*,\beta^*,\xi^*) = {\xi^*}^T(\mathbf{1}-\alpha-\mu) + \frac{1}{2\lambda_2}(\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\left\langle x_i,x_j\right\rangle\alpha_j - 2\sum_{i=1}^{n}\alpha_i y_i x_i^T\psi + \|\psi\|_2^2) + \\ \beta_0^*\alpha^T y + \alpha^T\mathbf{1} - \frac{1}{\lambda_2}(\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\left\langle x_i,x_j\right\rangle\alpha_j - \alpha_i y_i x_i^T\psi) + \\ \frac{1}{\lambda_2}(\sum_{i=1}^{n}\alpha_i y_i x_i^T\psi - \|\psi\|_2^2). \end{aligned}$$

Nous pouvons simplifier l'expression en intégrant (a) et (b) de (2.8), en regroupant les différents termes et en introduisant la matrice de gram K :

$$g(\mu,\alpha,\psi,\beta_0^*,\beta^*,\xi^*) = \alpha^T\mathbf{1} + \tfrac{1}{2\lambda_2}\Big(-\alpha^T K\alpha + 2\textstyle\sum_{i=1}^{n}\alpha_i y_i x_i^T\psi - \|\psi\|_2^2\Big),$$

$$\text{avec}\quad K = \Big(y_i\left\langle x_i,x_j\right\rangle y_j\Big)_{1\le i,j\le n}.$$

Pour calculer l'optimum par rapport à la fonction h, on utilise la définition de la fonction conjuguée :

$$\begin{aligned} \inf_{\delta} g(\psi,\delta) &= -\sup_{\delta}(\psi^T\delta - \lambda_1\|\delta\|_1) \\ &= \begin{cases} 0 \text{ si } \|\psi\|_\infty \le \lambda_1 \\ -\infty \text{ sinon.} \end{cases} \end{aligned} \tag{2.9}$$

On remarque que les variables (μ, β_0, ξ) n'apparaissent pas dans le problème dual et ce dernier se reformule donc de la façon suivante :

$$\begin{cases} \max\limits_{\alpha,\psi} \alpha^T \mathbf{1} + \dfrac{1}{2\lambda_2}(-\alpha^T K \alpha + 2\sum\limits_{i=1}^{n} \alpha_i y_i x_i^T \psi - \|\psi\|_2^2) \\ \forall i \in [1,n], \alpha_i \in [0,1] \\ \forall j \in [1,p], \psi_j \in [-\lambda_1, \lambda_1] \\ \text{avec} \quad K = (y_i \langle (x_i, x_j \rangle y_j)_{1 \leq i,j \leq n}. \end{cases}$$

La condition $\alpha_i \in [0,1]$ provient du fait qu'à l'optimum (2.8) requiert $\mathbf{1} - \alpha - \mu = 0$, $\alpha \leq 0$ et $\mu \leq 0$.

2.1.3 Chemins de régularisation

2.1.3.1 Principe sous-jacent

Bien que les chemins de régularisation aient été popularisés dans le courant des années 2000 au sein de la communauté de machine learning, l'idée sous-jacente est relativement ancienne et peut-être rapprochée aux travaux de Markowitz des années 50 portant sur les modèles de choix de portefeuille, qui lui ont valu le Nobel d'économie en 1990. Afin de minimiser la variance, il a étudié les problèmes paramétriques quadratiques et s'est intéressé à certains domaines de l'espace des paramètres, où la fonction objectif évolue de façon linéaire à mesure que l'on fait varier le paramètre [Berkelaar et al., 1997]. La problématique plus générale consistant à s'interroger sur la capacité de prédiction de l'évolution de la solution d'un système relativement à la variation d'un paramètre, a émergé dans des contextes divers, par exemple dans le cadre de l'analyse de sensibilité de système [Gal, 1995, Heller, 1954]. L'importance de cette problématique a entraîné, en vue de sa résolution, l'émergence d'approches bien différentes telles que les méthodes d'homotopies de Osborne et al. [2000], de *prédiction / correction* [Park and Hastie, 2007] ou de recherche par composante de Friedman et al. [2007]. Parallèlement, la focalisation sur les problèmes dont la solution varie linéairement par morceaux relativement au paramètre, a conduit à la définition et à l'étude d'une classe de problèmes connus sous le nom *bi-parametric quadratic programming* [voir par exemple Jansen, 1997, Ghaffari-Hadigheh et al., 2008]. Un problème bi-paramétrique quadratique se définit de la façon suivante :

$$\begin{cases} \min\limits_{x \in \mathbb{R}^p} \ \dfrac{1}{2} x^T Q x + (c + \lambda \Delta c)^T x & \\ \qquad Ax = b + \mu \Delta b & (a) \\ \qquad x \geq 0 & (b), \end{cases} \tag{2.10}$$

avec Q une matrice de taille $p \times p$ symétrique définie positive [1], A une matrice de taille $m \times p$

1. si Q est définie positive elle est inversible et toutes ses sous-matrices le sont aussi

de rang $\min(p,m)$, b et Δb des vecteurs de taille m, c et Δc des vecteurs de de taille p, où p et m représentent respectivement la dimension des données et le nombre de contraintes linéaires du problème. Les paramètres λ et μ sont des réels positifs ou nuls.

Le problème quadratique (2.10) est doublement paramétré à travers les termes linéaires c et b qui varient respectivement selon les paramètres λ et μ, engendrent une variation quadratique de la fonction objectif $J(x) = \frac{1}{2}x^T Qx + (c + \lambda\Delta c)^T x$ et une variation linéaire de la solution x^* [Jansen, 1997]. Afin d'établir ce résultat il faut annuler le gradient du Lagrangien L associé au système (2.10), qui est défini comme suit :

$$L(x,\alpha,\gamma) = \frac{1}{2}x^T Qx + (c+\lambda\Delta c)^T x + \alpha^T(Ax - b - \mu\Delta b) - \gamma^T x,$$

avec α et γ les coefficients de Lagrange du problème.

Remarquons que γ doit être positif car il est associé à des contraintes d'inégalités. Ensuite le gradient de L se dérive naturellement :

$$\nabla_x L(x,\alpha,\gamma) = Qx + c + \lambda\Delta c + A^T\alpha - \gamma. \tag{2.11}$$

Les conditions d'optimalité (KKT), l'annulation du gradient (2.11), les conditions d'admissibilité et les conditions de complémentarité [2] conduisent à :

$$\begin{cases} Qx + c + \lambda\Delta c + A^T\alpha - \gamma & = 0 & (a) \\ Ax - b - \mu\Delta b & = 0 & (b) \\ x & \geq 0 & (c) \\ \gamma & \geq 0 & (d) \\ \gamma_j x_j & = 0, \quad \forall j \in [1,p]. & (e) \end{cases} \tag{2.12}$$

Remarquons que d'après la condition (2.12.e) on a $\gamma_j > 0 \Rightarrow x_j = 0$ et $x_j > 0 \Rightarrow \gamma_j = 0$. Ainsi il est possible de partitionner l'ensemble des variables (x,γ) en introduisant les ensembles $I_1 = \{j \in [1,p], \quad x_j > 0\}$ et $I_0 = \{j \in [1,p], \quad x_j = 0\}$. Soit w un vecteur, M une matrice, I et J deux ensembles d'indices, nous notons w_I et $M_{I,J}$ respectivement le vecteur formé par les composantes de w dont l'indice appartient à J, et la sous-matrice formée par les éléments de M dont l'indice appartient à $I \times J$. On peut réécrire le système (2.12) de la manière suivante :

$$Mz = v$$

$$\text{avec } z = \begin{pmatrix} x_{I_1}^T \\ \alpha \\ \gamma_{I_0} \end{pmatrix}, \; v(\lambda,\mu) = \begin{pmatrix} c_{I_1} + \lambda\Delta c_{I_1} \\ b + \mu\Delta b \\ c_{I_0} + \lambda\Delta c_{I_0} \end{pmatrix}, \; M = \begin{pmatrix} Q_{I_1,I_1} & A_{[1,m],I_1}^T & 0_{I_1,I_0} \\ A_{[1,m],I_1} & 0_{[1,m],[1,m]} & 0_{[1,m],I_0} \\ 0_{I_0,I_1} & A_{[1,m],I_0}^T & I_{I_0,I_0} \end{pmatrix}.$$

2. pour des détails sur les conditions KKT voir le chapitre 16 de Nocedal and Wright [2006]

Soit $(x^*, \gamma^*, \alpha^*)$ la solution du système (2.12) avec $\lambda = 0$ et $\mu = 0$. Si on note et $(x^*_{\lambda_1}, \alpha^*_{\lambda_1}, \gamma^*_{\lambda_1})$ la solution de (2.12) pour $\lambda = \lambda_1$ et $\mu = 0$ avec pour hypothèse que pour la valeur λ_1, les ensembles I_0 et I_1 ne changent pas et que la structure de (2.13) est préservée. Alors on a $z(x^*, \gamma^*, \alpha^*) = M^{-1}v(0,0)$, $z(x^*_{\lambda_1}, \gamma^*_{\lambda_1}, \alpha^*_{\lambda_1}) = M^{-1}v(\lambda_1, 0)$ et

$$z(x^*_{\lambda_1}, \gamma^*_{\lambda_1}, \alpha^*_{\lambda_1}) - z(x^*, \gamma^*, \alpha^*) = \lambda_1 (d_x, d_\gamma, d_\alpha)^T \qquad \text{avec} \quad (d_x, d_\gamma, d_\alpha)^T = M^{-1}(\Delta c_{I_1}, 0, \Delta c_{I_0})^T.$$

Ainsi

$$x^*_{\lambda_1} = x^* + \lambda_1 d_x,$$

et la solution est donc linéaire par rapport à λ.

Si on note $(x^*_{\mu_1}, \alpha^*_{\mu}, \gamma^*_{\mu})$ la solution de (2.12) avec $\lambda = 0$ et $\mu = \mu_1$ avec pour hypothèse que pour la valeur μ_1, les ensembles I_0 et I_1 ne changent pas et que la structure de (2.13) est préservée. Alors on a $z(x^*_{\mu_1}, \gamma^*_{\mu_1}, \alpha^*_{\mu_1}) = M^{-1}v(0, \mu_1))$ et

$$z(x^*_{\mu_1}, \gamma^*_{\mu_1}, \alpha^*_{\mu_1}) - z(x^*, \gamma^*, \alpha^*) = \mu_1 (d'_x, d'_\gamma, d'_\alpha)^T \qquad \text{avec} \quad (d'_x, d'_\gamma, d'_\alpha)^T = M^{-1}(0, \Delta b, 0)^T.$$

Ainsi

$$x^*_{\mu_1} = x^* + \mu_1 d'_x,$$

et la solution est donc aussi linéaire par rapport à μ.

Dans les deux situations, un changement des ensembles I_1 et I_0 entraîne une modification du système (2.12), de la matrice M et des termes d_x et d'_x. Le comportement reste linéaire mais les pentes sont différentes. C'est en ce sens que l'on parle d'*évolution linéaire par morceaux* de la solution. Les valeurs $\lambda^*(\mu) = (\lambda_i(\mu))_{1 \leq i \leq n_\mu}$ et $\mu^*(\lambda) = (\mu_i(\lambda))_{1 \leq i \leq n_\lambda}$ pour lesquelles se produit une modification des ensembles I_1 et I_0 sont appelés *points de rupture*. L'ensemble des solutions $x^*(\lambda, \mu)$ représenté dans un espace 2D dont les axes correspondent respectivement à λ et μ, définissent une surface à laquelle on peut associer le pavage définit par l'ensemble des points de rupture $P = \{(\lambda, \mu) \in \mathbb{R}^2, \text{ tel que } \lambda \in \lambda^* \text{ et } \mu \in \mu^*\}$. C'est à cause de cette dépendance envers ces deux paramètres que le problème (2.10) est appelé *bi-paramétrique*. La figure 2.3 montre un exemple de solution à ce type de problème.

2.1.3.2 Les chemins de régularisation

C'est à travers l'algorithme du LARS [Efron et al., 2004] donnant une résolution efficace du problème du LASSO (1.15), que les chemins de régularisation ont été popularisés au sein de la communauté machine learning. Le problème du LASSO fait intervenir un paramètre λ contrôlant l'influence de la pénalisation en norme 1 de la solution et donc la parcimonie du modèle. Le LASSO appartient à la classe générale des problèmes régularisés, qui, dans leur formulation font intervenir un terme dit *régularisant*, afin de contrôler la complexité du modèle

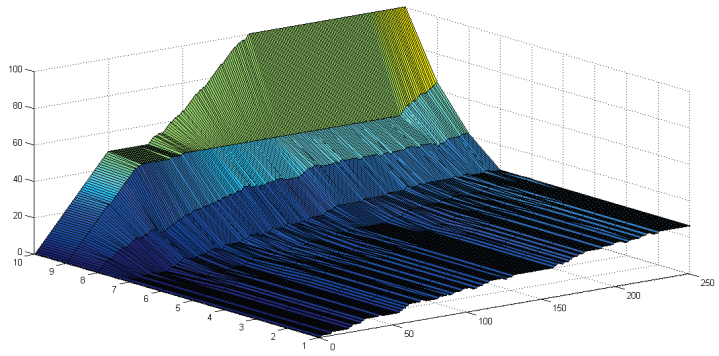


FIGURE 2.3 : Un exemple de l'ensemble des solutions (pour une composante) d'un QP bi-paramétrique (2.10) en fonction de ces deux paramètres λ et μ. On voit que cet ensemble de solutions définit une surface qui est un pavage bilinéaire entièrement spécifié par ses points de rupture.

pour limiter notamment le phénomène de sur-apprentissage. Un problème régularisé peut se formaliser de la manière suivante :

$$\beta^*(\lambda) = \underset{\beta \in \mathbb{R}^p}{\arg\min} \quad J(S_n, \beta) + \lambda \, Pen(\beta),$$

avec $S_n = (x_i, y_i)_{1 \leq i \leq n} \in (\mathbb{R}^p, \mathbb{R})^n$ un échantillon de données, β le modèle, J la fonction de coût, *Pen* le terme de pénalisation et λ le paramètre de régularisation associé.

Le chemin de régularisation se définit comme l'ensemble des solutions β^* associé à la plage de valeur $\lambda \in [0, +\infty]$. On parle de *chemin* dans le sens où β^* est une fonction de la variable positive λ, dont il est possible de donner une représentation graphique (voir figure 2.4). Un cas particulièrement adapté à l'utilisation des chemins de régularisation, est celui où la solution varie linéairement par morceaux par rapport à λ. En effet dans cette configuration, il est possible de calculer les dérivées de la solution relativement à λ via l'inversion d'un système linéaire [Rosset and Zhu, 2007] :

$$\frac{\partial \beta^*}{\partial \lambda} = M^{-1} v. \tag{2.13}$$

Ensuite il faut détecter les points de rupture pendant une phase où on recense toutes les contraintes du problème susceptibles d'être enfreintes et de conduire à la modification des dérivées de β. Le point de rupture correspond à $\lambda_r = \lambda + \delta_\lambda^*$ où δ_λ^* représente le pas maximum pour lequel on peut faire évoluer la solution, jusqu'à ce qu'une contrainte soit enfreinte. Ensuite la mise à jour de la solution β^* est aisée à effectuer :

$$\beta^* \leftarrow \beta^* + \delta_\lambda^* \frac{\partial \beta^*}{\partial \lambda}. \tag{2.14}$$

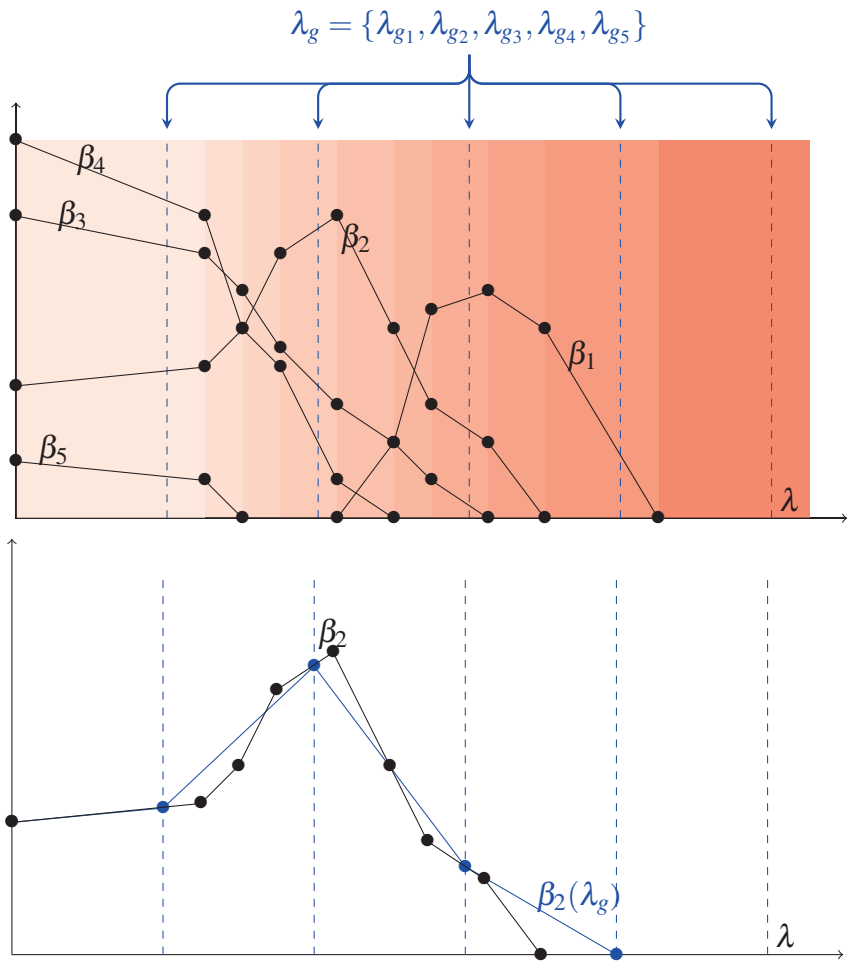
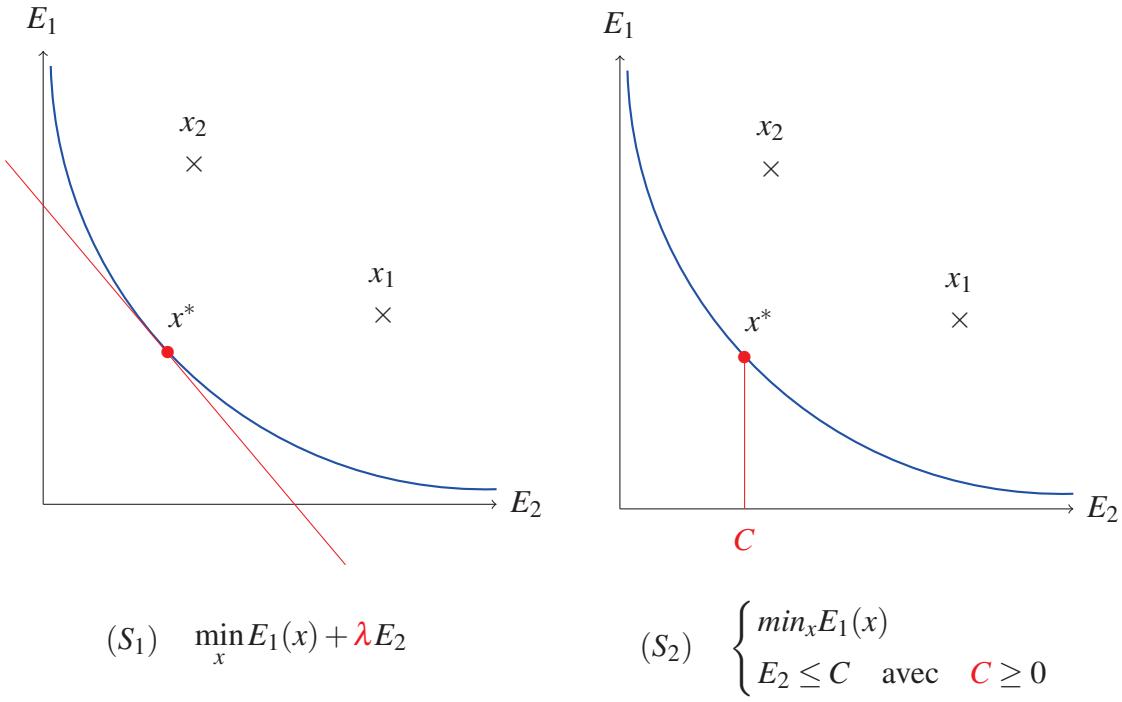


FIGURE 2.4 : La solution $\beta = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$ d'un problème paramétré (S) par λ est affichée sur la figure. Ici l'évolution étant linéaire par morceaux, il suffit pour prédire l'évolution de β de connaître les pentes (c'est-à-dire la dérivée de β par rapport à λ) ainsi que les points de rupture de pente (points noirs). La connaissance de ces 2 informations permet aussi de comprendre la dynamique de l'évolution de β et d'interpréter le modèle. Par exemple la variable β_1 est sélectionnée en premier dans le modèle mais est rejetée par la suite, au profit d'autres variables tel que β_2 ou β_3. Le vecteur $\lambda_g = \{\lambda_{g_1}, \lambda_{g_2}, \lambda_{g_3}, \lambda_{g_4}, \lambda_{g_5}\}$ représente une famille de paramètres pour laquelle on résout le problème (S). La famille λ_g n'est pas très bien adaptée à (S) dans la mesure où elle n'est pas bien calibrée par rapport à la dynamique de l'évolution de la solution. Par exemple la plage [λ_{g_2},λ_{g_3}] est trop large, il y a ajout de la variable β_1 et rejet de la variable β_4 au sein du modèle. Au contraire la valeur de λ_{g_5} est trop élevée et la régularisation est trop forte, on obtient alors la solution $\beta = 0$. Nous avons affiché sur la figure du bas une superposition entre l'évolution de la variable β_2 sur l'ensemble du chemin de régularisation et l'interpolation de la valeur de β_2 à partir de la famille λ_g. On constate des différences non négligeables entre les deux courbes, dues à la mauvaise calibration de la famille λ_g.



$$(S_1) \quad \min_x E_1(x) + \lambda E_2 \qquad (S_2) \quad \begin{cases} min_x E_1(x) \\ E_2 \leq C \quad \text{avec} \quad C \geq 0 \end{cases}$$

FIGURE 2.5 : Le front de Pareto est symbolisé par la courbe bleu, il représente l'ensemble des solutions non dominées. C'est le cas de la solution x^* qui peut s'identifier à l'optimum du problème (S_1) pour une certaine valeur de λ. Dans le cas où E_1 et E_2 sont convexes, le front de Pareto l'est également et il existe une équivalence entre (S_1) et (S_2). C'est à dire il existe une valeur C telle que x^* est solution de (S_1) et (S_2). Construire le chemin de régularisation par rapport à λ ou C est équivalent à parcourir le front de Pareto.

Dans le cadre de la sélection de modèle, la méthode des chemins de régularisation est particulièrement séduisante car elle permet de s'affranchir de l'utilisation des grilles de paramètres et retourne l'ensemble des solutions optimales (voir figure 2.4). La calibration d'une grille de paramètres est un problème délicat car on ne sait pas a priori quel intervalle $[\lambda_{\min}, \lambda_{\max}]$ et quel pas d'échantillonnage choisir. De mauvaises valeurs de bornes conduisent à explorer un mauvais domaine de l'espace des paramètres. Un pas d'échantillonnage trop faible augmente, d'une part, considérablement le temps de calcul et peut d'autre part aboutir à balayer des zones de paramètres, peu dynamiques où la solution n'évolue presque pas.

Ainsi un certain nombre de problèmes d'apprentissage ont été ré-étudiés afin d'analyser la possibilité de construction de leur chemin de régularisation. Cela a donné naissance à de nombreux algorithmes de chemin comme par exemple Hastie et al. [2004], Zou and Hastie [2005], Yuan and Lin [2006].

2.1.3.3 Relation des chemins de régularisations avec le front de Pareto

La notion front de Pareto a été introduite dans le domaine de l'économie (pour plus de détail voir Ruzika and Wiecek [2005]) où on s'intéresse à la minimisation simultanée de deux critères. Soit E_1 et E_2 des fonctions à valeurs positives, dépendant d'une variable x. La possible inter-dépendance de E_1 et E_2 amène une interrogation sur la valeur de x à sélectionner, dans le sens où une valeur x^* peut être un bon choix pour E_1 mais très mauvais pour E_2 et inversement. Une représentation pertinente consiste à représenter l'ensemble des solutions x en deux dimensions, dont les axes correspondent respectivement à E_1 et E_2 (voir figure 2.5).

On peut distinguer une zone représentant l'ensemble des valeurs possibles des solutions. Une solution x est dite *dominée* s'il existe une solution x' tel que : $E_1(x') \leq E_1(x)$ et $E_2(x') \leq E_2(x)$. Dans ce cas il est préférable de choisir x' plutôt que x. Le front de Pareto se définit comme l'ensemble des solutions non dominées. Si de plus les termes $E_1(x)$ et $E_2(x)$ sont convexes (cette hypothèse est vérifiée en particulier pour les problèmes quadratiques bi-paramétrique (2.10)) alors le front de Pareto est lui même convexe et comme on peut l'observer sur la figure 2.5, les chemins des problèmes (S_1) et (S_2) relativement respectivement à λ et μ peuvent s'interpréter comme le parcours du front de Pareto.

2.1.3.4 Limitations des chemins de régularisation

Si la méthode du chemin de régularisation présente de nombreux avantages, elle n'est pourtant pas exempte d'inconvénients. Elle est déjà réputée pour faire montre d'une certaine instabilité. Si à une étape du calcul, pour des raisons numériques, on commence à s'écarter un peu du chemin, la méthode tend alors à diverger. Afin de remédier à cet inconvénient et stabiliser l'algorithme, il est courant d'intégrer un mécanisme de correction, permettant de recalculer une solution exacte sur le chemin en résolvant le problème d'optimisation pour une valeur donnée du paramètre de régularisation. Autre inconvénient, le calcul du chemin peut aussi s'avérer, si l'on n'y prend gare, d'une complexité de calcul excessive. Si le problème est de grande taille et que l'on cherche à partir de la « mauvaise » extrémité, là où tous les points sont actifs, le calcul de ce point initial peut s'avérer extrêmement coûteux. Même en partant de la bonne extrémité, là où tous les coefficients sont nuls, le nombre de points de rupture du chemin s'accroît proportionnellement à la dimension du problème et la distance entre deux ruptures peut être si petite qu'elle entraîne mécaniquement des erreurs numériques qui débouchent là encore sur une instabilité. Pour remédier à ce problème, il existe des méthodes proposant de calculer des approximations sous-optimales du chemin (voir par exemple Karasuyama and Takeuchi [2011]).

2.1.4 Conclusion

La méthode des chemins de régularisation s'avère être particulièrement pertinente dans la mesure où elle permet de suivre la dynamique de l'évolution de la solution que l'on souhaite analyser. Ce problème du DRSVM a été introduit par Wang et al. [2006] avec un algorithme permettant de construire un chemin de régularisation. Une autre méthode, plus efficace, pour résoudre le problème DRSVM à été ensuite proposée par Ye et al. [2011]. Cependant cette approche résout le problème relativement à un couple de paramètres (λ_1, λ_2) donné, perdant l'aspect interprétable des chemin de régularisation pour l'analyse de la solution. Nous avons donc retenu l'algorithme de Wang afin d'analyser l'évolution de la forme de la solution.

2.2 DRSVM et chemin de régularisation

Dans cette section nous allons nous intéresser à la résolution du DRSVM, via la technique des chemins de régularisation. Puis nous analyserons l'algorithme de Wang et al. [2006] qui calcule le chemin de régularisation du DRSVM, via la résolution d'un problème alternatif. Enfin nous allons mettre en évidence certaines limitations de cette approche via l'étude d'un exemple caractéristique.

2.2.1 L'algorithme DRSVM construit un chemin de régularisation

De part sa structure le problème DRSVM (équation (2.2)) fait intervenir les paramètres λ_1 et λ_2, respectivement associés aux termes de pénalisation L_1 et L_2. Dans le cadre d'une tâche d'apprentissage, ils nécessitent d'être réglés afin d'optimiser les performances du classifieur. Dans cette optique, la construction d'un chemin de régularisation par rapport à l'un des paramètres, est une approche intéressante dans la mesure où elle détermine toutes les solutions du problème. D'autre part, la résolution directe du problème d'optimisation n'est pas toujours possible pour des problèmes en grandes dimensions et il est alors judicieux de construire le modèle à partir d'une solution simple et l'enrichir itérativement.

Une analyse des conditions d'optimalité du problème DRSVM a conduit Wang et al. [2006] à proposer deux algorithmes qui construisent respectivement les chemins par rapport à la régularisation L_1 et L_2. Dans le cadre de notre problématique de construction d'un classifieur interprétable, nous nous sommes plus particulièrement intéressés à l'algorithme du chemin en L_1 que nous avons analysé et mis en œuvre. Nous rappelons la forme initiale du problème DRSVM :

$$\min_{\beta_0,\beta} \quad \sum_{i=1}^{n} \max[0, 1 - y_i(\beta_0 + \langle \beta, x_i \rangle)] + \frac{\lambda_2}{2} \|\beta\|_2 + \lambda_1 \|\beta\|_1 .$$

Le problème faisant intervenir la fonction *hinge* ainsi que la norme L_1, il est non-différentiable. Wang propose de résoudre un problème différentiable équivalent, en transformant sous la forme de contraintes les termes non-différentiables de la fonction de coût initiale. Cela est licite car le problème (2.2) étant convexe, on peut alors montrer que pour toute valeur de λ_1, il existe un réel positif s tel que ce dernier est équivalent au problème d'optimisation suivant :

$$\begin{cases} \min\limits_{\beta_0,\beta,\varepsilon} \sum\limits_{i=1}^{n} \varepsilon_i + \dfrac{\lambda_2}{2} \|\beta\|_2^2 & \\ \qquad \forall i \in [1,n], \quad 1 - y_i(\beta_0 + \langle \beta, x_i \rangle) \leq \varepsilon_i & (a) \\ \qquad \forall i \in [1,n], \quad \varepsilon_i \geq 0 & (b) \\ \qquad \|\beta\|_1 \leq s & (c). \end{cases} \tag{2.15}$$

A partir du problème (2.15) posé, Wang calcule les conditions d'optimalité associées afin de construire un chemin de régularisation par rapport à s. Ce paramètre contrôle l'influence de la pénalisation sur la norme 1 de β. Plus sa valeur est élevée, plus la contrainte est relâchée, et moins la norme 1 est pénalisée. En début de chemin ($s = 0$), la solution est très parcimonieuse et $f(\cdot) = \beta_0$ car la pénalisation est infinie. Au fur et à mesure que nous relâchons la contrainte, des composantes de β deviennent non nulles et la solution est de plus en plus complexe. En fin de chemin (c'est-à-dire quand la contrainte n'est plus active), la solution est équivalente à un SVM classique. Notons que les chemins de régularisation sont astucieusement utilisés pour les problèmes dont les paramètres évoluent de façon linéaire par morceaux quand nous relâchons la contrainte. En effet, il suffit alors de repérer les points de rupture où le système change, qui se traduisent par une variation de la pente des paramètres du système. Entre deux points de rupture, l'évolution des paramètres étant linéaire, nous pouvons alors facilement calculer leur variation sur chaque segment. Nous allons commencer par détailler les calculs des conditions d'optimalité qui conduisent à l'établissement d'une linéarité par morceaux du DRSVM.

2.2.2 Conditions d'optimalité du DRSVM

Les conditions d'optimalité fournissent de précieuses informations sur la forme de la solution recherchée. Afin d'établir le chemin de régularisation, nous allons utiliser les conditions d'optimalité afin d'analyser l'évolution de la solution, en fonction du paramètre par rapport auquel on souhaite construire le chemin. Quelques notations utilisées pendant la construction du chemin sont tout d'abord introduites.

Soit $r = (r_i)_{1 \leq i \leq n}$, le vecteur résidu associé aux points de la base de donnée :

$$r_i = 1 - y_i(\beta_0 + \langle \beta, x_i \rangle), \quad \forall i \in [1,n]. \tag{2.16}$$

Pour un point x_i donné, le résidu r_i associé, quantifie la manière dont x_i enfreint la contrainte sur la marge. Plus r_i est grand et plus ε_i est grand, cela se répercute par une augmentation de la valeur de la fonction objectif de (2.15).

Soit $c = (c_j)_{1\leq j\leq p}$, un vecteur appelé corrélation généralisée qui se définit de la façon suivante :

$$c_j = \lambda_2\beta_j - \sum_{i=1}^{p} \alpha_i y_i x_{ij}. \tag{2.17}$$

Pour une variable j_0 donnée, la corrélation généralisée quantifie le pouvoir de discrimination des données. Plus les variables sont pertinentes et plus la valeur de leur corrélation généralisée est élevée.

Les variables r et c sont des artifices de calcul utilisés lors de la phase de détection de rupture de pente du chemin. Notons que leur introduction découle indirectement de la non-différentiabilité de la fonction *hinge* et de la norme 1.

Il est aussi nécessaire d'introduire des ensembles de points et de variables pour gérer la phase de différentiation DRSVM :

- $\mathscr{R} = \{i \in [1,n],\ r_i < 0\}$ (points bien classés)
- $\mathscr{E} = \{i \in [1,n],\ r_i = 0\}$ (points sur la marge)
- $\mathscr{L} = \{i \in [1,n],\ r_i > 0\}$ (points saturés)
- $\mathscr{V}_\beta = \{j \in [1,p],\ \beta_j \neq 0\}$ (variables actives)
- $\mathscr{V}_0 = \{j \in [1,p],\ \beta_j = 0\}$ (variables inactives)

Enfin le Lagrangien L du problème (2.15) se définit de la façon suivante :

$$L(\beta,\beta_0,\varepsilon,\alpha,\mu,\eta) = \sum_{i=1}^{n}\varepsilon_i + \lambda_2\frac{\|\beta\|_2^2}{2} + \sum_{i=1}^{n}\alpha_i(1 - y_i(\beta_0 + \langle\beta, x_i\rangle) - \varepsilon_i) \tag{2.18}$$

$$- \sum_{i=1}^{n}\mu_i\varepsilon_i + \eta(\|\beta\|_1 - s). \tag{2.19}$$

Les variables α, μ et η représentent les coefficients de Lagrange respectivement associés aux contraintes (a), (b) et (c) du problème (2.15). Les conditions d'optimalité sont obtenues par l'annulation du gradient du lagrangien par rapport à $\beta_{\mathscr{V}_\beta}$ (les composantes actives de β), β_0 et ε :

$$\begin{cases} \forall j \in \mathscr{V}_\beta\ , & \lambda_2\beta - \sum_{i=1}^{n}\alpha_i y_i x_{ij} + sign(\beta_j)\eta = 0 \quad (\nabla_{\beta_{\mathscr{V}_\beta}} L = 0) \quad (a) \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \qquad (\nabla_{\beta_0} L = 0) \quad (b) \\ \forall i \in [1,n]\ , & 1 - \alpha_i - \mu_i = 0 \qquad (\nabla_{\varepsilon} L = 0) \quad (c). \end{cases} \tag{2.20}$$

Les conditions de complémentarité KKT associées à (a), (b) et (c) de (2.15) s'écrivent de la façon suivante :

$$\begin{cases} (a)\ \forall i \in [1,n]\ , & \alpha_i(r_i - \varepsilon_i) = 0 \\ (b)\ \forall i \in [1,n]\ , & \mu_i \varepsilon_i = 0 \\ (c)\ \forall \eta \neq 0\ , & \eta(\|\beta\|_1 - s) = 0. \end{cases} \tag{2.21}$$

Tant que la contrainte (c) de (2.15) est active, $\eta \neq 0$ et la condition (2.21.c) implique :

$$\|\beta\|_1 = s. \tag{2.22}$$

Nous avons aussi par définition de l'ensemble $\mathscr{E}$ les équations suivantes :

$$1 - (\beta_0 + \sum_{j=1}^{p} \beta x_{ij}) = 0, \quad \forall i \in \mathscr{E}. \tag{2.23}$$

Les équations (a) et (b) de (2.20), de (2.22) et de (2.23), forment le système linéaire S suivant :

$$(S) \begin{cases} \forall j \in \mathscr{V}_\beta\ , \lambda_2 \beta - \sum_{i=1}^{n} \alpha_i y_i x_{ij} + sign(\beta_j)\eta & = 0 \\ \sum_{i=1}^{n} \alpha_i y_i & = 0 \\ \forall i \in \mathscr{E},\ 1 - (\beta_0 + \sum_{j=1}^{p} \beta x_{ij}) & = 0 \\ \|\beta\|_1 & = s. \end{cases}$$

Le système fait intervenir $(|\mathscr{V}_\beta| + |\mathscr{E}| + 2)$ équations et $(|n| + |p| + 2)$ inconnues (β_0, β, α et η). Cependant il est possible de réduire le nombre d'inconnues globales, en constatant d'une part que seules les composantes de β appartenant à $\mathscr{V}_\beta$ sont inconnues (par définition $\forall j \in \mathscr{V}_0, \beta_j = 0$), et en utilisant d'autre part les conditions KKT et l'annulation du lagrangien, afin de réduire le nombre de composantes inconnues de α.

- Premièrement remarquons que l'équation (2.20.c) impose : $\forall i \in [1,n]\ , 0 \leq \alpha_i \leq 1$ (μ est par définition positif).
- Soit i_0 tel que $\alpha_{i_0} \neq 0$, l'équation (2.21.a) entraîne : $r_{i_0} = \varepsilon_{i_0}$. Si de plus $i_0 \in \mathscr{L}$, $\varepsilon_{i_0} > 0$ implique $\mu_{i_0} = 0$ (d'après l'équation (2.21.b)) et donc $\alpha_{i_0} = 1$ (d'après (2.20.c)). Ainsi $\forall i \in \mathscr{L} \Rightarrow \alpha_i = 1$.
- Soit $i_0 \in \mathscr{R}$, la présence de $\varepsilon_{i_0} \geq 0$ dans la fonction objectif et la contrainte (b) de (2.15) implique $\varepsilon_{i_0} = 0$. L'équation (2.21.a) entraîne alors $\alpha_{i_0} = 0$.

Ainsi, seules les composantes de α appartenant à $\mathscr{E}$ sont inconnues et le système (S) fait intervenir $(|\mathscr{E}| + |\mathscr{V}_\beta| + 2)$ inconnues. La forme de (S) indique une évolution linéaire des paramètres β, β_0, α et η par rapport à s à condition que le système ne change pas. C'est-à-dire tant que les ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$ restent identiques. Ainsi pour prédire l'évolution des paramètres il suffit de calculer leur dérivée par rapport à s en inversant le système dérivé (S') déduit directement du système (S) :

$$\begin{cases} \forall j \in \mathscr{V}_\beta \ , \lambda_2 \dfrac{\Delta\beta_j}{\Delta s} - \displaystyle\sum_{i\in\mathscr{E}} \frac{\Delta\alpha_i}{\Delta s} y_i x_{ij} + sign(\beta_j)\frac{\Delta\eta}{\Delta s} & = 0 \\ \displaystyle\sum_{i\in\mathscr{E}} \frac{\Delta\alpha_i}{\Delta s} y_i & = 0 \\ \forall i \in \mathscr{E} \ , \dfrac{\Delta\beta_0}{\Delta s} + \displaystyle\sum_{j\in\mathscr{V}_\beta} \frac{\Delta\beta_j}{\Delta s} x_{ij} & = 0 \\ \displaystyle\sum_{j\in\mathscr{V}_\beta} sign(\beta_j)\frac{\Delta\beta_j}{\Delta s} & = 1. \end{cases}$$

Une fois que les dérivées des différents paramètres sont calculées, il reste à prédire les changements du système (S') qui correspondent à des ruptures de pente des paramètres. Cela est détaillé dans la section 2.2.4 qui traite du fonctionnement général de l'algorithme.

2.2.3 Initialisation

Un chemin peut se parcourir dans 2 sens différents : relaxation ou renforcement de la contrainte. Le parcours par relaxation est plus intéressant dans la mesure il commence d'une solution simple (interprétable) qui se complexifie au cours du chemin. La phase d'initialisation de l'algorithme ($s = 0$) distingue deux cas : les classes sont équilibrées ou les classes sont déséquilibrées. Selon le cas, l'analyse mathématique montre que la valeur de la fonction objective, les ensembles de points ($\mathscr{R}$, $\mathscr{E}$ et $\mathscr{L}$) et les ensembles de variables ($\mathscr{V}_\beta$ et $\mathscr{V}_0$) sont différents. Les différents cas d'initialisation sont détaillés dans le tableau 2.1. Dans le cas où les classes sont déséquilibrées, il est nécessaire d'utiliser un solveur externe afin d'amorcer la construction du chemin de régularisation.

2.2.4 Fonctionnement général

A chaque itération, on calcule les dérivées des paramètres β, β_0, α, et η par rapport à s par inversion du système (S'). De ces dernières, on déduit les dérivées de résidu r (équation (2.16)) et de la corrélation généralisée c (équation (2.17)). Ensuite il faut repérer les changements du système (S'), c'est à dire déterminer le pas δs minimum qui conduit à la violation

TABLE 2.1 : initialisation DRSVM, ensemble des cas possibles

	Ensemble des initialisations possibles					
	β_0	$r_i, i \in I^+$	$r_i, i \in I^-$	$\mathscr{E}$	$\mathscr{R}$	$\mathscr{L}$
$\lvert I^+\rvert = \lvert I^-\rvert$	1	0	2	I^+	$\emptyset$	I^-
	$]-1,1[$	$1-\beta_0 > 0$	$1-\beta_0 > 0$	$\emptyset$	$\emptyset$	$[1,n]$
	-1	2	0	I^-	$\emptyset$	I^+
$\lvert I^+\rvert < \lvert I^-\rvert$	1	2	0	I^-	$\emptyset$	I^+
$\lvert I^+\rvert > \lvert I^-\rvert$	1	0	2	I^+	$\emptyset$	I^-

d'une contrainte du système. En premier lieu, il est nécessaire de recenser les contraintes qui sont susceptibles d'être enfreintes. La violation d'une contrainte est appelée *événement*. Au total 5 événements différents peuvent se produire :

- un point sort de $\mathscr{E} \Leftrightarrow \alpha_i = 0$ (le point rentre dans $\mathscr{R}$) ou $\alpha_i = 1$ (le point rentre dans $\mathscr{L}$)
- un point rentre dans $\mathscr{E} \Leftrightarrow r_i = 0$
- une variable devient inactive $\Leftrightarrow \beta_j = 0$
- une variable devient active $\Leftrightarrow |c_j| = \eta$
- la contrainte sur $\|\beta\|_1$ devient inactive $\Leftrightarrow \eta = 0$ (c'est une condition d'arrêt de l'algorithme qui correspond à la fin du chemin).

Pour déterminer l'événement qui survient en premier, on calcule pour chaque contrainte le pas δs maximum par rapport auquel on peut faire évoluer la solution. L'expression des différents pas δs est décrite en (2.24) :

$$\begin{cases} \delta_s^1 = \min\limits_{i\in\mathscr{E}} \max\Big(\frac{1-\alpha_i}{\frac{\Delta\alpha_i}{\Delta s}}, \frac{0-\alpha_i}{\frac{\Delta\alpha_i}{\Delta s}}\Big) \\ \delta_s^2 = \min\limits_{i\in\mathscr{E}'} \frac{0-r_i}{\frac{\Delta r_i}{\Delta s}} \\ \delta_s^3 = \min\limits_{j\in\mathscr{V}'_\beta} \frac{0-\beta_j}{\frac{\Delta\beta_j}{\Delta s}} \\ \delta_s^4 = \min\limits_{j\in\mathscr{V}_0} \max\Big(\frac{-\eta-c_j}{-\frac{\Delta c_j}{\Delta s}+\frac{\Delta\eta}{\Delta s}}, \frac{\eta-c_j}{-\frac{\Delta c_j}{\Delta s}+\frac{\Delta\eta}{\Delta s}}\Big) \\ \delta_s^5 = 0-\eta. \end{cases} \tag{2.24}$$

Le premier événement qui survient correspond à $\delta s = \min(\delta_s^1, \delta_s^2, \delta_s^3, \delta_s^4, \delta_s^5)$ et à l'événement associé détecté (voir voir 2.2), on met à jour les ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$, ainsi que

TABLE 2.2 : Liste des différents événements avec leur condition d'apparition et le pas δs associé.

événement	condition	pas
$\mathscr{E} \to \mathscr{R}$	$\alpha_i = 0$	δs^1
$\mathscr{E} \to \mathscr{L}$	$\alpha_i = 1$	δs^1
$\mathscr{R} \to \mathscr{E}$	$r_i = 0$	δs^2
$\mathscr{L} \to \mathscr{E}$	$r_i = 0$	δs^2
$\mathscr{V}_\beta \to \mathscr{V}_0$	$\beta_j = 0$	δs^3
$\mathscr{V}_0 \to \mathscr{V}_\beta$	$\lvert c_j \rvert = \eta$	δs^4
Fin chemin	$\lambda_1 = 0$	δs^5

la valeur des différents paramètres :

$$\begin{cases} \beta_0 \leftarrow \beta_0 + \delta s \dfrac{\Delta \beta_0}{\Delta s} & \\ \beta_j \leftarrow \beta_j + \delta s \dfrac{\Delta \beta_j}{\Delta s}\,, & \forall j \in \mathscr{V}_\beta \\ \alpha_i \leftarrow \alpha_i + \delta s \dfrac{\Delta \alpha_i}{\Delta s}\,, & \forall i \in \mathscr{E} \\ \eta \leftarrow \eta + \delta s \dfrac{\Delta \eta}{\Delta s} & \\ r_i \leftarrow r_i + \delta s \dfrac{\Delta r_i}{\Delta s}\,, & \forall i \in \mathscr{L} \cup \mathscr{R} \\ c_j \leftarrow c_j + \delta s \dfrac{\Delta c_j}{\Delta s}\,, & \forall j \in \mathscr{V}_0 \end{cases}$$

A chaque itération k, on vérifie si la condition $\eta = 0$ ou si le nombre d'itérations maximum fixé est atteint. Dans l'affirmative, la procédure se termine et on renvoie l'état de la solution de l'itération courante : $f^k(\cdot) = \beta_0^k + \langle \beta^k, \cdot \rangle$. Dans le cas contraire, une nouvelle itération est effectuée jusqu'à ce qu'une des deux conditions soit atteinte.

2.2.5 Simulations et problèmes rencontrés

Une implémentation de l'algorithme DRSVM a été réalisée sous MATLAB. L'ensemble des solutions du chemin a été contrôlé à l'aide d'un solveur quadratique appelé CVX de Grant et al. [2008]. A chaque itération k, le couple $(\beta_0^{(k)}, \beta^{(k)})$ doit correspondre à l'optimum du problème (2.15) avec les paramètres $(\lambda_2, s^{(k)})$. Nous effectuons alors une résolution du problème avec les paramètres $(\beta_0^{(k)}, \beta^{(k)})$ à l'aide de CVX et comparons les différentes valeurs des paramètres du

système : β_0, β et α. Aussi nous avons, pour tester la validité de notre implémentation, reproduit les jeux de données utilisés par Wang. Le DRSVM étant un problème quadratique, il peut être résolu à l'aide des solveurs disponibles sur le marché. Un codage sur CPLEX [CPLEX, 2005] est en cours de réalisation.

2.2.5.1 Gestion des données discrètes

Nous avons appliqué l'algorithme de chemin, sur des données discrète et avons observé des problèmes d'inversion de matrice des dérivées. Plus précisément, avons avons rencontré des situations, où des points rentrent simultanément dans $\mathscr{E}$, ce qui entraîne une dépendance de certaines lignes de la matrice dérivée. Afin de résoudre ce problème, nous avons artificiellement ajouté aux données, un bruit gaussien d'une amplitude infinitésimale $k = 10^{-12} \max |x_{ij}|$. Ainsi nous avons empêché la sélection multiple d'événements conduisant aux problèmes d'inversion matricielle. Aussi nous avons contrôlé, à l'aide du solveur CVX appliqué sur les données non bruitées, que nous obtenons le même chemin et que la valeur des coefficients de la solution est identique à la précision numérique près.

2.2.5.2 Gestion de l'instabilité numérique

Nous avons rencontré des problèmes d'instabilité numérique au cours de l'implémentation de l'algorithme. Cela est dû principalement aux erreurs de précision pouvant survenir lors de l'inversion du système dérivé (S'). Ces erreurs, aussi faibles soient-elles, peuvent-être source de perturbation importante. En effet, certaines variables, à l'instar des coefficients de Lagrange α_i peuvent devenir négatifs ou supérieurs à 1. Cela a une influence directe sur la gestion des événements et peut conduire à un mauvais choix et faire dériver le chemin. Afin de remédier à ce problème, il est judicieux de réinitialiser automatiquement à chaque itération, les composantes des variables α, β r et c qui ont des valeurs théoriquement fixées. En procédant de la sorte, on s'assure que la gestion des événements se réalise correctement.

2.2.5.3 Gestion de la désactivation de variable

Nous avons remarqué, lors de la mise en œuvre du DRSVM, que dans le cas de désactivation d'une variable, l'algorithme ne convergeait pas. En effet, si on analyse l'équation suivante :

$$\lambda_2 \beta_j - \sum_{i=1}^{n} y_i \alpha_i x_{ij} + \text{sign}(\beta_j)\eta = 0 \ , \forall j \in \mathscr{V}_\beta . \tag{2.25}$$

Nous remarquons que lorsqu'une variable va être désactivée (la valeur de β_j tend vers zéro), la quantité $\sum_{i=1}^{n} y_i \alpha_i x_{ij}$ est très proche de η. Ainsi par continuité la valeur de la corrélation

généralisée c_j de la variable qui vient d'être désactivée est égale à η. Cela implique que, lors de l'étape suivante, l'algorithme va sélectionner la même variable et ne parvient pas à réaliser le processus de désélection de variable. Pour éviter ce phénomène il suffit, à chaque itération, de garder en mémoire la dernière variable désélectionnée, puis d'interdire de sélectionner à nouveau, lors de l'itération suivante une variable qui vient d'être désactivée. A l'étape suivante, le système ayant évolué, on a $c_j < \eta$ et on peut alors autoriser de nouveau sa sélection.

2.2.5.4 Problème de surdétermination de l'algorithme DRSVM

FIGURE 2.6 : Données unidimensionnelles constituées de 2 classes de 5 points chacune. Ce jeu de données est utilisé afin d'illustrer la possible surdétermination de la solution du DRSVM

Nous avons remarqué que sur certains exemples l'algorithme DRSVM aboutissait à la construction de chemins sur lesquels apparaissait une surdétermination de la solution (β_0,β). Cela se traduit par des problèmes d'inversion du système dérivé du DRSVM. Afin de comprendre les raisons d'apparition de ce phénomène, nous allons commencer par analyser un exemple qui illustre de façon simple, un cas de surdétermination de la solution. Ensuite dans la section suivante, nous reviendrons sur les raisons de ce problème et sur les façons de le résoudre.

Nous considérons une base de donnée $(x_i)_{1\leq i\leq 10}$ unidimensionnel constituée de classes équilibrées (voir figure 2.6). Nous allons observer l'évolution du chemin de régularisation sur les premières itérations. Nous fixons initialement le biais β_0 à zéro, qui est une solution du problème (voir le tableau 2.1).

- Itération 1 :
  Le premier événement est l'activation de l'unique variable du problème j_0.
- Itération 2 :
  Les seuls événements à considérer sont : (1) un point sort de $\mathscr{E}$, (2) un point rentre dans $\mathscr{E}$ et (3) $\eta = 0$. Les deux classes étant parfaitement séparées selon j_0, (3) ne peut pas directement survenir. D'autre part le choix initial de $\beta_0 = 0$ impose que tous les points sont dans $\mathscr{L}$ et donc (1) est impossible. Ainsi un point i_0 correspondant à $\text{argmax}_{1\leq i\leq 10}|x_i|$, transite dans $\mathscr{E}$.
- Itération 3 :
  L'équation $\sum_{i=1}^{n}\alpha_i y_i = 0$ implique qu'il est nécessaire d'avoir au moins deux points de classe opposée dans $\mathscr{E}$ afin que les coefficients α_i associés atteignent zéro et ainsi permettre aux points de transiter vers $\mathscr{R}$. Donc α_{i_0} ne peut pas atteindre seul la valeur zéro et (1) n'est pas possible. Ainsi le seul événement possible est à nouveau (2). Une analyse

de l'évolution des résidus associés aux points appartenant à la même classe que i_0, montre que leurs résidus augmentent. A l'inverse les points appartenant à la classe opposée voient leurs résidus diminuer et le point i_1 ayant la valeur absolue la plus élevée va transiter dans $\mathscr{E}$.

– Itération 4 :
A ce stade il est maintenant possible d'envisager (1) puisque deux points de classes opposées sont dans $\mathscr{E}$. Cependant l'analyse des cardinaux des ensembles ($|\mathscr{V}_\beta| = 1$ et $|\mathscr{E}|$) indique que la solution ne peut plus évoluer. En effet le classifieur $f : x \to f(x) = \beta_0 + \beta_{j_0} x$, a au total deux degrés de liberté (β_0 et β_{j_0}) qui peuvent varier sur le chemin. Mais ces derniers sont en fait complètement déterminés par les équations $r_{i_0} = 0$ et $r_{i_1} = 0$ (voir la figure 2.7). En outre, l'invariabilité de β_{j_0} entraîne la constance du paramètre s par rapport auquel le chemin est construit.

En conclusion, cet exemple illustre que pour certaines configurations de jeux de donnée, il est possible de rencontrer des cas où le chemin de régularisation en s se retrouve figé. Néanmoins s'il est possible de faire évoluer les coefficient α_{i_0} et α_{i_1} tout en maintenant s constant, on pourra rééquilibrer les cardinaux de $\mathscr{E}$ et de $\mathscr{V}_\beta$ et continuer la construction du chemin. Cette approche sera développée dans la section suivante.

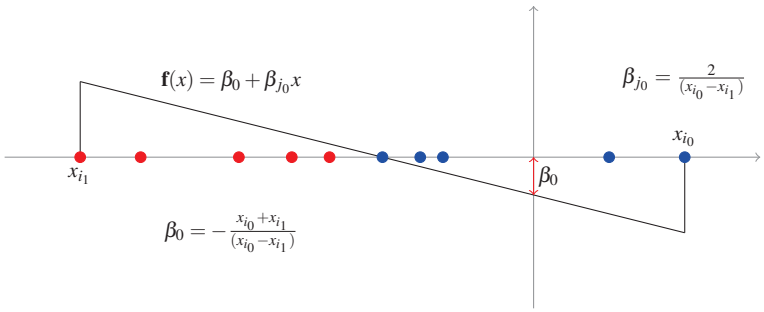


FIGURE 2.7 : Nous affichons la forme du classifieur f à l'itération 4 . A cette étape les points x_{i_0} et x_{i_1}, par le biais des équations sur leur résidu, déterminent complètement les paramètres β_0 et β_{j_0}. Cela peut aussi se remarquer sur la figure, où nous indiquons leur expression analytique en fonction de x_{i_0} et x_{i_1}. La solution ne peut plus alors évoluer sur le chemin de régularisation en s.

2.3 Proposition d'un chemin en λ_1 pour le DRSVM, via l'analyse de la sous-différentielle

L'exemple de la partie 2.2.5.4 révèle des cas de sur-dimensionnement de la solution pour certaines parties du chemin du DRSVM. Dans ces situations, l'analyse suggère qu'il faudrait pouvoir faire évoluer les ensembles $\mathscr{E}$, $\mathscr{R}$, $\mathscr{L}$,$\mathscr{V}_\beta$ et $\mathscr{V}_0$, tout en maintenant le paramètre s constant,

c'est à dire sans faire varier la solution (β_0, β). Le paramètre s est introduit à l'issu de la transformation d'un problème non-différentiable sans contrainte (2.2), en un problème différentiable avec les contraintes de (2.15). La formulation originelle du *DRSVM* quant à elle, fait intervenir le paramètre λ_1 qui, à l'instar de s, régule l'influence de la norme 1. On peut se demander s'il est possible de construire un chemin de régularisation directement par rapport à λ_1. Dans l'affirmative il faudrait étudier le comportement d'un tel chemin dans la situation problématique évoquée ci-dessus. Cependant le problème initial n'étant pas différentiable, il est nécessaire d'introduire au préalable un concept mathématique appelé *sous-différentielle*, afin de calculer les conditions d'optimalité du problème. Ce concept étend la notion de dérivation pour certaines classes de fonctions non différentiable et possède de riches propriétés. Notamment un théorème d'optimalité, intéressant pour dériver la construction d'un chemin de régularisation.

Ainsi nous allons étudier, via le prisme de la théorie de la sous-différentielle, le problème initial du DRSVM afin de déduire un chemin de régularisation par rapport au paramètre λ_1.

2.3.1 Théorie de la sous-différentielle

2.3.1.1 Introduction sous-différentielle

Dans le cadre de l'optimisation mathématique, on peut être amené à manipuler des fonctions non différentiables, c'est à dire possédant un ou plusieurs point(s) de singularité. La différentiation est une étape cruciale dans la mesure où elle fournit des informations sur l'optimum recherché. Il a donc été assez naturel de développer des concepts généralisant la différentielle et pouvant être utilisés afin de déterminer les solutions d'un problème d'optimisation. La figure 2.8 illustre l'intuition qui a mené à la définition standard de la sous-différentielle classique pour les fonctions convexes. La fonction f représentée à gauche est différentiable, $\nabla f(a)$ le gradient de f au point a se définit simplement comme la pente de la fonction tangente à f au point a. En revanche, ce n'est pas le cas pour la fonction g qui est singulière en a et qui donc possède une infinité de tangentes en a. Néanmoins les fonctions convexes différentiables ont la propriété d'être en tout point au-dessus de leurs tangentes. C'est en s'inspirant de cette propriété que l'on construit la notion de sous-gradient pour les fonctions convexes à valeurs réelles.

Définition 1 (sous-gradient)**.** *Un sous-gradient d'une fonction* $f : \mathbb{R}^p \to \mathbb{R}$ *en a, est un vecteur* $u \in \mathbb{R}^p$*, tel que :*

$$\forall z \in \mathbb{R}^p, f(z) \geq f(a) + u^T(z-a).$$

Géométriquement, un sous-gradient u de f en a, s'interprète comme le vecteur normal d'une droite d tangente à f en a qui est en-dessous de cette dernière. La sous-différentielle se définit alors comme l'ensemble des sous-gradients .

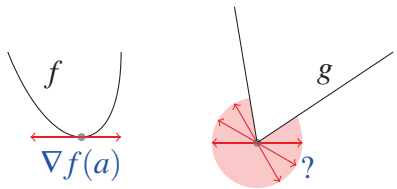


FIGURE 2.8 : La fonction f représentée sur la partie gauche est différentiable en a et possèdent une unique tangente en ce point. La pente de cette tangente s'interprète comme le gradient en a. A droite la fonction g est singulière en a, il existe une infinité de tangentes en ce point. On remarque néanmoins qu'il est possible de définir l'ensemble des tangentes inférieures à la fonction (représenté en rouge).

Définition 2 (sous-différentielle)**.** *La sous-différentielle $\partial f(a)$ d'une fonction $f : \mathbb{R}^p \to \mathbb{R}$ en a est l'ensemble :*

$$\partial f(a) = \{u \in \mathbb{R}^p, \forall z \in \mathbb{R}^p, f(z) \geq f(a) + u^T(z-a)\}.$$

Notons que l'appellation de *sous-différentielle* peut générer une confusion avec la notion de différentielle. En effet la différentielle d'une fonction est une forme linéaire tandis que la sous-différentielle est un ensemble de vecteurs. En revanche le sous-gradient est bien homogène au gradient et généralise ce dernier. Dans le cas où une fonction convexe f est différentiable en a, le gradient est le seul vecteur vérifiant la définition 1 et la sous-différentielle est alors réduite à l'ensemble singleton gradient.

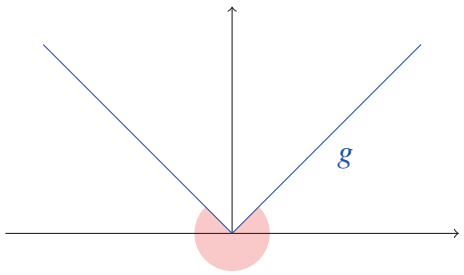
2.3.1.2 Exemples de sous-différentielles de fonctions usuelles non différentiables

A titre d'illustration nous déterminons la sous-différentielle des fonctions suivantes :

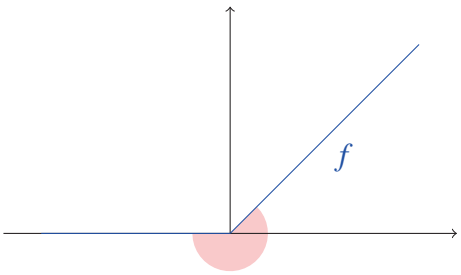
$$f : x \to \max(0,x) \text{ et } g : x \to |x|.$$

Notons que ces fonctions interviennent directement dans la fonction de coût du *DRSVM* et leurs sous-différentielles seront utilisées dans la section 2.3.2. Ces deux fonctions ne sont pas différentiables en zéro mais il est néanmoins possible d'expliciter leur sous-différentielles. Ces dernières sont calculées et représentées graphiquement sur la figure 2.9 . Les différentes contraintes issues de la définition de la sous-différentielle, définissent un domaine représenté en rouge, tout vecteur appartenant à ce domaine est un sous-gradient.

Mais que se passe-t-il pour les fonctions qui ont des dérivées infinies en certains points ? Étudions le cas de la fonction convexe $f : x \to -\sqrt{x}$ qui a une pente infinie en 0. Par définition un vecteur $v \in \partial f(0)$ si v vérifie la propriété suivante : $\forall z \in \mathbb{R}^+, -\sqrt{(z)} \geq 0 + vz$. Si $v \geq 0, \forall z > 0$ on a $-\sqrt{(z)} < vz$ et donc la propriété n'est pas respectée. Si $v < 0$, il est possible d'extraire un



$\partial g(0)$



$\partial f(0)$

$$
\begin{aligned}
& v \in \partial g(0) \\
\Leftrightarrow & \forall z \in \mathbb{R}, g(z) \geq 0 + v(z-0) \\
\Leftrightarrow & \begin{cases} \forall z \in \mathbb{R}^+ g(z) = z \geq vz \\ \forall z \in \mathbb{R}^- g(z) = -z \geq vz \end{cases} \\
\Leftrightarrow & \begin{cases} v \leq 1 \\ v \geq -1 \end{cases} \\
\Leftrightarrow & v \in [-1,1]
\end{aligned}
\qquad
\begin{aligned}
& v \in \partial f(0) \\
\Leftrightarrow & \forall z \in \mathbb{R}, f(z) \geq 0 + v(z-0) \\
\Leftrightarrow & \begin{cases} \forall z \in \mathbb{R}^+ f(z) = z \geq vz \\ \forall z \in \mathbb{R}^- f(z) = 0 \geq vz \end{cases} \\
\Leftrightarrow & \begin{cases} v \leq 1 \\ v \geq 0 \end{cases} \\
\Leftrightarrow & v \in [0,1]
\end{aligned}
$$

FIGURE 2.9 : Calcul et représentation des sous-différentielles en zéro des fonctions maximum $f(\cdot) = \max(0, \cdot)$ (droite) et valeur absolue $g(\cdot) = |\cdot|$ (gauche).

réel z pour lequel la propriété n'est pas respectée, en choisissant $z > \frac{1}{v^2}$. Ainsi il ne peut exister de sous-gradient et la sous-différentielle est alors réduite à l'ensemble vide.

2.3.1.3 Quelques propriétés générales de la sous-différentielle

Nous ne présentons pas ici une liste exhaustive des nombreuses propriétés de la sous-différentielle, mais introduisons les propriétés nécessaires et suffisantes afin de calculer les conditions d'optimalité du problème DRSVM (équation (2.2)). Nous commençons cette section par la présentation du théorème d'optimalité pour la sous-différentielle. Après détermination de la sous-différentielle globale du problème DRSVM, nous allons utiliser le théorème ci-dessous afin d'extraire les informations nécessaires à l'établissement d'un chemin de régularisation (Rockafellar and Wets [2009], page 422) :

Théorème 1 (Règle de Fermat généralisée pour les fonctions convexes)**.** *Soit $f : \mathbb{R}^p \to \mathbb{R}$ une fonction convexe, a^* un optimum de f si et seulement si :*

$$0 \in \partial f(a^*).$$

Cependant la fonction de coût du DRSVM est trop complexe pour être calculée directement, avant d'appliquer le théorème 1, il faut la scinder en termes plus simples dont on peut déterminer facilement la sous-différentielle. Ensuite nous pouvons appliquer le théorème de sommation suivante pour déduire la sous-différentielle globale (Hiriart-Urruty and Lemaréchal [1996], page 232) :

Théorème 2 (Théorème de sommation)**.** *Soit $f_i : \mathbb{R}^p \to \mathbb{R}, i \in [1,n]$ une famille de fonctions convexes alors on a l'égalité suivante :*

$$\partial(\sum_{i=1}^{n} f_i)(a) = \oplus_{i=1}^{n} \partial f_i(a).$$

Une partie de la forme du DRSVM fait intervenir des pré-compositions affines et peut être calculée par l'application du théorème suivant (page 232 Hiriart-Urruty and Lemaréchal [1996]) :

Théorème 3 (Théorème de pré-composition affine)**.** *Soit $f(\cdot) = g(A(\cdot))$ avec A une matrice de taille $n \times p$ et g une fonction convexe alors :*

$$\partial f(a) = \{w \in \mathbb{R}^p \text{ tel que } w = A^T v \text{ avec } v \in \partial g(Aa)\}.$$

Le DRSVM fait intervenir la fonction $f = \|\cdot\|_1$, qui peut se décomposer en une somme de fonctions dont les ensembles de définition sont des sous-espaces disjoints de l'ensemble de définition de f, cette dernière est dite séparable :

$$f(a) = \sum_{i=1}^{p} f_i(a_i) \quad \text{avec} \quad f_i(a_i) = |a_i|. \tag{2.26}$$

En ce cas la sous-différentielle peut-être évaluée à l'aide du théorème suivant (Rockafellar and Wets [2009], page 426) :

Théorème 4 (Théorème de sommation pour les fonctions séparables)**.** *Soit f une fonction convexe qui pour $\forall a \in \mathbb{R}^p$ se décompose de la manière suivante : $f(a) = \sum_{i=1}^{p} f_i(a_i)$ avec $a = (a_1, \cdots, a_p)$, $a_i \in \mathbb{R}^{p_i}$ et $f_i : \mathbb{R}^{p_i} \to \mathbb{R}$ alors :*

$$\partial f(a) = \prod_{i=1}^{p} \partial f_i(a_i).$$

Le symbole $\prod$ représente le produit cartésien pour les ensembles. Nous allons maintenant présenter une interprétation de la sous-différentielle en tant que variable afin de simplifier le calcul de la sous-différentielle du problème DRSVM.

2.3.1.4 Interprétation de la sous-différentielles en tant que variable

La sous-différentielle d'une fonction f au point a, est par définition un ensemble de vecteurs respectant des contraintes. Ces ensembles varient selon le point a de l'espace où la sous-différentielle est évaluée. Décrire globalement la sous-différentielle d'une fonction $f : \mathscr{X} \to \mathbb{R}$ revient à caractériser les différents ensembles $\partial f(a)$, $\forall a \in \mathscr{X}$. Si nous faisons l'hypothèse $\forall a \in \mathscr{X}, \partial f(a) \neq \{\emptyset\}$ alors il est possible de ré-interpréter la sous-différentielle comme une variable, assujettie à un certain nombre de contraintes dépendant de a.

Par exemple, la sous-différentielle associée à la valeur absolue au point a peut se décrire simplement par une unique variable réelle α. Si $a = 0$, α peut prendre n'importe quelle valeur sur l'intervalle $[-1,1]$ sinon $\alpha = sign(a)$. Le tableau 2.3 décrit complètement les sous-différentielles des fonctions valeur absolue et fonction maximum. Pour des raisons de commodité, nous allons utiliser cette interprétation de la sous-différentielle comme variable lors du calcul des conditions d'optimalité du DRSVM. Cette interprétation est licite dans le cas du DRSVM car ce dernier ne fait intervenir que des termes convexes qui, en aucun point de $\mathbb{R}^p$, ne possèdent de dérivée infinie.

TABLE 2.3 : Valeurs et intervalles des sous-différentielles des fonctions $f = |\cdot|$ et $g = \max(0,\cdot)$.

a	$\partial_a f$	$\partial_a g$
< 0	$\{-1\}$	$\{0\}$
> 0	$\{1\}$	$\{1\}$
$= 0$	$[-1,1]$	$[0,1]$

2.3.2 DRSVM et formalisation du chemin en λ_1

Dans cette section la construction d'un chemin alternatif L_1 du DRSVM par le biais du concept de la sous-différentielle est abordée. Nous rappelons la forme initiale du problème DRSVM :

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} \max[0, 1 - y_i \beta(x_i)] + \frac{\lambda_2}{2} \|\beta\|_2 + \lambda_1 \|\beta\|_1 .$$

Pour des raisons de commodité nous reformulons le problème en remplaçant β_0 et β par une variable unique $\delta = (\beta, \beta_0)$. La sous-différentielle sera calculée par rapport à δ. Calculer la sous-différentielle du problème *DRSVM* consiste à trouver tous les vecteurs respectant la définition (1) de J. Nous décomposons cette dernière en différents termes afin de simplifier les

calculs :

$$\begin{cases} \min\limits_{\delta} \quad J(\delta) = J_1(\delta) + J_2(\delta) + J_3(\delta) \\ J_1(\delta) = \sum\limits_{i=1}^{n} J_1^i(\delta) \quad \text{avec } J_1^i(\delta) = \max[0, 1 + A_i\delta] \\ J_2(\delta) = \delta^T B \delta \\ J_3(\delta) = \|C\delta\|_1 \\ A_i = -y_i \begin{pmatrix} 1 & x_i^T \end{pmatrix}, B = \lambda_2 \begin{pmatrix} I_p & 0_{1,p} \\ 0_{p,1} & 0 \end{pmatrix} \text{ et } C = \lambda_1 \begin{pmatrix} I_p & 0_{1,p} \\ 0_{p,1} & 0 \end{pmatrix}. \end{cases} \tag{2.27}$$

Les termes J_1, J_2 et J_3 du problème 2.27, correspondent respectivement à la fonction hinge, à la norme 2 et la norme 1. Notons que si l'on considère le résidu du point x_i comme étant une fonction de δ, on a $r_i(\delta) = 1 + A_i\delta$. La fonction J_1^i n'est pas différentiable. Cependant elle peut se décrire comme la composition de la fonction affine r_i par la fonction hinge h. La sous-différentielle de J_1^i peut-être déterminée par l'application du théorème de pré-composition affine (théorème 3) :

$$\begin{aligned} v \in \partial J_1^i(\delta) \Leftrightarrow v = A_i^T \alpha_i \qquad & \text{avec } \alpha_i \in \partial h(r_i(\delta)) \\ \Leftrightarrow v = \begin{pmatrix} -y_i x_i \alpha_i \\ -y_i \alpha_i \end{pmatrix} \qquad & \text{avec } \alpha_i \text{ tel que } : \begin{cases} \alpha_i = 0 & \text{si } r_i(\delta) < 0 \\ \alpha_i = 1 & \text{si } r_i(\delta) > 0 \\ \alpha_i \in [0,1] & \text{si } r_i(\delta) = 0. \end{cases} \end{aligned}$$

Ensuite il suffit d'utiliser le théorème de sommation (théorème 2) afin de calculer $J_1 = \sum_{i=1}^{n} J_1^i$:

$$\begin{aligned} v \in \partial J_1(\delta) \Leftrightarrow v = \sum_{i=1}^{n} v_i \qquad & \text{avec } \forall i \in [1,n], v_i \in \partial J_1^i(\delta) \\ \Leftrightarrow v = \begin{pmatrix} -\sum_{i=1}^{n} y_i x_i \alpha_i \\ -\sum_{i=1}^{n} y_i \alpha_i \end{pmatrix} \quad & \text{avec } \alpha_i \text{ tel que } : \begin{cases} \alpha_i = 0 & \text{si } r_i(\delta) < 0 \\ \alpha_i = 1 & \text{si } r_i(\delta) > 0 \\ \alpha_i \in [0,1] & \text{si } r_i(\delta) = 0. \end{cases} \end{aligned}$$

La fonction J_2 est différentiable, sa sous-différentielle se réduit au singleton gradient :

$$\begin{aligned} v \in \partial J_2(\delta) \quad &\Leftrightarrow \quad v = \lambda_2 B \delta \\ &\Leftrightarrow \quad v = \lambda_2 \begin{pmatrix} \beta \\ 0 \end{pmatrix}. \end{aligned}$$

La fonction J_3 est la composée d'une fonction affine par la norme 1. Cette dernière peut se

décomposer en $f(a) = \sum_{j=1}^{p} f_j(a_j)$ avec $f_j(a_j) = |a_j|$, nous pouvons alors expliciter sa sous-différentielle à l'aide du théorème de sommation pour les fonctions séparables (théorème 4) :

$$\begin{aligned} v \in \partial f(a) \quad &\Leftrightarrow \quad v = \prod_{j=1}^{p} \partial f_j(a_j) \\ &\Leftrightarrow \quad v = (\Gamma_j)_{1\leq j\leq p} \qquad \text{avec } \Gamma_j \text{ tel que } : \begin{cases} \Gamma_j = sign(a_j) & \text{si } a_j \neq 0 \\ \Gamma_j \in [-1,1] & \text{si } a_j = 0. \end{cases} \end{aligned}$$

Une fois que l'on a calculé $\partial f(a)$, il suffit d'utiliser le théorème de pré-composition affine (théorème 3) pour déterminer la sous-différentielle de J_3 :

$$\begin{aligned} v \in \partial J_3^i(\delta) \quad &\Leftrightarrow \quad v = C^T\Gamma \qquad \text{avec } \Gamma \in \partial f(C(\delta)) \text{ tel que } \Gamma = (\gamma, \Gamma_{p+1})^T \\ &\Leftrightarrow \quad v = \lambda_1 \begin{pmatrix} \gamma \\ 0 \end{pmatrix} \qquad \text{avec } \gamma_j \text{ tel que } : \begin{cases} \gamma_j = sign(\beta_j) & \text{si } \beta_j \neq 0 \\ \gamma_j \in [-1,1] & \text{si } \beta_j = 0. \end{cases} \end{aligned}$$

Nous pouvons à ce stade réintroduire les ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$. Ces derniers peuvent être réinterprétés comme des domaines associés aux sous-différentielles α et γ :

TABLE 2.1 : Définition des ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$

$\mathscr{R}$	$\{i \in [1,n],\ r_i(\delta) < 0\}$	$i \in [1,n],\ \alpha_i = 0$
$\mathscr{E}$	$\{i \in [1,n],\ r_i(\delta) = 0\}$	$i \in [1,n],\ \alpha_i \in [0,1]$
$\mathscr{L}$	$\{i \in [1,n],\ r_i(\delta) > 0\}$	$i \in [1,n],\ \alpha_i = 1$
$\mathscr{V}_0$	$\{j \in [1,p],\ \beta_j = 0\}$	$j \in [1,p],\ \gamma_j \in [-1,1]$
$\mathscr{V}_\beta$	$\{j \in [1,p],\ \beta_j \neq 0\}$	$j \in [1,p],\ \gamma_j = sign(\beta_j)$

Enfin la sous-différentielle de J se déduit par l'application du théorème de sommation (théorème 2) :

$$v \in \partial J(\delta) \quad \Leftrightarrow \quad v = \begin{pmatrix} -\sum_{i=1}^{n} y_i x_i \alpha_i + \lambda_2 \beta + \lambda_1 \gamma \\ -\sum_{i=1}^{n} y_i \alpha_i \end{pmatrix},$$

$$\text{avec } \alpha_i \text{ tel que } : \begin{cases} \alpha_i = 0 & \text{si } r_i(\delta) < 0 \\ \alpha_i = 1 & \text{si } r_i(\delta) > 0 \\ \alpha_i \in [0,1] & \text{si } r_i(\delta) = 0, \end{cases}$$

$$\text{et } \gamma_j \text{ tel que } : \begin{cases} \gamma_j = sign(\beta_j) & \text{si } \beta_j \neq 0 \\ \gamma_j \in [-1,1] & \text{si } \beta_j = 0. \end{cases}$$

Une fois que nous avons défini les sous-diférentielles (α, γ) et déterminé la sous-différentielle globale du problème DRSVM (équation (2.2)), nous pouvons appliquer le théorème 1 afin de calculer les équations d'optimalité :

$$\begin{cases} \sum_{i=1}^{n} y_i x_i \alpha_i + \lambda_2 \beta + \lambda_1 \gamma = 0 \quad (a) \\ -\sum_{i=1}^{n} y_i \alpha_i = 0 \quad (b). \end{cases} \tag{2.28}$$

Nous remarquons qu'il est possible de découpler les variables γ et β. En effet si $j \in \mathscr{V}_\beta$ alors $\gamma_j = sign(\beta_j)$ et (2.28.a) ne fait alors intervenir plus que (α,λ_1,β). Si $j \in \mathscr{V}_0$ alors $\beta_j = 0$ et (2.28.a) ne fait alors intervenir plus que (α,λ_1,γ). La combinaison des équations (2.28.a) associées à $\mathscr{V}_\beta$, de l'équation (2.28.b) et des équations sur le résidu associé à $\mathscr{E}$, permet de former le système suivant :

$$(S^2) \begin{cases} \forall j \in \mathscr{V}_\beta, & -\sum_{i=1}^{n} y_i x_{ij} \alpha_i + \lambda_2 \beta_j & = -sign(\beta_j)\lambda_1 \\ & -\sum_{i=1}^{n} y_i \alpha_i & = 0 \\ \forall i \in \mathscr{E}, & y_i \sum_{j=1}^{p} x_{ij} \beta_j + y_i \beta_0 & = 1. \end{cases} \tag{2.29}$$

Ce système fait intervenir les inconnues β_0, β, α. La tableau 2.1 nous indique que si nous connaissons les ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$, seules $|\mathscr{E}|$ composantes de α et $|\mathscr{V}_\beta|$ composantes de β sont inconnues. De là découle que le système (S^2) se compose de $(|\mathscr{E}| + |\mathscr{V}_\beta| + 1)$ équations pour le même nombre d'inconnues. Les paramètres β, β_0 et α dépendent directement de la valeur de λ_1 tant que les ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$ restent inchangés. Afin de construire un chemin de régularisation, il faut tout d'abord s'assurer de son existence via l'analyse du système (S^2). Pour cela nous introduisons préalablement les ensembles $\mathscr{V}_0^* = \mathscr{V}_0 \cap \{j, -1 < \gamma_j < 1\}$ et $\mathscr{E}^* = \mathscr{E} \cap \{i, 0 < \alpha_i < 1\}$ afin d'étudier l'évolution des paramètres α, β, et β_0 sur des voisinages de λ_1 où il est assuré que le système (S^2) reste constant. Ainsi sous l'hypothèse $\mathscr{V}_0^* = \emptyset$ et $\mathscr{E}^* = \emptyset$, il existe un réel $\delta\lambda_1 < 0$ tel que les ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$ ne changent pas, c'est à dire tel que la structure de (S^2) est préservée. Les variables de ce système étant différentiables par rapport à λ_1, il est possible de dériver le système suivant :

$$(S^3) \begin{cases} \forall j \in \mathscr{V}_\beta, & -\sum_{i \in \mathscr{E}} y_i x_{ij} \frac{\Delta\alpha_i}{\Delta\lambda_1} + \lambda_2 \frac{\Delta\beta_j}{\Delta\lambda_1} & = -sign(\beta_j) \\ & -\sum_{i \in \mathscr{E}} y_i \frac{\Delta\alpha_i}{\Delta\lambda_1} & = 0 \\ \forall i \in \mathscr{E}, & \sum_{j \in \mathscr{V}_\beta} x_{ij} \frac{\Delta\beta_j}{\Delta\lambda_1} + \frac{\Delta\beta_0}{\Delta\lambda_1} & = 0. \end{cases} \tag{2.30}$$

Nous remarquons que ce système ne fait plus intervenir la variable λ_1 et donc la dépendance des paramètres α, β et β_0 par rapport à λ_1 est linéaire. Pour une valeur de $\delta\lambda_1$ tel que $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$ demeurent inchangés, l'évolution des paramètres peut-être déterminée à partir de l'inversion du système (S^3) :

$$\begin{pmatrix} \alpha(\lambda_1+\delta_{\lambda_1}) \\ \beta(\lambda_1+\delta_{\lambda_1}) \\ \beta_0(\lambda_1+\delta_{\lambda_1}) \end{pmatrix} = \begin{pmatrix} \alpha(\lambda_1) \\ \beta(\lambda_1) \\ \beta_0(\lambda_1) \end{pmatrix} + \delta_{\lambda_1} W^{-1} \begin{pmatrix} -sign(\beta_{\mathscr{V}_\beta}) \\ 0 \\ 0 \end{pmatrix},$$

$$\text{avec} \quad W = \begin{pmatrix} -x_{\mathscr{E},\mathscr{V}_\beta}^T \,.\, Dy_{\mathscr{E}} & \lambda_2 I_{\mathscr{V}_\beta} & 0_{\mathscr{V}_\beta,1} \\ -y_{\mathscr{E}}^T & 0_{1,\mathscr{V}_\beta} & 0 \\ 0_{\mathscr{E},\mathscr{E}} & x_{\mathscr{E},\mathscr{V}_\beta} & 1_{\mathscr{E},1} \end{pmatrix}.$$

$y_{\mathscr{E}}$ est un sous-vecteur de y sur l'ensemble $\mathscr{E}$, $D_{y_{\mathscr{E}}}$ est une matrice diagonale générée à partir de $y_{\mathscr{E}}$, $x_{\mathscr{E},\mathscr{V}_\beta}$ est une sous-matrice de x sur les ensembles $\mathscr{E}$ et $\mathscr{V}_\beta$ et $I_{\mathscr{V}_\beta}$ est la matrice identité de taille $|\mathscr{V}_\beta|$.

Maintenant supposons que $\mathscr{V}_0^*$ ou $\mathscr{E}^*$ soit non vide. Dans cette configuration, la commutation $\mathscr{E} \to \{\mathscr{R},\mathscr{L}\}$ ou $\mathscr{V}_0 \to \mathscr{V}_\beta$, est immédiate et le système (S^2) est instantanément modifié quelque soit la valeur de $\delta\lambda_1$. Cette commutation modifie également (S^3), ce qui conduit à une modification des pentes des paramètres α, β et β_0 par rapport à λ_1. Cependant cette rupture des pentes des paramètres n'engendre pas de discontinuité pour ces derniers car elle se produit de façon immédiate.

Ainsi la prise en compte de l'évolution globale des ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$ ainsi que l'évolution linéaire des paramètres α, β et β_0 sur des voisinages de λ_1 où (S^2) demeure inchangé, permet d'affirmer la linéarité par morceaux de la solution par rapport à λ_1 et d'aboutir à la construction d'un chemin de régularisation.

2.3.3 Initialisation

L'ensemble des cas d'initialisation est synthétisé dans le tableau 2.2.

2.3.3.1 Cas 1 : les classes sont équilibrées

Initialement le paramètre de régularisation λ_1 étant infini, les conditions d'optimalité (2.28.a) impliquent $\gamma = 0$ et donc d'après le tableau 2.1 nous avons $\beta = 0$. La condition d'optimalité (2.28.b) est l'unique contrainte que doit respecter la solution $f(\cdot) = \beta_0$. Supposons que β_0 appartient à $]1,+\infty[$, les points de la classe « plus » ont un résidu $r_i(\delta) = 1-\beta_0 < 0$ et $\alpha_i = 0$

TABLE 2.2 : initialisationDRSVM

	Ensemble des initialisations possibles								
	γ	$\alpha_i, i \in I^+$	$\alpha_i, i \in I^-$	β_0	$r_i, i \in I^+$	$r_i, i \in I^-$	$\mathscr{E}$	$\mathscr{R}$	$\mathscr{L}$
$I^+ = I^-$	0	1	1	1	0	2	I^+	$\emptyset$	I^-
	0	1	1	$]-1,1[$	$1-\beta_0 > 0$	$1-\beta_0 > 0$	$\emptyset$	$\emptyset$	$[1,n]$
	0	1	1	-1	2	0	I^-	$\emptyset$	I^+
$I^+ < I^-$	0	1	$[0,1]$	1	2	0	I^-	$\emptyset$	I^+
$I^+ > I^-$	0	$[0,1]$	1	1	0	2	I^+	$\emptyset$	I^-

alors que les points de la classe « moins » ont un résidu $r_i(\delta) = 1+\beta_0 > 0$ et $\alpha_i = 1$. Dans ce cas la contrainte (2.28.b) n'est pas respectée et l'hypothèse est invalidée. De manière symétrique nous pouvons montrer que le choix $\beta_0 \in]1,+\infty[$ contredit la condition d'optimalité. Supposons maintenant que β_0 appartient à $]-1,1[$, tous les points ont un résidu $r_i(\delta) = 1-y\beta_0 > 0$ et $\alpha_i = 1$. Supposons que $\beta_0 = 1$ alors les points de la classe « moins » ont un résidu $r_i(\delta) = 2 > 0$ et $\alpha_i = 1$ alors que les points de la classe « plus » ont un résidu $r_i(\delta) = 0$, les classes étant équilibrées la condition d'optimalité (2.28.b) implique que $\alpha_i = 1$. Supposons que $\beta_0 = -1$ alors les points de la classe « plus » ont un résidu $r_i(\delta) = 2 > 0$ et $\alpha_i = 1$ alors que les points de la classe « moins » ont un résidu $r_i(\delta) = 0$. Les classes étant équilibrées la condition d'optimalité (2.28.b) implique que $\alpha_i = 1$. Ensuite, nous faisons varier la valeur de λ_1 jusqu'à ce qu'une contrainte soit violée. Indépendamment de l'initialisation choisie, la valeur de la sous-différentielle $\alpha = 1_{n,1}$ ne peut évoluer à cause de (2.28.b). Les seules contraintes qui peuvent être enfreintes portent sur la sous-différentielle γ. Initialement $\gamma = 0$, puis au fur et à mesure que l'on fait diminuer λ_1 la valeurs des composantes de γ augmentent jusqu'à ce qu'une première composante atteigne 1 en valeur absolue. Cela correspond à l'activation d'une variable j_0 qui est sélectionnée comme suit :

$$\lambda_1^0 = \max_{j \in \mathscr{V}_0} |\sum_{i=1}^{n} y_i \alpha_i x_{ij}| \quad \text{et} \quad j_0 = \arg\max_{j \in \mathscr{V}_0} |\sum_{i=1}^{n} y_i \alpha_i x_{ij}|. \tag{2.31}$$

Afin d'illustrer l'influence du paramètre λ_2 sur le forme du chemin, nous détaillons le cas où $\beta_0 = 0$ (une analyse similaire des autres types d'initialisation possibles conduit aux même conclusions). Dans cette situation l'ensemble $\mathscr{E} = \emptyset$ et donc la valeur de β_0 est libre. Il y a deux contraintes susceptibles d'être enfreintes : un résidu $r_i = 0$ ou une composante de γ atteint 1 en

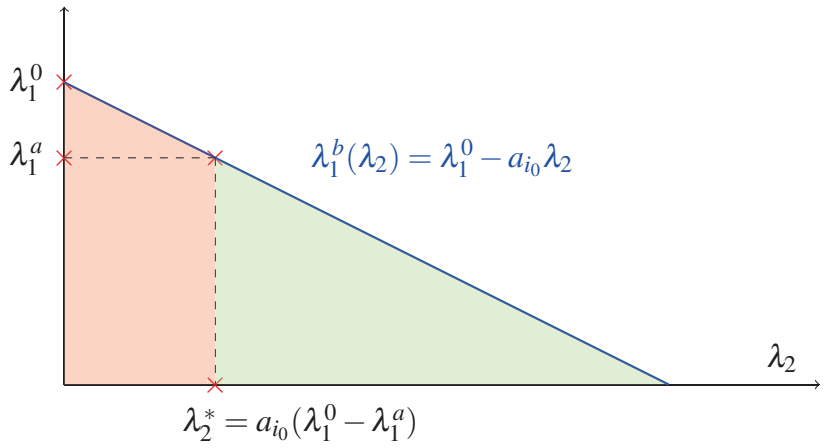


FIGURE 2.10 : La figure illustre la dépendance entre le choix du paramètre λ_2 et la forme du chemin à la deuxième itération. A cette étape, deux contraintes sont enfreintes pour une valeur de λ_1 égale à λ_1^a soit λ_1^b . L'expression de λ_1^b fait intervenir directement λ_2 et il est alors possible de définir 2 plages de valeurs de λ_2 pour lesquelles on a soit $\lambda_1^a > \lambda_1^b$ ou $\lambda_1^b < \lambda_1^a$. Ces 2 plages sont séparées pour la valeur λ_2^* qui correspond au cas limite où les contraintes sont enfreintes de façon simultanée.

valeur absolue. Cela correspond à deux valeurs $\lambda_1^1 = \max(\lambda_1^a, \lambda_1^b)$ avec λ_1^a et λ_1^b tel que :

$$\lambda_1^a = \max_{j \in \mathscr{V}_0 - \{j_0\}} |\sum_{i=1}^{n} y_i \alpha_i x_{ij}|$$
$$\lambda_1^b = \max_{i \in \mathscr{E}^*} \left(\lambda_1^0 - \lambda_2 a_i\right)$$
$$\text{avec} \quad a_i = \frac{1}{sign(\beta_{j_0}) y_{i_0} x_{i_0 j_0}} \quad \text{et} \quad \mathscr{E}^* = \{i \in [1,n], sign(\beta_{j_0}) y_{i_0} x_{i_0 j_0} > 0\}.$$

L'ensemble $\mathscr{E}^*$ est un sous-ensemble de $\mathscr{E}$, il représente les points susceptibles d'enfreindre la contrainte sur le résidu. Le choix $\lambda_1^1 = \lambda_1^a$ correspond à l'activation d'une nouvelle variable j_1 et $\lambda_1^1 = \lambda_1^b$ correspond à un point qui atteint $\mathscr{E}$. On peut remarquer que λ_1^b fait intervenir explicitement le paramètre λ_2. En d'autres termes, ce dernier conditionne le choix des événements et donc par conséquent la forme globale du chemin. Notons i_0 le point qui le premier, viole la contrainte sur le résidu. L'expression de λ_1^b montre que le choix du point i_0 est indépendant de la valeur de λ_2. Il est possible alors de donner une valeur analytique de λ_2^* pour laquelle les deux contraintes sont enfreintes de façon simultanée (voir la figure 2.10). A partir de la valeur λ_2^* on définit deux plages $]0, \lambda_2^*[$ et $]\lambda_2^*, +\infty[$ pour lesquelles on a respectivement $\lambda_1^b > \lambda_1^a$ et $\lambda_1^b < \lambda_1^a$. Ainsi choisir une valeur faible du paramètre telle que $\lambda_2 < \lambda_2^*$ conduit à faire faire transiter un point vers $\mathscr{E}$ plutôt que d'activer une nouvelle variable. Cela est cohérent avec le comportement du DRSVM, car plus le ratio (λ_2 / λ_1) est faible, plus le classifieur est parcimonieux.

2.3.3.2 cas 2 : les classes sont déséquilibrées

A l'instar du cas équilibré quand λ_1 est infini, on a $\gamma = 0$, $\beta = 0$ et seule la condition d'optimalité (2.28.b) doit être respectée. Pour les mêmes raisons invoquées pour le cas équilibré, on démontre que l'hypothèse selon laquelle β_0 appartient à l'intervalle $]-\infty,-1[\cup]1,+\infty[$ conduit à une contradiction.

Pour détailler le calcul, posons l'hypothèse que la classe plus est prépondérante. Il faut considérer 3 cas selon la valeur définie initialement pour β_0.

Supposons $\beta_0 \in]-1,1[$ alors les points de la classe « plus » et « moins » ont tous un résidu $r_i(\delta) = 1 \pm \beta_0 > 0$ et $\alpha_i = 1$. Ainsi on a $\sum_{i\in I^+} \alpha_i = |I^+|$ et $\sum_{i\in I^-} \alpha_i = |I^-|$. Par hypothèse $|I^+| > |I^-|$ et la condition (2.28.b) est invalidée.

Supposons $\beta_0 = -1$ alors les points de la classe « plus » ont un résidu $r_i(\delta) = 2 > 0$ et $\alpha_i = 1$ et les points de la classe « moins » ont un résidu $r_i(\delta) = 0$ et $\alpha_i \in [0,1]$. Ainsi on a $\sum_{i\in I^+} \alpha_i = |I^+|$ et $\sum_{i\in I^-} \alpha_i \leq |I^-|$. Par hypothèse $|I^+| > |I^-|$ et la condition (2.28.b) est invalidée.

Supposons $\beta_0 = 1$ alors les points de la classe « plus » ont un résidu $r_i(\delta) = 0$ et $\alpha_i = [0,1]$ et les points de la classe « moins » ont un résidu $r_i(\delta) = 2 > 0$ et $\alpha_i = 1$. La seule condition à respecter pour l'initialisation est la suivante :

$$\sum_{i\in I^+} \alpha_i = |I^-|.$$

Dans l'hypothèse que la classe « moins » est prépondérante, le calcul est symétrique, seule le cas $\beta_0 = -1$ est valide et aboutit à la condition :

$$\sum_{i\in I^-} \alpha_i = |I^+|.$$

La suite de l'initialisation est plus complexe dans la mesure où dans le cas général elle n'est pas unique. De plus le choix de l'initialisation peut conduire à des construction différentes de la première partie du chemin de régularisation. En effet le choix de α_{I^-} conditionne la valeur de la sous-différentielle γ. Tant que le paramètre λ_1 est infini on a $\gamma = 0$. Cependant, dès que ce dernier à une valeur finie, alors la valeur de γ dépend explicitement de la valeur de α_{I^-} et d'après (2.28.a) on a :

$$\gamma_j(\lambda_1, \alpha_{I^-}) = \frac{1}{\lambda_1}\Big(\sum_{i\in I^+} x_{ij} - \sum_{i\in I^-} \alpha_i x_{ij}\Big). \tag{2.32}$$

Choisir des initialisations différentes de α_{I^-} a pour conséquence de potentiellement modifier la composante j_0 de γ qui enfreint la première la contrainte $|\gamma_{j_0}| \leq 1$. Afin d'adopter une méthode automatique pour le choix de γ, nous nous proposons de choisir la valeur minimale de λ_1 pour laquelle on peut garantir les contraintes sur les sous-différentielles. Notons que cette valeur de λ_1 n'est pas nulle grâce à la présence d'un terme constant dans l'expression de γ (voir équation (2.32)). Pour chaque jeu de données cette valeur est unique et peut être déterminée par la résolution du problème linéaire suivant :

$$\begin{cases} \min\limits_{\lambda_1, \alpha_{I^-}} \lambda_1 \\ \sum\limits_{i \in I^-} \alpha_i = |I^+| \\ 0 \leq \alpha_i \leq 1 \ , \forall i \in I^+ \\ -1 \leq \gamma_j(\lambda_1, \alpha_{I^-}) \leq 1 \ , \forall j \in \mathscr{V}_0. \end{cases} \tag{2.33}$$

La résolution du problème (2.33) conduit à définir une valeur initiale λ_1^0 pour laquelle on débute la construction du chemin. Dans les jeux de données que nous avons manipulés, cette initialisation fournit une solution satisfaisante. Cependant nous avons constaté dans des cas où le déséquilibre est très important que la résolution du système (2.33) retournait une solution telle que : $\mathscr{E} > \mathscr{V}_\beta$, ce qui introduit une surdétermination de la solution. Dans cette situation, nous avons appliqué un solveur externe au problème DRSVM pour la valeur λ_1^0 afin d'initialiser la solution et de continuer la construction du chemin.

2.3.4 Fonctionnement de l'algorithme après la phase d'initialisation

Dans le mode de fonctionnement général de l'algorithme 1, on calcule à chaque itération le pas $|\delta\lambda_1|$ maximum pour lequel il est possible de faire évoluer la solution sans enfreindre une seule contrainte associée aux sous-différentielles α ou γ. On commence par résoudre le système linéaire (2.30) afin de déterminer les dérivées des différents paramètres. Ensuite, il faut envisager les différents cas d'infraction de contraintes qui correspondent à des modifications du système (S^2). Nous conservons la notion d'événement qui correspond à la violation d'une contrainte. Il y en tout 5 événements possibles, ils correspondent soit à la violation de l'un des 4 types de contraintes, soit à la relaxation totale de la régularisation L_1 :

1. une composante de la sous-différentielle α atteint une de ces 2 bornes admissibles, $\alpha_i = 0$ ou $\alpha_i = 1$
2. $r_i(\delta) = 0$ et la composante α_i associée devient libre dans l'intervalle $[0, 1]$
3. $\beta_j = 0$ et la composante γ_j associée devient libre dans l'intervalle $[-1, 1]$
4. une composante de la sous-différentielle γ atteint une de ces 2 bornes admissibles, $\gamma_j = -1$ ou $\gamma_j = 1$
5. la contrainte sur la norme 1 est complètement relâchée $\Rightarrow \lambda_1 = 0$ (condition d'arrêt de l'algorithme)

Data: $S_n = (x_i, y_i)_{1 \leq i \leq n}$, λ_2 et N_{max}
Result: chemin de régularisation : $(\beta_0^k, \beta^k, \lambda_1^k, \alpha^k, \gamma^k, r^k), k = 1, N^\star \leq N_{max}$
Initialize : β_0^0, β^0, α^0, γ^0, r^0, $\mathscr{R}^0$, $\mathscr{E}^0$, $\mathscr{L}^0$ et λ_1^0 ; (voir table 2.2)
$k \leftarrow 0$
while $\lambda_1^k \neq 0$ *et* $k < N_{max}$ **do**
  calcul des dérivées des variables (résoudre le système linéaire (2.30)) : $\frac{\Delta \alpha_{\mathscr{E}}}{\Delta \lambda_1}$, $\frac{\Delta \beta_{\mathscr{V}_\beta}}{\Delta \lambda_1}$, $\frac{\Delta \beta_0}{\Delta \lambda_1}$ et $\frac{\Delta r_{\mathscr{E}}}{\Delta \lambda_1}$;
  calcul des pas associés aux événements (2.35) ;
  calcul du point de rupture : $\delta_{\lambda_1} \leftarrow \max(\delta_{\lambda_1}^1, \delta_{\lambda_1}^2, \delta_{\lambda_1}^3, \delta_{\lambda_1}^4, \delta_{\lambda_1}^5)$;
  mise à jour variables (voir équation (2.36)) :
  mise à jour des ensembles (Table 2.3) :
  $\lambda_1^{k+1} \leftarrow \lambda_1^k + \delta_{\lambda_1}$;

  $k \leftarrow k + 1$;
end
$N^\star = k$;

Algorithm 1: chemin λ_1-DRSVM

TABLE 2.3 : Liste des différents événements avec leur condition d'apparition et le pas δs associé.

événement	condition	pas
$\mathscr{E} \to \mathscr{R}$	$\alpha_i = 0$	δs^1
$\mathscr{E} \to \mathscr{L}$	$\alpha_i = 1$	δs^1
$\mathscr{R} \to \mathscr{E}$	$r_i = 0$	δs^2
$\mathscr{L} \to \mathscr{E}$	$r_i = 0$	δs^2
$\mathscr{V}_\beta \to \mathscr{V}_0$	$\beta_j = 0$	δs^3
$\mathscr{V}_0 \to \mathscr{V}_\beta$	$\lvert\gamma_j\rvert = 1$	δs^4
Fin chemin	$\lambda_1 = 0$	δs^5

Ensuite il faut déterminer les différents pas $\delta\lambda_1$ associés aux violations des différentes contraintes. A titre d'exemple nous détaillons le calcul du pas correspondant à la violation d'une contrainte de la composante j de la sous-différentielle γ (activation d'une nouvelle va-

riable). Soit $\delta\lambda_1 < 0$ tel que $\gamma_j(\lambda_1) = 1$ alors la condition d'optimalité (2.28.a) implique :

$$\begin{aligned}
\gamma_j(\lambda_1 + \delta\lambda_1) &= \frac{\sum_{i=1}^{n} y_i x_{ij} \alpha_i(\lambda_1 + \delta\lambda_1)}{\lambda_1 + \delta\lambda_1} \\
\iff \quad 1 &= \frac{\sum_{i=1}^{n} y_i x_{ij} \alpha_i(\lambda_1) + \delta\lambda_1 \sum_{i\in\mathscr{E}} y_i x_{ij} \frac{\Delta\alpha_i}{\Delta\lambda_1}}{\lambda_1 + \delta\lambda_1} \\
\iff \quad \lambda_1 + \delta\lambda_1 &= \sum_{i=1}^{n} y_i x_{ij} \alpha_i(\lambda_1) + \delta\lambda_1 \sum_{i\in\mathscr{E}} y_i x_{ij} \frac{\Delta\alpha_i}{\Delta\lambda_1} \\
\iff \quad \delta\lambda_1 &= \frac{\sum_{i=1}^{n} y_i x_{ij} \alpha_i(\lambda_1) - \lambda_1}{-\sum_{i\in\mathscr{E}} y_i x_{ij} \frac{\Delta\alpha_i}{\Delta\lambda_1} + 1} \\
\iff \quad \delta\lambda_1 &= \frac{\lambda_1(\gamma_j - 1)}{-\sum_{i\in\mathscr{E}} y_i x_{ij} \frac{\Delta\alpha_i}{\Delta\lambda_1} + 1}.
\end{aligned} \tag{2.34}$$

De manière analogue, on peut déterminer les pas $\delta\lambda_1$ correspondant à la violation des différentes contraintes :

$$\begin{cases}
\delta^1_{\lambda_1} = \max\limits_{i\in\mathscr{E}} \min\Big(\frac{1-\alpha_i}{\frac{\Delta\alpha_i}{\Delta\lambda_1}}, \frac{0-\alpha_i}{\frac{\Delta\alpha_i}{\Delta\lambda_1}}\Big) \\
\delta^2_{\lambda_1} = \max\limits_{i\in\mathscr{E}'} \frac{0-r_i}{\frac{\Delta r_i}{\Delta\lambda_1}} \quad \text{avec} \quad \mathscr{E}' = \{i \notin \mathscr{E}, (0-r_i)/\frac{\Delta r_i}{\Delta\lambda_1} < 0\} \\
\delta^3_{\lambda_1} = \max\limits_{j\in\mathscr{V}'_\beta} \frac{0-\beta_j}{\frac{\Delta\beta_j}{\Delta\lambda_1}} \quad \text{avec} \quad \mathscr{V}'_\beta = \{i \in \mathscr{V}_0, (0-\beta_j)/\frac{\Delta\beta_j}{\Delta\lambda_1} < 0\} \\
\delta^4_{\lambda_1} = \max\limits_{j\in\mathscr{V}_0} \min\Big(\frac{\lambda_1(\gamma_j-1)}{-\sum_{i\in\mathscr{E}} \frac{\Delta\alpha_i}{\Delta\lambda_1} y_i x_{ij} + 1}, \frac{\lambda_1(\gamma_j+1)}{-\sum_{i\in\mathscr{E}} \frac{\Delta\alpha_i}{\Delta\lambda_1} y_i x_{ij} - 1}\Big) \\
\delta^5_{\lambda_1} = 0 - \lambda_1.
\end{cases} \tag{2.35}$$

Le premier événement qui survient correspond à $\delta^f_{\lambda_1}(\lambda_2) = \max(\delta^1_{\lambda_1}, \delta^2_{\lambda_1}, \delta^3_{\lambda_1}, \delta^4_{\lambda_1}, \delta^5_{\lambda_1})$ et à l'événement associé détecté 2.3, on met à jour les ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ et $\mathscr{V}_0$, ainsi que la valeur des différents paramètres :

$$\begin{cases}
\beta_0 \leftarrow \beta_0 + \delta\lambda_1 \frac{\Delta\beta_0}{\Delta\lambda_1} & \\
\beta_j \leftarrow \beta_j + \delta\lambda_1 \frac{\Delta\beta_j}{\Delta\lambda_1}\,, & \forall j \in \mathscr{V}_\beta \\
\alpha_i \leftarrow \alpha_i + \delta\lambda_1 \frac{\Delta\alpha_i}{\Delta\lambda_1}\,, & \forall i \in \mathscr{E} \\
\gamma_i \leftarrow \frac{\lambda_1}{\lambda_1+\delta\lambda_1}\gamma_i + \frac{\delta\lambda_1}{\lambda_1+\delta\lambda_1} \sum_{i\in\mathscr{E}} y_i x_{ij} \frac{\Delta\alpha_i}{\Delta\lambda_1}\,, & \forall i \in \mathscr{V}_0 \\
r_i \leftarrow r_i + \delta\lambda_1 \frac{\Delta r_i}{\Delta\lambda_1}\,, & \forall i \in \mathscr{L}\cup\mathscr{R} \\
\lambda_1 \leftarrow \lambda_1 + \delta\lambda_1. &
\end{cases} \tag{2.36}$$

A chaque itération k, on vérifie si la condition $\lambda_1 = 0$ ou si le nombre d'itérations maximum demandé est atteint. Dans l'affirmative la procédure se termine et on renvoie l'état de la solution à l'itération courante : $f^k(\cdot) = \beta_0^k + \langle \beta^k, \cdot \rangle$. Dans le cas contraire, une nouvelle itération est effectuée jusqu'à ce qu'une des deux conditions soit atteinte.

2.3.5 Influence du paramètre λ_2 sur l'évolution de α,γ, β_0 et β

Nous avons montré dans la section 2.3.3.1 qu'il est possible d'analyser explicitement l'influence de λ_2 sur la solution lors de l'initialisation. Il est aussi possible de faire ressortir l'action de λ_2 sur le chemin en reformulant le système S^3 par l'introduction de variables intermédiaires :

$$(S^4)\begin{cases} \forall j \in \mathscr{V}_\beta, & -\sum_{i\in\mathscr{E}} y_i x_{ij} \dfrac{\Delta A_i}{\Delta\lambda_1} & + \dfrac{\Delta B_j}{\Delta\lambda_1} & = -sign(B_j) \\ & -\sum_{i\in\mathscr{E}} y_i \dfrac{\Delta A_i}{\Delta\lambda_1} & & = 0 \\ \forall i \in \mathscr{E}, & & \sum_{j\in\mathscr{V}_\beta} x_{ij} \dfrac{\Delta B_j}{\Delta\lambda_1} + \dfrac{\Delta B_0}{\Delta\lambda_1} & = 0, \end{cases}$$

$$\text{avec} \quad A = \alpha, \quad B = \lambda_2\beta \quad \text{et} \quad B_0 = \lambda_2\beta_0.$$

On peut constater que le système (S^4) ne fait plus intervenir le paramètre λ_2. Il ne dépend que de la matrice de données x, de son vecteur label associé y et des ensembles $\mathscr{E}$ et $\mathscr{V}_\beta$ à l'étape courante. Ainsi les variations des variables A, B et B_0 sont indépendantes du paramètre λ_2. En ré-exprimant les conditions de transition des différents paramètres en fonction de A, B et B_0 on peut mettre en exergue leur dépendance par rapport à λ_2 :

$$\begin{cases} \delta^1_{\lambda_1} = \max\limits_{i\in\mathscr{E}} \min\Big(\dfrac{1-\alpha_i}{\frac{\Delta A_i}{\Delta\lambda_1}}, \dfrac{0-\alpha_i}{\frac{\Delta A_i}{\Delta\lambda_1}}\Big) \\ \delta^2_{\lambda_1} = \lambda_2 \max\limits_{i\in\mathscr{E}'} \dfrac{0-r_i}{-y_i(\frac{\Delta B_0}{\Delta\lambda_1} + \sum_{j\in\mathscr{V}_\beta} \frac{\Delta B_j}{\Delta\lambda_1})} \\ \delta^3_{\lambda_1} = \lambda_2 \max\limits_{j\in\mathscr{V}'_\beta} \dfrac{0-\beta_j}{\frac{\Delta B_j}{\Delta\lambda_1}} \\ \delta^4_{\lambda_1} = \max\limits_{j\in\mathscr{V}_0} \min\Big(\dfrac{\lambda_1(\gamma_j-1)}{-\sum_{i\in\mathscr{E}} \frac{\Delta A_i}{\Delta\lambda_1} y_i x_{ij} + 1}, \dfrac{\lambda_1(\gamma_j+1)}{-\sum_{i\in\mathscr{E}} \frac{\Delta A_i}{\Delta\lambda_1} y_i x_{ij} - 1}\Big) \\ \delta^5_{\lambda_1} = 0 - \lambda_1. \end{cases}$$

Ainsi plus la valeur de λ_2 est grande et plus les événements 2 et 3 sont retardés au profit des événements 1 et 4. Cela est cohérent, car une valeur élevée de λ_2 retarde les éventuelles annulations de variables ainsi que l'entrée des points dans $\mathscr{E}$ et par conséquent conduit à une solution moins parcimonieuse en comparaison à une simple régularisation L_1.

2.3.6 Robustesse du λ_1 chemin

Afin de justifier la robustesse du λ_1 chemin nous allons nous appuyer sur la forme de la matrice dérivée à inverser :

$$W = \begin{pmatrix} -x^T_{\mathscr{E},\mathscr{V}_\beta} \,.\, Dy_{\mathscr{E}} & \lambda_2 I_{\mathscr{V}_\beta} & 0_{\mathscr{V}_\beta,1} \\ -y^T_{\mathscr{E}} & 0_{1,\mathscr{V}_\beta} & 0 \\ 0_{\mathscr{E},\mathscr{E}} & x_{\mathscr{E},\mathscr{V}_\beta} & 1_{\mathscr{E},1} \end{pmatrix}. \tag{2.37}$$

Nous avons rencontré des problèmes de surdétermination de la solution sur le chemin en s, c'est à dire une impossibilité de faire évoluer la solution tout en faisant varier s. Ce cas survient quand $\mathscr{E} = \mathscr{V}_\beta + 1$ car la solution n'a alors plus de degré de liberté pour pouvoir varier quand s augmente. Si nous analysons la forme de la matrice (2.37), nous constatons que la dernière ligne fait intervenir uniquement les variables β_0 et β. Dans l'hypothèse où $\mathscr{E} = \mathscr{V}_\beta + 1$, on a $\frac{\Delta\beta}{\Delta\lambda_1} = 0$ et $\frac{\Delta\beta_0}{\Delta\lambda_1} = 0$ et la solution ne peut plus varier. Cependant les sous-différentielles α et γ peuvent continuer à évoluer par rapport à λ_1. Ainsi les seules contraintes susceptibles d'être violées sont relatives aux sous-différentielles. En effet si nous observons l'ensemble des conditions de transitions (2.35), l'invariance de β et β_0 implique que les événements 2 et 3 ont des pas $\delta\lambda_1$ infinis. et donc que par conséquent les contraintes ne peuvent être enfreintes.

Si c'est une composante de γ qui est enfreinte la première alors une nouvelle variable est activée et $|\mathscr{V}_\beta| \leftarrow |\mathscr{V}_\beta| + 1$. A l'itération suivante on a $|\mathscr{V}_\beta| > |\mathscr{E}|$ et les variables β_0 et β n'étant plus surdéterminées la solution peut à nouveau évoluer. Si c'est une composante de α qui est enfreinte la première alors un point quitte l'ensemble $\mathscr{E}$ et $|\mathscr{E}| \leftarrow |\mathscr{E}| - 1$. A l'itération suivante on a $|\mathscr{E}| < |\mathscr{V}_\beta|$ et les variables β_0 et β n'étant plus surdéterminées la solution peut à nouveau évoluer.

Dans toutes les configurations c'est l'évolution des sous-différentielles α et γ qui rend possible le rééquilibrage les cardinaux des ensembles $\mathscr{E}$ et $\mathscr{V}_\beta$. Ainsi tout en maintenant la solution constante, on fait varier λ_1 jusqu'à ce qu'une contrainte associée à une sous-différentielle soit enfreinte afin de nous ramener à une situation où la solution peut à nouveau varier par rapport à λ_1. Les problèmes de sur-dimensionnement de la solution survenant lors de la construction du chemin de régularisation par rapport au paramètre s, peuvent s'expliquer via une analyse similaire à partir du système (S) :

$$(S) \quad \begin{cases} \forall j \in \mathscr{V}_\beta \,,\, \lambda_2\beta - \sum_{i=1}^{n} \alpha_i y_i x_{ij} + sign(\beta_j)\eta & = 0 \\ \sum_{i=1}^{n} \alpha_i y_i & = 0 \\ \forall i \in \mathscr{E},\ 1 - (\beta_0 + \sum_{j=1}^{p} \beta x_{ij}) & = 0 \\ \|\beta\|_1 & = s. \end{cases}$$

En effet lorsque $\mathscr{E} = \mathscr{V}_\beta + 1$, la troisième ligne du système (S) indique que le paramètre β_0 ainsi que les composantes libres de β sont complètement déterminés par les équations et ne peuvent donc pas varier. La dernière ligne du système implique que le paramètre s est aussi fixé, ce qui conduit à une impasse pour la construction du chemin.

2.3.7 Simulation de chemin de régularisation λ_1

Nous nous proposons d'illustrer graphiquement la motivation de construction du λ_1-chemin plutôt qu'un s-chemin pour le *DRSVM*. Pour cela nous générons un ensemble de données où nous appliquons le λ_1-*DRSVM* et analysons le comportement des chemins de régularisation. Afin de distinguer le cas symptomatique (mis en évidence dans la section 2.2.5.4) du cas général, nous effectuons deux simulations correspondant à deux choix de paramètre λ_2. Pour chaque simulation nous représentons le chemin de régularisation de la solution (β_0, β) et de la sous différentielle α relativement à λ_1 et s. Nous commençons par construire le λ_1-chemin et nous en déduisons le s-chemin. En effet la condition KKT du problème (2.15) : $\eta(\|\beta\| - s) = 0$, implique $\|\beta\| = s$ sur tout le s-chemin. Ainsi il est possible de reconstruire le s-chemin en évaluant la norme 1 de β à chaque étape du λ_1-*DRSVM*. Une fois les deux chemins construits, nous analysons leur forme pour expliquer la surdétermination de la solution qui peut a priori apparaître sur le s-chemin.

2.3.7.1 Description des données

Les données se composent de deux classes équilibrées I^+ et I^-, comportant chacune 50 points de dimension 100. Les classes « plus » et « moins » sont générées à partir de lois normales dont les matrices de covariances sont diagonales et dont les espérances sont respectivement $\mu_{I^+} = (1_{1,50}, 0_{1,50})^T$ et $\mu_{I^-} = (0_{1,100})^T$. Ainsi seules les 50 premières variables sont discriminantes et sont susceptibles d'être sélectionnées pendant la construction du chemin de régularisation. Nous allons résoudre le problème DRSVM sur ce jeu de donnée pour deux valeurs de λ_2 d'ordre différent.

2.3.7.2 Cas 1 : λ_2 à une valeur élevée

La première simulation est réalisée avec une valeur de λ_2 élevée ($\lambda_2 = 10$) et l'influence de la pénalisation de la norme deux est prépondérante relativement à la norme un. Dans cette situation le DRSVM a tendance à activer en priorité des variables plutôt que de faire transiter des points vers $\mathscr{E}$. Ainsi la condition $|\mathscr{V}_\beta| > |\mathscr{E}|$ est largement vérifiée sur tout le chemin et il n'y a pas de surdétermination de la solution (β, β_0) au cours du chemin et il est alors possible de construire intégralement un s-chemin. Afin d'illustrer ce comportement, nous avons, pour des questions

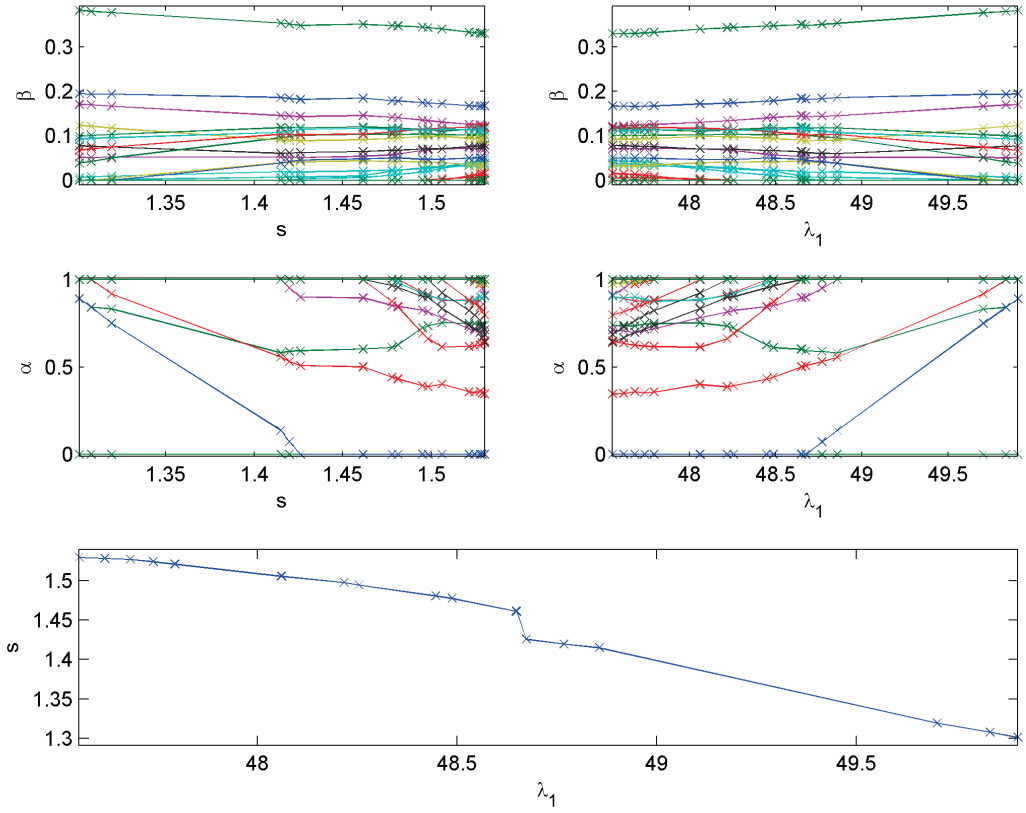


FIGURE 2.11 : Cette figure représente l'évolution de la solution et de la sous différentielle α en fonction des paramètres s et de λ_1. La valeur de λ_2 est suffisamment grande pour empêcher la surdétermination de la solution sur le s-chemin. La sous-différentielle α évolue de manière continue par morceaux. La figure de droite inférieure montre qu'il existe une stricte bijection entre s et λ_1.

de lisibilité, affiché sur la figure 2.11 seulement une partie du chemin du régularisation. On peut en premier lieu observer que l'évolution des paramètres (α, β) est bien continue par morceaux par rapport à s, ce qui montre qu'il est effectivement possible de construire un s-chemin. On peut noter aussi le nombre important de composantes de β non nulles, ce qui indique que la condition $|\mathscr{V}_\beta| > |\mathscr{E}|$ est bien vérifiée. Enfin on peut constater que l'évolution de s par rapport à λ_1 est strictement décroissante, il y a donc une stricte bijection entre ces deux paramètres. En d'autres termes, il n'y a pas de zone sur le λ_1-chemin où il est nécessaire de faire évoluer la sous-différentielle α à s constant. Ainsi on peut donc déduire indifféremment le λ_1-chemin à partir du s-chemin ou réciproquement le λ_1-chemin à partir du s-chemin.

2.3.7.3 Simulation avec λ_2 à une valeur basse

La deuxième simulation est réalisée avec une valeur plus faible de λ_2 faible ($\lambda_2 = 0.2$) et il existe une large zone sur le chemin où l'influence de la pénalisation de la norme un est prépondérante relativement à la norme deux. Dans cette situation le *DRSVM* a tendance à faire transiter des points vers $\mathscr{E}$ plutôt que d'activer des variables. Ainsi la condition $|\mathscr{V}_\beta| > |\mathscr{E}|$ est violée sur une bonne partie du chemin et une surdétermination de la solution (β, β_0) apparaît sur le chemin, empêchant la construction d'un *s*-chemin. La figure 2.12 illustre ce phénomène et on peut observer que l'évolution de la sous-différentielle α par rapport à s présente des points de discontinuité. Ces points de discontinuité sont de deux natures : une composante de α passe brutalement à la valeur 0 ou 1 (sortie d'un point de $\mathscr{E}$) ou bien la discontinuité de α coïncide avec l'activation d'une nouvelle variable. Dans ces deux situations, il est nécessaire de faire évoluer la sous-différentielle α par rapport à λ_1 en maintenant s constant. Cela peut aussi se remarquer en observant l'évolution de s par rapport à λ_1. En effet, on note la présence de plateaux et il n'y a donc pas de bijection entre ces deux paramètres. Nous avons souligné en rouge un plateau et repéré les zones correspondantes sur le α-chemin et λ_1-chemin, on note effectivement un point de discontinuité sur le *s*-chemin.

Nous avons généré un jeu de donnée pour lequel il existe une plage de valeurs de λ_2 où il est effectivement possible de construire un α-chemin, et une autre plage de valeurs où le choix du λ_1-chemin s'impose à cause de la discontinuité de la sous-différentielle α sur le *s*-chemin. Dans les deux cas le λ_1-chemin reste continu par morceaux pour β et α. Ainsi cela illustre le pertinence de construire a priori un λ_1-chemin plutôt qu'un *s*-chemin pour éviter tout problème de surdétermination.

2.4 Conclusion

Nous nous sommes intéressés dans ce chapitre à un problème de classification intégrant une double pénalisation $L_1 - L_2$ appelée *elastic net*. Dans une optique de construction de modèles interprétables, la méthode des chemins de régularisation, nous a semblé particulièrement adaptée, dans la mesure où elle permet de suivre et d'analyser l'évolution de la solution en fonction du paramètre de régularisation.

Cela nous a conduit à implémenter un algorithme 1 proposant la construction d'un chemin en L_1 à partir d'un problème équivalent (2.15) au DRSVM. Cependant, au cours d'une phase d'expérimentation, nous avons été confrontés à des situations problématiques, conduisant à des arrêts prématurés du chemin. En particulier, nous avons observé l'apparition d'une surdétermination de la solution pour des chemins parcimonieux. L'analyse de l'origine de cette surdétermination nous a incité à réétudier la formulation initiale du DRSVM (2.2), qui est non

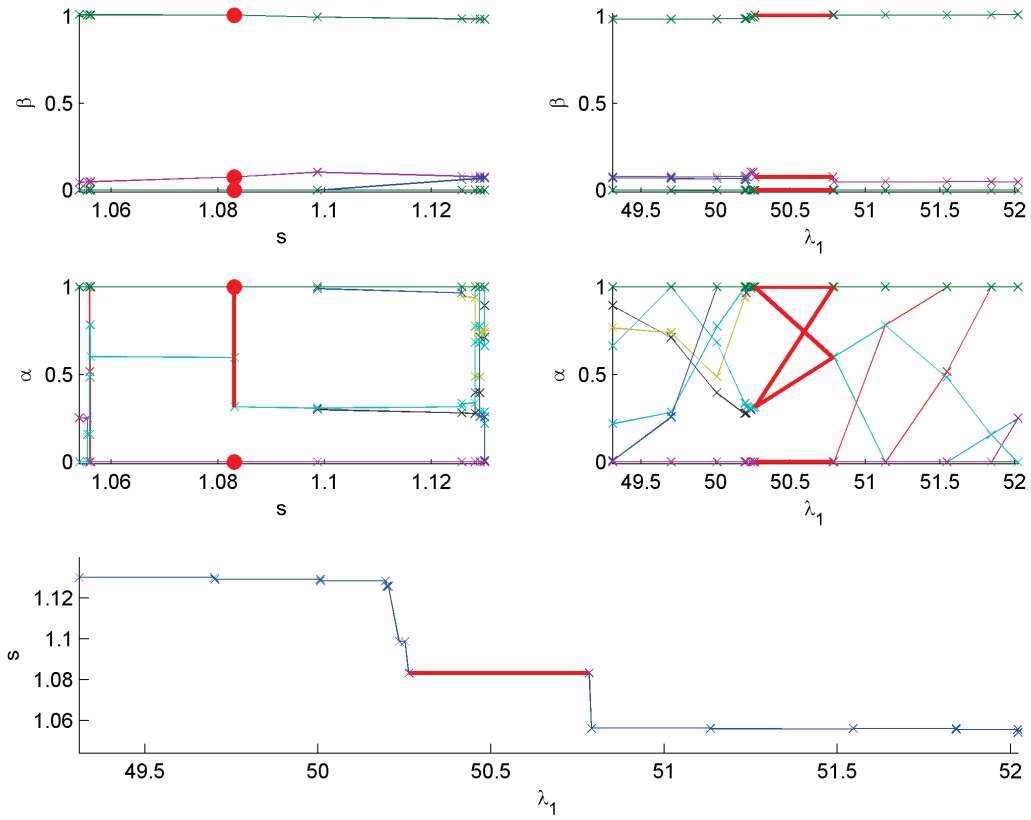


FIGURE 2.12 : Cette figure représente l'évolution de la solution et de la sous différentielle α en fonction des paramètres s et de λ_1. La valeur trop faible de λ_2 conduit à une surdétermination de la solution. Cette dernière est symbolisée par des points de discontinuités de la sous-différentielle α par rapport à s. La sous-différentielle α reste continue par morceaux par rapport à λ_1.

différentiable. Puis, nous avons choisi d'utiliser la théorie de la sous-différentielle, afin de dériver les conditions d'optimalité et de construire un chemin en L_1. Ensuite, nous avons démontré que notre construction de chemin permettait de gérer les arrêts prématurés en cours de chemin en recensant les situations pouvant conduire à une surdétermination de la solution. Enfin, nous avons, à travers une simulation, affiché les chemins associés respectivement au problème (2.15) et (2.2) et avons vérifié la présence d'une discontinuité sur le premier chemin. Nous avons également contrôlé la validité la construction de notre chemin à l'aide d'un solveur quadratique.

Un bon processus de sélection de variable est essentiel afin d'intégrer au sein du modèle final des informations pertinentes sur les données et obtenir une forme d'interprétabilité de la solution. Mais le DRSVM dans sa formulation initiale, est un algorithme linéaire et ne permet d'introduire d'autres formes de représentations des données. Or, il existe une approche intéressante, appelée *méthode à noyaux*, qui permet de changer l'espace de représentation des données et de prendre en compte leur structure propre. Ainsi, nous allons nous intéresser dans le chapitre suivant à des manières de combiner le DRSVM avec des méthodes à noyaux.

3

Kernelisation DRSVM

Sommaire

3.1 Machines à noyaux 82
3.1.1 Définition du noyau positif 83
3.1.2 Théorème de représentation 87
3.1.3 Noyaux et interprétabilité 88
3.2 Le modèle *kernel basis*, une approche multi-noyau 90
3.2.1 Les machines multi-noyau 90
3.2.2 Le modèle *kernel basis* 91
3.3 Formalisation du problème *kernel basis* via les RKHS 94
3.3.1 Présentation du *kernel basis* 94
3.3.2 Construction d'un espace de Hilbert associé au *kernel basis* 95
3.3.3 Théorème de représentation et *kernel basis* 99
3.3.4 Discussion 100
3.4 Kernelisation du DRSVM 101
3.4.1 Motivation 101
3.4.2 DRSVM et approche *kernel basis* 102
3.4.3 Chemin de régularisation et *kernel basis* 103
3.5 Expérimentations pour le *kernel basis* 105
3.5.1 Analyse du comportement du modèle KB_DRSVM 105
3.5.2 Influence de la norme L_1 sur la forme du modèle 110
3.5.3 KB_DRSVM et robustesse au bruit 113
3.5.4 Application à des données images 116
3.6 Conclusion 121

Dans le chapitre précédent nous avons proposé, dans un cadre linéaire, un algorithme robuste de classification intégrant un processus de sélection de variables et d'individus. L'intérêt « interprétable » de cette méthode est lié au fait qu'elle nous permet d'extraire les variables importantes pour notre problème. Mais nous ne pouvons pas les interpréter pour « expliquer » la solution. Ce fait est structurellement lié au caractère linéaire du modèle.

Une manière de promouvoir l'aspect interprétable de la solution et d'introduire des non-linéarités, consiste à rechercher une explication à travers des prototypes. L'idée est d'arriver à sélectionner, parmi les exemples disponibles, les exemples importants qui seraient des prototypes, chacun associé à une forme spécifique d'une zone de l'espace traduisant la manière dont il influence son environnement. Pour atteindre cet objectif, nous nous proposons d'utiliser une machine à noyau en associant au DRSVM non pas un seul noyau mais plusieurs noyaux de différentes formes. L'idée est, qu'à l'issue de l'apprentissage, nous disposions d'un ensemble de couples observations-noyaux permettant d'interpréter les résultats (comme l'illustre la figure 3.4). La difficulté est de trouver le moyen d'introduire les différents noyaux candidats dans le modèle DRSVM, ce que nous allons appeler la *kernelisation* du DRSVM.

Dans ce chapitre nous commençons par étudier les noyaux, certaines de leurs propriétés et notamment leur caractère interprétable. Nous avons ensuite recherché différentes architectures à noyau afin de l'intégrer au DRSVM. Cela nous a conduit à nous intéresser à une forme particulière de modèle appelé *kernel basis* dont nous avons illustré l'intérêt potentiel pour induire de l'interprétabilité au sein de la solution. Puis nous avons cherché à analyser cette forme de modèle sous l'angle de la théorie des RKHS (Reproducing Hilbert Space), un formalisme riche associé en particulier au théorème de représentation. Les résultats de cette étude nous ont conduit à nous intéresser à des pénalités de type $L_1 - L_2$. Nous expliquons alors comment à travers une approche dictionnaire, nous réalisons une *kernelisation* du DRSVM. Enfin nous avons mené une série d'expérimentations, afin de caractériser notre construction de modèle en vue d'évaluer sa capacité à induire de l'interprétabilité au sein de la solution tout en réalisant une tâche de classification.

3.1 Machines à noyaux

Parmi les algorithmes du domaine de reconnaissance de forme, une famille de méthodes est particulièrement séduisante : les machines à noyaux. La spécificité de cette approche repose sur la notion de noyau qui permet d'introduire efficacement de la non linéarité dans un modèle. Ce concept a suscité un grand intérêt de la part des communautés d'analyse mathématique, de statistique et de Machine Learning. La notion étant relativement ancienne [début $20^{\text{ième}}$ siècle Mercer, 1909], sa définition a connu plusieurs évolutions à travers l'histoire [Canu et al., 2002]. Ici nous nous limitons à l'étude des noyaux dits *positifs*. Les noyaux ont eu un fort impact sur

le développement des algorithmes d'apprentissage. Nous donnons ici trois définitions équivalentes du noyau positif, les preuves abordées ici sont inspirées du livre Support Vector Machines [Steinwart and Christmann, 2008].

3.1.1 Définition du noyau positif

De façon très informelle un noyau est une mesure de similarité entre deux objets supposés de même nature. Un noyau prend en entrée deux arguments (les objets à comparer appartenant à un même ensemble) et renvoie une valeur scalaire, qui, dans l'acception mathématique générale, peut être à valeurs plus complexes. Cependant les noyaux à valeurs complexes étant rarement utilisés dans le cadre de l'apprentissage machine, nous nous limiterons dans cette section à la présentation de la définition des noyaux à valeurs réelles.

Les noyaux possèdent une structure riche dont les nombreuses propriétés ont été étudiées abondamment par la communauté [Kung, 2014]. L'intérêt pour les noyaux de la part de la communauté de Machine Learning, a émergé à l'issue de l'introduction des SVM par Cortes and Vapnik [1995]. L'algorithme SVM via le concept de *kernel trick*, permet de construire des modèles de classification dont les frontières de séparation complexes permettent de traiter des jeux de données qui ne sont pas linéairement séparables. Le cadre statistique unifié grâce à l'introduction de la notion de marge a ainsi conduit à rechercher d'autres architectures basées sur les noyaux.

Supposons l'ensemble des données d'étude $\mathscr{X}$ non vide, un noyau est une fonction $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ pour laquelle il existe un espace de Hilbert H (voir définition 3) et une fonction $\phi : \mathscr{X} \to H$ appelée *mapping* tel que k s'identifie à un produit scalaire.

Définition 3 (espace de Hilbert)**.** *H est appelé espace de Hilbert ssi il possède une structure d'espace vectoriel, un produit scalaire noté $\langle \cdot, \cdot \rangle_H$ et qu'il est complet.*

Définition 4 (Noyau 1)**.** *Soit $\mathscr{X}$ un ensemble, k est un noyau si il existe un espace de Hilbert H et une fonction $\phi : \mathscr{X} \to \mathbb{R}$ tel que :*

$$\forall (x,x') \in \mathscr{X}^2, \quad k(x,x') = \big\langle \phi(x), \phi(x') \big\rangle_H .$$

On peut démontrer à titre d'exemple que la fonction $k(x,x') = \exp\big(-\frac{(x-x')^2}{2}\big)$ est un noyau en remarquant qu'elle vérifie la définition 4 avec :

$$\begin{cases} H = L^2 \quad \text{(l'espace des fonctions de carré intégrables définies sur } \mathscr{X} \text{ à valeurs réelles)}, \\ \forall (f,g) \in H^2, \langle f,g \rangle_H = \int_{\mathscr{X}} f(x)g(x)dx, \\ \phi(x) = t \mapsto exp\big(-(x-t)^2\big). \end{cases}$$

C'est de cette définition, que le concept de *kernel trick* ou *astuce du noyau* a été dérivé par Vapnik. Bien qu'initialement introduite à travers les SVM, l'astuce du noyau a été appliquée quasi systématiquement à d'autres algorithmes linéaires faisant intervenir la notion de produit scalaire. On parle alors de *kernelisation* d'algorithme comme dans la PCA [Mika et al., 1999b], l'analyse discriminante de Ficher discriminant [Mika et al., 1999a], le maching pursuit [Vincent and Bengio, 2002], la régression logistique [Hastie and Tibshirani, 1990], la régression ridge [Hastie et al., 2005], les PLS [Rosipal et al., 2003]. En classification, l'astuce du noyau consiste à changer l'espace d'étude $\mathscr{X}$ en transférant les données initiales vers H via le mapping ϕ. Ainsi H est un espace de Hilbert dont la dimension est possiblement infinie qui devient l'espace dans lequel on apprend une fonction $f : z \mapsto b + \langle w, z \rangle_H$ selon l'algorithme de classification linéaire dont on souhaite réaliser la *kernelisation*. En théorie, cet espace de Hilbert est choisi de manière à ce que les données transformées soient linéairement séparables et que l'on puisse appliquer un algorithme linéaire dans cet espace. Une fois l'apprentissage effectué, on obtient une fonction f linéaire dans H mais a priori non linéaire dans l'espace initial d'étude. En effet, une non-linéarité du *mapping* ϕ engendre la non-linéarité de $f : x \to f(\phi(x))$ qui possède alors une capacité de séparation plus importante qu'un classifieur linéaire. L'astuce du noyau consiste à faire disparaître totalement l'expression de la fonction ϕ et d'exprimer la solution f du problème à minimiser uniquement à partir du noyau k.

Dans la pratique, il est difficile, voire impossible d'expliciter un *mapping* ϕ où l'espace H par rapport auxquels k est défini. Réciproquement, étant donnée une fonction k, on peut se demander si elle définit un noyau. Aussi il serait utile d'avoir un critère simple afin de répondre à cette question. Or, il existe une définition équivalente du noyau où $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ est une forme, symétrique et semi-définie positive :

Définition 5 (Noyau 2)**.** *k est un noyau si $\forall n$ fini, $\forall X_n = (x_i)_{1 \leq i \leq n} \subset \mathscr{X}, \forall \alpha_n = (\alpha_i)_{1 \leq i \leq n} \subset \mathbb{R}^n$,*

$$\sum_{1 \leq i,j \leq n} \alpha_i k(x_i, x_j) \alpha_j \geq 0.$$

La définition 5 a l'avantage d'être relativement manipulable et peut être ainsi utilisée afin de démontrer un certain nombre de propriétés élémentaires sur les noyaux. Soit un réel positif λ et deux noyaux k_1 et k_2, $(x_i)_{1 \leq i \leq n} \subset \mathscr{X}$ et $(\alpha_i)_{1 \leq i \leq n}$ alors :

$$\begin{aligned}
\sum_{1 \leq i,j \leq n} \alpha_i (k_1(x_i, x_j) + k_2(x_i, x_j)) \alpha_j &= \sum_{1 \leq i,j \leq n} \alpha_i k_1(x_i, x_j) \alpha_j + \sum_{1 \leq i,j \leq n} \alpha_i k_2(x_i, x_j) \alpha_j \quad \geq 0 \\
\sum_{1 \leq i,j \leq n} \alpha_i (\lambda k_1(x_i, x_j)) \alpha_j &= \lambda \sum_{1 \leq i,j \leq n} \alpha_i k_1(x_i, x_j) \alpha_j \quad \geq 0 \\
\sum_{1 \leq i,j \leq n} \alpha_i (k_1(x_i, x_j) \times k_2(x_i, x_j)) \alpha_j &\geq 0 \quad \text{(d'après le théorème du produit de Schur).}
\end{aligned}$$

Chaque ligne prouve respectivement que les combinaisons suivantes sont des noyaux :

$$k_1 + k_2,$$
$$\lambda k_1,$$
$$k_1 \times k_2.$$

Plus précisément, les noyaux possèdent la structure mathématique de cône convexe, qui est particulièrement intéressante pour générer facilement de nouveaux noyaux à partir de ceux connus. On peut ainsi montrer que les noyaux polynomiaux sont effectivement bien des noyaux. Supposons que k est un noyau au sens de la définition (4). Soit $(\alpha_i)_{1\leq i\leq n} \subset \mathbb{R}^n$ et $(x_i)_{1\leq i\leq n} \subset \mathscr{X}$ alors :

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i k(x_i,x_j)\alpha_j &= \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \left\langle \phi(x_i),\phi(x_j)\right\rangle_H \alpha_j \\
&= \left\langle \sum_{i=1}^{n} \alpha_i k(x_i,\cdot), \sum_{j=1}^{n} \alpha_j k(x_j,\cdot) \right\rangle_H \\
&= \| \sum_{i=1}^{n} \alpha_i k(x_i,\cdot)\|_H^2 \qquad \geq 0.
\end{aligned}$$

Ainsi un noyau au sens de 4 est un noyau au sens de 5.

La troisième définition du noyau est liée à la notion de RKHS (Reproducing Kernel Hilbert Space) [Aronszajn, 1950]. Un RKHS $\mathscr{H}$ est un espace de Hilbert de fonctions définies de $\mathscr{X}$ vers $\mathbb{R}$ tel que $\forall x \in \mathscr{X}$ la fonctionnelle d'évaluation $\delta_x : \mathscr{H} \to \mathbb{R}$ est continue :

Définition 6 (RKHS)**.** *$\mathscr{H}$ est un RKHS ssi $\forall x \in \mathscr{X}$, la fonctionnelle d'évaluation $\delta_x(f) = f(x)$ est continue.*

Dans un RKHS la convergence en norme d'une fonction vers sa limite implique la convergence ponctuelle :

$$\lim_{n\to\infty} \|f_n - f\|_{\mathscr{H}} = 0 \Rightarrow \forall x \in \mathscr{X} \lim_{n\to\infty} f_n(x) = f(x). \tag{3.1}$$

La propriété (3.1) est particulièrement importante dans le cadre de l'apprentissage car on désire que lorsqu'on se rapproche d'un minima au sens de la norme de l'espace de fonctions, les sorties de la solution approchée soient proches des sorties de la solution optimale. Il existe une autre définition du RKHS, faisant intervenir une fonction $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ appelée noyau reproduisant et qui vérifie les propriétés suivantes :

Définition 7 (Noyau 3)**.** *k est un noyau ssi*

$$\forall x \in \mathscr{X},\ k(x,.) \in \mathscr{H},$$
$$\forall x \in \mathscr{X}, \forall f \in \mathscr{H},\ f(x) = \langle f(.), k(x,\cdot),\rangle_{\mathscr{H}}.$$

L'évaluation d'une fonction en un point x peut être calculée grâce à son produit scalaire avec la fonction $k(x,\cdot)$. Les deux définitions du RKHS sont équivalentes. En effet, d'après la définition 6, δ_x est une forme linéaire de $\mathscr{H}$ (autrement dit un élément du dual de $\mathscr{H}$) et continue (par hypothèse), on peut lui appliquer le théorème de représentation de Riesz dans $\mathscr{H}$ en tant qu'espace de Hilbert, c'est à dire :

$$\exists! k_x \in \mathscr{H} \quad \text{tel que} \quad \forall f \in \mathscr{H}, \quad \delta_x(f) = \langle f(), k_x() \rangle_{\mathscr{H}}.$$

Si on pose $\forall (x,x') \in \mathscr{X}^2$, $k(x,x') = \langle k_x(), k'_x() \rangle_{\mathscr{H}}$ alors :

$$\begin{array}{ll} \forall (x,x') \in \mathscr{X}^2, & k(x,x') = k_x(x') \Rightarrow \forall x \in \mathscr{X}, \quad k(x,.) = k_x(.) \in \mathscr{H} \\ \forall\, f \in \mathscr{H} \text{ et x } \in \mathscr{X}, & \langle f(.), k(x,.) \rangle_{\mathscr{H}} = \langle f(.), k_x(.) \rangle_{\mathscr{H}} = f(x). \end{array}$$

Ainsi k est bien un noyau reproduisant sur $\mathscr{H}$ au sens de (7).

Réciproquement si on a $x \in \mathscr{X}$ et $(f_n)_{n\in\mathbb{N}}$ une suite de fonctions de $\mathscr{H}$ qui convergent vers f au sens de la norme $\|\cdot\|_{\mathscr{H}}$, induite par $\langle \cdot,\cdot \rangle_{\mathscr{H}}$, c'est à dire tel que : $\|\cdot\|_{\mathscr{H}} = \langle \cdot,\cdot \rangle_{\mathscr{X}}^{1/2}$, alors :

$$|\delta_x(f_n) - \delta_x(f)| = |f_n(x) - f(x)| = |\langle f_n - f, k(x,.) \rangle|.$$

Si on applique le théorème de Cauchy-Schwarz, on obtient :

$$|\delta_x(f_n) - \delta_x(f)| \leq \|f_n - f\|_{\mathscr{H}} \cdot \|k(x,.)\|_{\mathscr{H}} \Rightarrow \lim_{n\to\infty} |f_n - f| = 0.$$

Ainsi δ_x est continue et $\mathscr{H}$ est bien un RKHS au sens de 6 et les deux formulations du RKHS sont bien équivalentes.

Si k est un noyau au sens de (5) alors si on pose $G = span(k(\cdot,x))$ (l'ensemble des fonctions combinaisons linéaires finies de $k(x,\cdot)$), G est un espace vectoriel de fonctions à valeurs réelles définies sur $\mathscr{X}$. Soit $(f,g) \in G^2$, $\exists (f_i)_{1\leq i\leq n}$ et $\exists (g_i)_{1\leq i\leq n'}$ tels que :

$$f(\cdot) = \sum_{i=1}^{n} f_i k(\cdot, x_i) \quad \text{et} \quad g(\cdot) = \sum_{i=1}^{n'} g_i k(\cdot, x_i).$$

On peut définir la forme bilinéaire symétrique suivante $\langle f,g \rangle_H$:

$$\langle f,g \rangle_H = \sum_{i=1}^{n} \sum_{j=1}^{n'} f_i g_i k(x_i, x_j).$$

On peut montrer que $\langle \cdot,\cdot \rangle_G$ est un produit scalaire, c'est à dire une forme symétrique définie positive.

$$\langle f,f \rangle_G = \sum_{i=1}^{n} \sum_{j=1}^{n} f_i f_j k(x_i, x_j) \geq 0,$$

car k est par hypothèse semi-positif.

Supposons que $\langle f, f\rangle_G = 0$ et $x \in X$ alors :

$$|f(x)| = \langle f(\cdot), k(x,\cdot)\rangle_H \leq \langle f(\cdot), f(\cdot)\rangle_H \langle k(x,\cdot), k(x,\cdot)\rangle_H = 0 \text{ (d'après Cauchy-Schwarz). (3.2)}$$

Donc $\forall x \in \mathscr{X}, f(x) = 0 \Rightarrow f = 0$. Ainsi $\langle \cdot,\cdot\rangle_G$ est bien un produit scalaire et G un espace préhilbertien.

Si on pose $\mathscr{H}$ comme la fermeture de G, c'est à dire $\mathscr{H} = \overline{G}$, et que l'on construit $\langle,\rangle_{\mathscr{H}}$ comme le prolongement par continuité de $\langle,\rangle_G$, $\mathscr{H}$ est complet et c'est donc un espace de Hilbert.

Soit $f \in \mathscr{H}$ alors $\exists (f_n)_{n\in\mathbb{N}}$ et $(x_n)_{n\in\mathbb{N}}$ tel que : $f = \sum_{n=0}^{\infty} f_n k(x_n,\cdot)$,

$$\begin{aligned}
\langle f(\cdot), k(x,\cdot)\rangle_{\mathscr{H}} &= \left\langle \sum_{n=0}^{\infty} f_n k(x_n,\cdot), k(x,\cdot) \right\rangle_{\mathscr{H}} \\
&= \sum_{n=0}^{\infty} f_n \langle k(x_n,\cdot), k(x,\cdot)\rangle_H \\
&= \sum_{n=0}^{\infty} f_n k(x_n,x) \\
&= f(x).
\end{aligned}$$

Ainsi la définition 7 est vérifiée et k est un noyau reproduisant et G est donc un RKHS

Réciproquement, supposons que k est un noyau reproduisant, si on pose $\forall x \in \mathscr{X}$, $\phi(x) = k(x,)$ alors :

$$\forall (x,x') \in \mathscr{X}^2, \quad k(x,x') = \left\langle k(x,\cdot), k(x',\cdot)\right\rangle_{\mathscr{H}} = \left\langle \phi(x), \phi(x')\right\rangle_{\mathscr{H}}.$$

Ainsi k est bien un noyau au sens du Kernel Trick 4, ce qui achève le raisonnement circulaire et montre l'équivalence des trois définitions du noyau. Notons que la définition (7) est importante dans la mesure où elle est liée au théorème de représentation, qui établit que pour certains problèmes d'apprentissage dont la résolution est envisagée dans un RKHS $\mathscr{H}$, la forme de la solution optimale peut s'exprimer directement en fonction d'un noyau reproduisant.

3.1.2 Théorème de représentation

Un RKHS, grâce à sa régularité, possède un certain nombre de propriétés liant les fonctions de cet espace et son noyau reproduisant. La richesse de la structure de cet espace de fonctions est utile lors de la recherche de solutions. En particulier un théorème important des RKHS appelé

le théorème de représentation, garantit que la solution a une forme particulière que l'on peut relier au noyau reproduisant [Kimeldorf and Wahba, 1971]. Plus précisément, si l'on considère le problème de minimisation régularisée suivant :

$$\min_{f \in \mathscr{H}} \quad R_{emp}(S_n, l, f) \; + \; \lambda \Omega(\|f\|_{\mathscr{H}}), \tag{3.3}$$

avec $S_n = (x_i, y_i)_{1 \leq i \leq n}$ un échantillon de données étiquetées, R_{emp} le risque empirique, l une fonction de coût quelconque, Ω une fonction régularisatrice strictement croissante, $\mathscr{H}$ un RKHS dont le noyau reproduisant est k et λ un paramètre de régularisation qui contrôle l'influence de Ω sur la solution. Le théorème de représentation affirme qu'il existe une famille de scalaires telle que la solution de (3.3) s'exprime sous la forme :

Théorème 5 (Théorème de représentation, théorème 4.2 dans Schölkopf et al. [2001])**.** *La fonction f^*, solution du problème de minimisation régularisée 3.3, admet une représentation de la forme :*

$$f^*(\cdot) = \sum_{1 \leq i \leq n} \alpha_i k(x_i, \cdot),$$

où les α_i sont des réels traduisant l'influence du i-ème l'exemple dans la construction de la solution.

Ce résultat est très important car il affirme que toute solution au problème (3.3) peut s'exprimer comme une combinaison linéaire de la fonction noyau. La structure de la solution est très particulière, car pour caractériser la solution il suffit de trouver les coefficients associés à la décomposition. La démonstration de ce théorème est basée sur la décomposition orthogonale (voir Schölkopf et al. [2001]).

3.1.3 Noyaux et interprétabilité

La capacité des noyaux à générer des modèles non linéaires a fortement participé à leur popularisation au sein de la communauté Machine Learning. De plus, ces derniers présentent à plusieurs niveaux un aspect interprétable, dont l'utilisation peut être pertinente afin d'apprendre des machines moins opaques, où il est possible d'extraire une certaine forme d'intelligibilité à partir de la forme de la solution.

3.1.3.1 Le modèle compare les individus et non les variables

Le paradigme du noyau renverse la façon dont est construit le modèle auquel il est intégré. Car, pour discriminer un nouvel individu, on n'utilise pas directement la valeur de ses variables, mais on évalue sa similarité relativement aux points de la base d'apprentissage, via l'utilisation

du noyau. En particulier, pour certains problèmes d'apprentissage, le théorème de représentation 5 garantit la forme de la solution. La forme de la solution f dépend uniquement d'un sous-ensemble de points dont le coefficient $\alpha_i \neq 0$. Pour un modèle de classification, on peut analyser l'influence des points d'apprentissage sur la frontière de décision du séparateur, en analysant leurs coefficients associés au noyau dans la décomposition de la solution.

3.1.3.2 Le noyau est une mesure de similarité

Pour des données vectorielles, on peut identifier le noyau linéaire dans l'espace euclidien $\mathbb{R}^p$ au produit scalaire usuel, mesure de similarité par excellence :

$$k(x,x') = \langle x,x' \rangle_{\mathbb{R}^p} = x^T x',$$

avec p la dimension des données.

Mais cette définition représente les données uniquement comme des vecteurs multidimensionnels décontextualisés. Les données réelles, de part leur signification physique, possèdent des structures qui leur sont propres. Une image, par exemple, peut se représenter sous la forme vectorielle, mais l'écriture matricielle permet de prendre en compte le caractère spatial de l'information. Aussi, il peut être intéressant de construire des noyaux capables d'encapsuler de telles structures [Gärtner, 2003]. Cela a conduit à s'intéresser à certains objets mathématiques, afin de proposer des noyaux spécifiques à des données structurées telle que les chaînes de caractères [Lodhi et al., 2002] en décomposant les mots en sous-ensembles possédant une cohérence grammaticale, puis en comparant terme à terme ces sous-ensembles, ou encore les graphes [Kashima et al., 2004] qui sont décomposés en chemins dont on compare les attributs.

3.1.3.3 Noyaux experts

Le noyau peut être interprété également comme un moyen d'introduire un a priori expert, au sein du modèle auquel il est intégré. Un expert d'un domaine spécialisé peut définir manuellement un noyau en utilisant sa connaissance des données, et contrôler la validité de construction en s'appuyant sur les nombreuses propriétés des noyaux. Par exemple, dans une application de classification de données génétiques [Pavlidis et al., 2001], le noyau k a été construit manuellement de façon à prendre en compte le caractère hétérogène des données (génétiques et phylogénétiques), à partir d'une segmentation pertinente d'un point de vue biologique, des variables $x = (x_g, x_p)$:

$$k(x,x') = k(x_g,x'_g) + k(x_p,x'_p)$$

Cet exemple démontre ainsi la nécessité de pouvoir construire des fonctions noyaux dédiées aux données considérées, afin de pouvoir augmenter la pertinence du modèle construit par appren-

tissage. Il montre également l'intérêt de la souplesse de définition des noyaux pour fusionner au sein d'une même fonction des fonctions noyaux propres à chaque sous-ensemble.

3.2 Le modèle *kernel basis*, une approche multi-noyau

L'approche des machines à noyaux est extrêmement populaire de part sa capacité à générer des fonctions de décision non-linéaires à partir d'algorithmes linéaires standards d'apprentissage statistique. De plus cette approche peut induire une certaine forme d'interprétabilité de part la nature des noyaux (voir la discussion sur l'interprétabilité dans la section 3.1.3). Cependant la plupart des noyaux utilisés font intervenir au sein de leur formulation des paramètres, comme le degré pour le noyau polynomial ou la largeur de bande pour le noyau gaussien. Un choix inapproprié de paramètre est équivalent à effectuer le kernel trick dans un espace de Hilbert $\mathscr{H}$ mal adapté pour séparer linéairement les données. Nous présentons dans cette section deux approches permettant d'attaquer ce problème, les noyaux multiples et l'approche dite *kernel basis*. Nous verrons que, bien que très populaires, les noyaux multiples ne permettent pas d'interpréter les résultats comme nous le souhaitons alors que l'utilisation d'une base de noyaux le peut.

3.2.1 Les machines multi-noyau

Afin de prendre en compte la nature hétérogène de certaines données, il peut être judicieux d'utiliser des combinaisons de noyaux élémentaires, afin d'obtenir un modèle plus souple. C'est dans cette optique que l'approche appelée *multi-noyau* ou MKL (*Multiple Kernel Learning*) a été introduite [Lanckriet et al., 2004, Bach et al., 2004], afin de généraliser l'approche mono-noyau (utilisation d'un noyau unique pour la kernelisation). La méthode se fonde sur une propriété fondamentale des noyaux qui ont une structure conique, ce qui permet d'envisager toutes les combinaisons linéaires positives comme candidat potentiel de noyau. La solution MKL prend la forme suivante :

$$\begin{cases} f(x) = \sum_{i=1}^{n} \alpha_i \sum_{l=1}^{m} \beta_l k_l(x_i, x) & \\ \text{avec} \quad \forall l \in [1,m],\ \beta_l \geq 0 & (a) \\ \text{et} \quad \sum_{l=1}^{m} \beta_l = 1 & (b). \end{cases} \tag{3.4}$$

Notons que, puisque l'on envisage seulement des combinaisons positives, les conditions (3.4).(a,b) induisent une certaine parcimonie dans la décomposition des noyaux utilisés.

L'approche a été appliquée avec succès à la détection d'objets au sein d'une image, par une méthode de classification MKL [Vedaldi et al., 2009] en utilisant une famille multi-noyau d'his-

togrammes. Le problème d'optimisation associé au MKL étant complexe, certains algorithmes ont été proposés afin de réduire le temps de calcul et Rakotomamonjy et al. [2007] donne une résolution efficace de l'optimisation des poids des noyaux par une descente de gradient conjugué. Notons que si l'approche MKL a été initialement proposée pour le modèle SVM, elle a été par la suite étendue à la régression par Sonnenburg et al. [2005]. Dans une perspective d'interprétabilité, une proposition originale de Szafranski and Grandvalet [2014] consiste à utiliser la norme CAP [Zhao et al., 2009], afin d'intégrer au sein du modèle des informations de hiérarchie et de structure liée à la famille de noyau.

Cependant, nous pouvons observer que la formulation du MKL, de part sa formulation intrinsèque, introduit un découplage entre les coefficients associés respectivement aux points et aux noyaux (voire équation (3.4)). Le MKL construit un noyau plus souple, mais il est uniformément appliqué sur l'ensemble des données.

On peut envisager d'autres architectures de machines à noyaux, afin de pouvoir faire moduler l'influence du noyau en fonction de la partie locale de l'espace. C'est dans cette approche que Lewis et al. [2006] se propose de construire un classifieur à noyaux de type SVM dans le cas de l'utilisation de noyaux gaussiens où l'influence du noyau est pondérée localement. Dans la même optique Gönen and Alpaydin [2008] propose un modèle qui représente les coefficients de pondération comme des fonctions spatiales.

Dans un approche semblable, il existe une architecture multi-noyau, issue du domaine du traitement du signal, appelé *kernel basis* [Zhu and Hastie, 2001], qui permet d'assouplir la forme du modèle afin de pouvoir adapter localement la forme du noyau. Cette forme de modèle a particulièrement retenu notre attention et nous nous sommes intéressés à elle dans une perspective de construction de classifieurs interprétables.

3.2.2 Le modèle *kernel basis*

3.2.2.1 Forme du modèle *kernel basis*

Dans la section précédente, nous avons remarqué qu'il existe une certaine ambiguïté sur le terme de MKL (*Multiple Kernel Learning*), dans la mesure où il n'induit aucune précision sur la manière dont sont appliqués les noyaux au sein de la machine entraînée, mais indique seulement que plusieurs noyaux ont été optimisés pendant la phase d'apprentissage.

Afin d'articuler la discussion sur les différentes approches multi-noyau et d'illustrer leurs différences, le modèle *kernel basis* est introduit via l'équation (3.5). Soit $S_n = (x_j)_{1\leq j\leq n}$ ensemble de données et $\mathscr{K} = (k_l)_{1\leq l\leq m}$ une famille du noyau, nous disons qu'une machine multi-noyau

f est de la forme *kernel basis* si elle se décompose de la manière suivante :

$$f(x) = \sum_{j=1}^{n} \sum_{l=1}^{m} B_{jl} k_l(x_j, x), \tag{3.5}$$

avec B une matrice de taille $n \times m$.

Notons qu'il est possible d'inclure un biais au sein du modèle (3.5) en ajoutant le noyau constant dans la famille de noyaux $\mathscr{K}$.

Dans le cas particulier où la famille de noyaux se réduit à un singleton, c'est à dire pour $m = 1, \mathscr{K} = \{k\}$ alors B se réduit à un vecteur de pondération et la formulation (3.5) est alors équivalente à la forme d'une machine mono-noyau. Par exemple dans le cas des SVM, on peut identifier le vecteur B aux coefficients de Lagrange $\alpha = (\alpha_i)_{1 \leq i \leq n}$.

L'approche *multiple kernel learning* (MKL) peut aussi être interprétée comme un cas particulier du modèle (3.5) en choisissant $B = \alpha\beta^T$, avec α et β des vecteurs de tailles respectives n et m :

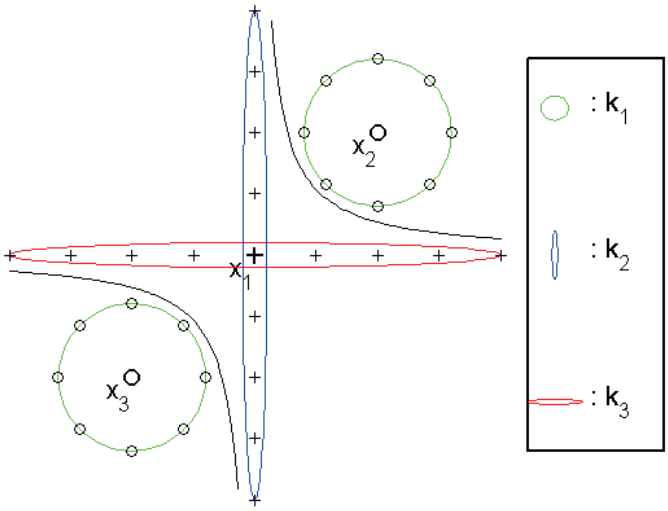
$$\begin{aligned} f(x) &= \sum_{j=1}^{n} \sum_{l=1}^{m} \alpha_j \beta_l k_l(x_j, x) \\ &= \sum_{j=1}^{n} \alpha_j K(x_j, x) \qquad \text{avec} \quad K(x', x) = \sum_{l=1}^{m} \beta_l k_l(x', x). \end{aligned}$$

Dans cette configuration, nous pouvons remarquer que la structure finale de la machine peut être mise en perspective avec celle de la machine mono-noyau car le noyau associé K, appris au cours d'un processus d'optimisation, est appliqué de manière uniforme sur l'ensemble des données d'observation. Dans sa forme la plus générale, la formulation du modèle (3.5) permet de moduler l'influence des noyaux selon les points considérés.

Le modèle *kernel basis* a été déjà appliqué à des algorithmes d'apprentissage. Guigue et al. [2005] ont développé un algorithme appelé *kernel basis pursuit* pour la discrimination de signaux via l'algorithme LARS. Aussi, Suard and Mercier [2009] ont étendu le RVM (Relevant Vector Machine) avec l'algorithme, comparé leurs résultats à l'algorithme MKLSVM et ainsi montré que cette approche donne des modèles plus interprétables que le MKLSVM. Notons que ces algorithmes ont été conçus à l'origine pour la régression.

3.2.2.2 Modèle *kernel basis* et interprétabilité

Dans une perspective d'interprétabilité, il peut être pertinent d'essayer d'extraire de l'information utile à partir d'un modèle généré à partir d'associations pertinentes entre les noyaux et les points d'observation. La principale différence entre le *kernel basis* et l'approche MKL réside dans la relation de dépendance entre les points d'apprentissage et les noyaux au travers des coefficients de couplage de la matrice B_{jl}. Cette plus grande souplesse du modèle permet



$$B = \begin{pmatrix} 0 & B_{12} & B_{13} \\ B_{21} & 0 & 0 \\ B_{31} & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix}$$

FIGURE 3.1 : Exemple d'un modèle de type *kernel basis* afin d'illustrer l'équation (3.5) avec $n = 33$ données d'observation issues d'un problème bi-classe (la classe 1 est représentée par '+' et la classe 2 est représentée par '$\circ$'), $m = 3$ noyaux (k_1, k_2 and k_3) et la matrice de couplage associée B.

de modifier l'influence du noyau sur un point d'observation selon sa localisation dans l'espace d'observation. Les éléments de B ont pour vocation à capturer les possibles synergies entre un jeu de données S_n et une famille de noyaux.

La figure 3.1 illustre le sens que l'on peut donner à la valeur des coefficients de B. Un coefficient nul : $B_{jl} = 0$ indique un manque de synergie entre le point x_j et le noyau k_l alors que $B_{jl} \neq 0$ indique que le noyau k_l présente une synergie avec la donnée x_j pour la tâche que l'on cherche à résoudre.

Dans l'exemple présenté, seulement 3 points (x_1,x_2 et x_3) participent au modèle. Chacun de ces points possède une synergie propre avec la famille de noyaux utilisée : x_1 est connecté à 2 noyaux k_2, k_3 (*i.e.* B_{12}, $B_{13} \neq 0$) alors que k_1 est associé à deux points x_2, x_3 (*i.e.* B_{21}, $B_{31} \neq 0$). Les couples (x_1, k_2), (x_1, k_3), (x_2, k_1) et (x_3, k_1) sont les seules combinaisons (point, noyau) de la base d'apprentissage qui participent au modèle. L'analyse de ces différents couples est essentielle pour l'interprétabilité. Cette notion est définie de la façon suivante :

Définition 8 (Couplage actif)**.** (j,l) *est appelé couple actif entre l'observation* x_j *et le noyau* k_l *ssi,* $B_{jl} \neq 0$.

Dans la suite du manuscrit, les expérimentations effectuées se rapporteront à cette définition pour expliquer les résultats obtenus et comprendre les modèles.

L'architecture du *kernel basis* nous a semblé particulièrement intéressante, dans la problématique de génération de classifieurs interprétables. Notons que le *kernel basis*, est une forme de

modèle et non un problème d'apprentissage en soi. Afin d'obtenir une solution ayant la forme *kernel basis*, il est nécessaire de définir une fonction de coût ainsi qu'une pénalité afin de poser le problème à minimiser. Dans la problématique étudiée ici, nous nous focalisons sur les tâches de classification. La fonction standard Hinge s'avère a priori être une bonne candidate, en particulier car elle ne nécessite pas de réglage de paramètres supplémentaires. Il reste à s'interroger sur la manière de pénaliser un modèle *kernel basis*. Nous avons commencé par étudier la pertinence à régulariser le modèle en norme 2. Puis, nous avons recherché des constructions de produits scalaires afin de générer un espace $\mathbb{G}$ de fonctions, où l'on puisse dériver un théorème de représentation pour le *kernel basis*.

3.3 Formalisation du problème *kernel basis* via les RKHS

Dans cette section, nous allons détailler la formulation d'une solution *kernel basis* et démontrer que cela s'inscrit dans un espace propre à des résolutions mathématiques, en l'occurrence un RKHS. En effet un RKHS vérifie un certain nombre de propriétés fournissant des informations sur la forme que prend la solution pour certains types de problèmes d'apprentissage auxquels nous sommes confrontés.

3.3.1 Présentation du *kernel basis*

Comme évoqué précédemment, l'efficacité des algorithmes basés sur les machines à noyaux tels que les SVM dépend en grande partie du choix du noyau utilisé. Afin de pouvoir traiter des données complexes, les MKL sont intéressants car ils permettent de créer un modèle plus souple, par le biais de l'optimisation d'un noyau k, une pondération positive de m noyaux $(k^l)_{1\leq l\leq m}$:

$$k = \sum_{l=1}^{m} \beta_l \, k^l \quad \text{avec} \quad \forall l \in [1,m],\ \beta_l \geq 0 \quad \text{et} \quad \sum_{l=1}^{m} \beta_l = 1.$$

La fonction k étant par hypothèse un noyau, il existe un RKHS $\mathscr{H}_k$ dont k est le noyau reproduisant. Ainsi si on envisage la résolution du problème d'apprentissage régularisé (3.3) dans l'espace $\mathscr{H}_k$, on peut appliquer le théorème de représentation et on obtient une solution de la forme :

$$f(x) = \sum_{i=1}^{n} \sum_{l=1}^{m} \gamma_{il} k^l(x_i, x) \quad \text{avec} \quad \gamma_{il} = \beta_l \times \alpha_i.$$

Cependant si nous observons la forme de la solution optimale, l'expression des coefficients γ introduit un découplage entre les points et les noyaux. Le modèle *kernel basis* introduit un

degré supplémentaire de liberté dans le sens où il induit une corrélation entre chaque point et chaque noyau indépendante via une famille de réels $(\alpha_{il})_{1\leq i\leq n}^{1\leq l\leq m}$:

$$f(x) = \sum_{i=1}^{n}\sum_{l=1}^{m} \alpha_{il} k^l(x_i, x). \tag{3.6}$$

On peut se demander s'il est possible de dériver un théorème de représentation associé au *kernel basis*, c'est à dire pour lequel la forme de la solution optimale est de la forme (3.6). Cela est équivalent à rechercher une définition de produit scalaire et de construire un espace de Hilbert associé tel que la solution du problème d'optimisation (3.3) a pour forme la fonction (3.6).

3.3.2 Construction d'un espace de Hilbert associé au *kernel basis*

Nous allons détailler ici la construction d'un espace associé au modèle *kernel basis* en nous inspirant de la théorie des RKHS. Soit un ensemble fini de noyaux $K = (k^j)_{1\leq j\leq m}$, c'est à dire des formes bilinéaires symétriques semi-positives. Nous supposons que tous ces noyaux sont définis sur le même ensemble $\mathscr{X}^2$ ($\mathscr{X}$ est l'espace des données) et qu'ils ne sont pas identiquement nuls sur $\mathscr{X}$. Quand nous fixons une variable du noyau k^l à la valeur x, ce dernier devient une fonction monovariable que nous notons k^l_x.

$$\forall x' \in \mathscr{X},\ k^l_x(x') = k^l(x, x').$$

Soit G l'ensemble des fonctions s'exprimant comme une combinaison linéaire finie des fonctions k^l_x :

$$G = \Big\{ f : \mathscr{X} \to \mathbb{R}, \exists n \in \mathbb{N} \text{ et } (x_i)_{1\leq i\leq n} \subset \mathscr{X} \text{ tel que } f = \sum_{i=1}^{n}\sum_{l=1}^{m} \alpha_{il} k^l_{x_i} \quad \text{avec } \alpha_{il} \in \mathbb{R} \Big\}.$$

Par construction G est un espace vectoriel de fonctions définies sur $\mathscr{X}$ à valeurs réelles. Soit $(f,g) \in G^2$, $\exists (f_{ij})_{1\leq i\leq n, 1\leq j\leq m} \in \mathbb{R}^{n\times m}$, $\exists (g_{ij})_{1\leq i\leq n', 1\leq j\leq m} \in \mathbb{R}^{n'\times m}$, $(x_i)_{1\leq i\leq n} \subset \mathscr{X}$ et $(x'_i)_{1\leq i\leq n'} \subset \mathscr{X}$ tels que :

$$f(\cdot) = \sum_{i=1}^{n}\sum_{l=1}^{m} f_{il} k^l_{x_i} \text{ et } g(\cdot) = \sum_{i=1}^{n'}\sum_{l=1}^{m} g_{il} k^l_{x'_i}.$$

Nous définissons $\langle f, g\rangle_G$ de la façon suivante :

$$\langle f, g\rangle_G = \sum_{i=1}^{n}\sum_{j=1}^{n'}\sum_{l=1}^{m} f_{il} k^l(x_i, x'_j) g_{jl}. \tag{3.7}$$

Remarquons que la définition (3.7) est à rapprocher de celle utilisée lors de la construction d'un RKHS à partir d'un noyau k^l, où l'on construit préalablement un espace pré-Hilbertien G_l à l'aide du produit scalaire $\langle \cdot, \cdot \rangle_{G_l}$ tel que :

$$\begin{cases} \langle f, g \rangle_{G_l} = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i k^l(x_i, x'_j) \beta_j \\ \text{avec } G_l = \{ \sum_{i=1}^{n} \alpha_i k^l(x_i, \cdot),\ (x_i)_{1 \leq i \leq n} \subset \mathscr{X} \}. \end{cases}$$

Remarquons qu'il est possible d'exprimer respectivement G et $\langle \cdot, \cdot \rangle_G$ en fonction de G_l et de $\langle \cdot, \cdot \rangle_{G_l}$:

$$G = \oplus_{l=1}^{m} G_l \text{ et } \quad \langle \cdot, \cdot \rangle_G = \sum_{l=1}^{m} \langle \cdot, \cdot \rangle_{G_l}.$$

Cependant si nous n'imposons pas de condition sur les noyaux qui génèrent G, la forme bilinéaire $\langle \cdot, \cdot \rangle_G$ n'est a priori pas une fonction. En observant la définition de $\langle \cdot, \cdot \rangle_G$ on peut conjecturer que la condition de liberté de la famille de noyaux est nécessaire et suffisante :

$$\nexists (\lambda_l)_{1 \leq l \leq m} \in \mathbb{R}^n - \{0\},\ \sum_{l=1}^{m} \lambda_l k^l = 0 \quad \text{(famille libre).} \tag{3.8}$$

Nous allons au préalable montrer que la condition (3.8) est nécessaire en raisonnant par l'absurde. Supposons qu'il existe $(\lambda_l)_{1 \leq l \leq m} \in \mathbb{R}^n - \{0\}$ tel que :

$$\sum_{l=1}^{m} \lambda_l k^l = 0.$$

Soit $l_0 \in [1, m]$ tel que $\lambda_{l_0} \neq 0$ on a

$$k^{l_0} = -\frac{1}{\lambda_{l_0}} \sum_{l=1, l \neq l_0}^{m} \lambda_l k^l.$$

Les noyaux étant par définition semi-positifs, nous avons $\forall x \in \mathscr{X}$, $k^{l_0}(x, x) \geq 0$. De plus nous avons supposé que nos noyaux ne sont pas identiquement nuls ainsi $\exists x_0 \in \mathscr{X}$ tel que $k^{l_0}(x_0, x_0) > 0$. Remarquons que toutes les fonctions k_x^l peuvent se décomposer de la manière suivante :

$$k_x^l = \sum_{i=1}^{m} \delta_i^l k_x^i,$$

avec δ_i^l le symbole de Kronecker.

Ainsi on a :

$$\left\langle k_x^l, k_{x'}^{l'} \right\rangle_G = \delta_l^{l'} k^l(x,x') = \begin{cases} k^l(x,x') \text{ si } l = l' \\ 0 \text{ si } l \neq l' \end{cases}$$

et

$$\begin{aligned} k^{l_0}(x_0,x_0) &= \left\langle k_{x_0}^{l_0}, k_{x_0}^{l_0} \right\rangle_G \\ &= \left\langle k_{x_0}^{l_0}, -\frac{1}{\lambda_0} \sum_{l=1, l\neq l_0}^{m} \lambda_l k_{x_0}^l \right\rangle_G \\ &= -\frac{1}{\lambda_0} \sum_{l=1, l\neq l_0}^{m} \left\langle k_{x_0}^{l_0}, k_{x_0}^l \right\rangle_G \\ &= -\frac{1}{\lambda_0} \sum_{l=1, l\neq l_0}^{m} \delta_{l_0}^l k^l(x_0,x_0) \\ &= 0. \end{aligned}$$

Cela nous conduit à un résultat absurde. Ainsi la condition (3.8) est nécessaire.

Nous conjecturons que la condition (3.8) est aussi suffisante mais n'ayant pu le démontrer avec certitude, nous proposons une condition alternative sur les noyaux dont il est facile de démontrer qu'elle est une condition suffisante afin de garantir que $\left\langle k_x^l, k_{x'}^{l'} \right\rangle_G$ est bien une fonction :

$$\forall l \neq l', G_l \cap G_l' = \{f : x \mapsto 0\}. \tag{3.9}$$

De façon équivalente, cela veut dire que nous supposons que les espaces de pré-Hilbert G_l engendrés par la famille de noyaux $(k^l)_{1\leq l\ m}$ n'ont aucune fonction en commun à exception de la fonction nulle.

Supposons maintenant que la condition 3.9 est vérifiée. Soient deux fonctions $(f,g) \in G^2$ telles que :

$$f(\cdot) = \sum_{i=1}^{n} \sum_{l=1}^{m} f_{il} k_{x_i}^l \quad \text{et} \quad g(\cdot) = \sum_{i=1}^{n'} \sum_{l=1}^{m} g_{il} k_{x_i'}^l.$$

On peut exprimer f et g sous la forme :

$$\begin{aligned} &f(\cdot) = \sum_{l=1}^{m} f_l \text{ et } g(\cdot) = \sum_{l=1}^{m} g_l \\ &\text{avec } f_l(\cdot) = \sum_{i=1}^{n} f_{il} k_{x_i}^l \text{ , } g_l(\cdot) = \sum_{i=1}^{n'} g_{il} k_{x_i'}^l \text{ et } (f_l, g_l) \in G_l^2. \end{aligned}$$

On a :

$$\langle f,g \rangle_G = \sum_{l=1}^{m} \langle f_l, g_l \rangle_{G_l}.$$

L'indépendance des noyaux impose l'unicité de la décomposition de f et g respectivement sur la famille $(f_l)_{1\leq l\leq m}$ et $(g_l)_{1\leq l\leq m}$. Donc $\langle f,g\rangle_G$ ne peut prendre qu'une seule valeur, c'est donc une fonction et la condition (3.9) est suffisante. Supposons dorénavant que la condition (3.9) est respectée. Nous allons montrer que G muni du produit scalaire $\langle\cdot,\cdot\rangle_G$ est un espace pré-Hilbertien.

Premièrement, on constate que de part sa définition $\langle\cdot,\cdot\rangle_G$ est une forme bilinéaire et symétrique. De plus $\langle\cdot,\cdot\rangle_G$ est positive car par hypothèse $\forall l\in[1,m]$, k^l est semi positif :

$$\begin{aligned}\langle f,f\rangle_G &= \sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n'} f_{il}k^l(x_i,x_j)f_{jl}\\ &= \sum_{l=1}^{m}\Big(\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n'} f_{il}k^l(x_i,x_j)f_{jl}}_{\geq 0 \quad \text{car } k^l \text{ est semi-positif}}\Big)\\ \Rightarrow \langle f,f\rangle_G &\geq 0.\end{aligned}$$

Ainsi $\langle\cdot,\cdot\rangle_G$ est une forme bilinéaire symétrique positive et il est alors possible de lui appliquer le théorème de Cauchy-Schwarz généralisé afin de démontrer qu'elle est définie positive.

Soit $f\in G$ tel que $f(\cdot)=\sum_{l=1}^{m} f_l(\cdot)=\sum_{i=1}^{n}\sum_{l=1}^{m} f_{il}k^l_{x_i}$ et $\langle f,f\rangle_G=0$. Au préalable remarquons que $\forall x\in\mathscr{X}$, $f_{l_0}(x)$ l'évaluation au point x de la projection de f sur G_{l_0}, peut se ré-exprimer de la manière suivante :

$$\begin{aligned}\left\langle f(\cdot),k^{l_0}(x,\cdot)\right\rangle_G &= \sum_{i=1}^{n}\sum_{l=1}^{m} f_{il}k^l(x_i,x)\delta^l_{l_0}\\ &= \sum_{i=1}^{n} f_{il_0}k^{l_0}(x_i,x)\\ &= f_{l_0}(x).\end{aligned}$$

Soient $x_0\in\mathscr{X}$ et $l_0\in[1,m]$, on a, d'après l'inégalité de Cauchy :

$$f_l(x_0)=\left\langle f(\cdot),k^{l_0}(x_0,\cdot)\right\rangle_G \leq \underbrace{\langle f(\cdot),f(\cdot)\rangle_G}_{=0}\times\left\langle k^{l_0}_{x_0},k^{l_0}_{x_0}\right\rangle_G = 0$$

$$\begin{aligned}\Rightarrow \quad & \forall x\in\mathscr{X},\ \forall l\in[1,n],\ f_l(x)=0\\ \Rightarrow \quad & \forall x\in\mathscr{X}, f(x)=0\\ \Rightarrow \quad & f=0.\end{aligned}$$

Ainsi $\langle\cdot,\cdot\rangle_G$ étant défini positif, c'est un produit scalaire et G est un espace pré-Hilbertien.

Nous définissons l'espace $\mathbb{G}$ comme la fermeture de G, *i.e.* $\mathbb{G} = \overline{G}$ et nous posons le produit scalaire $\langle \cdot, \cdot \rangle_{\mathbb{G}}$ comme le prolongement par continuité de $\langle \cdot, \cdot \rangle_G$ dans $\mathbb{G}$. Par construction $\mathbb{G}$ étant complet, c'est un espace de Hilbert. Ainsi nous avons, à partir d'une famille de noyau $(k^l)_{1 \leq l \leq m}$, construit un espace de fonction G qui est un espace de Hilbert dont le produit scalaire est $\langle \cdot, \cdot \rangle_{\mathbb{G}}$

3.3.3 Théorème de représentation et *kernel basis*

Soit la norme de fonction $\|\cdot\|_{\mathbb{G}}$ induite par le produit scalaire $\langle \cdot, \cdot \rangle_{\mathbb{G}}$ telle que :

$$\text{pour } f \in \mathbb{G}, \quad \|f\|_{\mathbb{G}} = \sqrt{\langle f, f \rangle_{\mathbb{G}}}. \tag{3.10}$$

Notons R_{emp} le risque empirique (voir équation **??**), relativement à l une fonction de coût quelconque, $D = (x_i, y_i)_{1 \leq i \leq n}$ les données d'apprentissage, Ω la fonction régularisatrice strictement croissante à valeurs réelles et λ le paramètre de régularisation.

Le problème de minimisation régularisé relativement à l'espace de fonctions $\mathbb{G}$ s'écrit de la façon suivante :

$$\min_{f \in \mathbb{G}} J(f) = R_{emp}(D, l, f) + \lambda \Omega(\|f\|_{\mathbb{G}}). \tag{3.11}$$

Alors, l'optimum $f^* = \arg\min_{f \in \mathbb{G}} J(f)$ appartient à l'espace G_D défini comme suit :

$$G_D = \Big\{ \sum_{i=1}^{n} \sum_{l=1}^{m} \alpha_{il} k^l_{x_i} \quad \forall \alpha \in \mathbb{R}^{nm} \Big\}. \tag{3.12}$$

Autrement dit la solution est de la forme *kernel basis* :

$$f^*(\cdot) = \sum_{i=1}^{n} \sum_{l=1}^{m} \alpha_{il} k^l(x_i, \cdot). \tag{3.13}$$

Démonstration. L'ensemble d'apprentissage D étant fini, l'espace G_D est fermé et nous pouvons lui appliquer le théorème du supplémentaire orthogonal :

$$G_D \oplus G_D^{\perp} = \mathbb{G}.$$

Soit $f \in \mathbb{G}$, $\exists (f^{G_D}, f^{G_D^{\perp}}) \in G_D \times G_D^{\perp}$ telles que :

$$f = f^{G_D} + f^{G_D^{\perp}}.$$

Puisque $f^{G_D^{\perp}} \in \mathbb{G}$ donc $\exists (f_l^{G_D^{\perp}})_{1 \leq l \leq m}$ tel que : $f^{G_D^{\perp}} = \sum_{l=1}^{m} f_l^{G_D^{\perp}}$. Notons la fonction g définie

comme l'application du noyau k^l au point x_i : $g = k^l_{x_i}$.

$$\begin{aligned}\left\langle f^{G_D^\perp}, g\right\rangle_{\mathbb{G}} &= \sum_{j=1}^{m} \left\langle f_j^{G_D^\perp}, g_j\right\rangle_{\mathbb{G}_j} \\ &= \left\langle f_l^{G_D^\perp}, k^l_{x_i}\right\rangle_{\mathbb{G}_l} \quad \text{car } \forall j \neq l, g_j = 0 \\ &= f_l^{G_D^\perp}(x_i),\end{aligned}$$

car $\mathbb{G}_l$ est un RKHS.

Par définition on a $g \in G_D$ et $\left\langle f^{G_D^\perp}, g\right\rangle_{\mathbb{G}} = 0$, donc $\forall l \in [1,m]$, $f_l^{G_D^\perp}(x_i) = 0$ donc $f^{G_D^\perp}(x_i) = 0$. Comme le risque empirique R_{emp} ne dépend de f qu'à travers son évaluation sur l'échantillon fini $(x_i)_{1\leq i\leq n}$, la minimisation de R_{emp} est indépendante de $f^{G_D^\perp}$.

Par ailleurs, la fonction Ω étant supposée strictement croissante, elle est minimale quand $f^{G_D^\perp}$ est nul :

$$\Omega(\|f\|_{\mathbb{G}}) = \Omega\Big(\left\|f^{G_D}\right\|_{\mathbb{G}} + \left\|f^{G_D^\perp}\right\|_{\mathbb{G}}\Big) > \Omega\Big(\left\|f^{G_D^\perp}\right\|_{\mathbb{G}}\Big). \tag{3.14}$$

Ainsi l'indépendance de R_{emp} vis à vis de $f^{G_D^\perp}$ et la minimisation de Ω pour $f^{G_D^\perp} = 0$ impliquent que le minimum de (3.13) est atteint en $f^{G_D^\perp}$. Ainsi $f = f^{G_D} \in G_D$.

□

Ainsi la solution du problème d'optimisation dans $\mathbb{G}$ est de la forme *kernel basis*.

Nous avons constaté que l'espace $\mathbb{G}$ est un RKHS car il possède un noyau reproduisant $K = \sum_{l=1}^{m} k^l$. En effet soit $x \in \mathscr{X}$ et $f \in G$ alors :

$$\begin{aligned}\langle f, K(x,\cdot)\rangle_{\mathbb{G}} &= \sum_{l=1}^{m} \left\langle f^l, k^l_x\right\rangle_{\mathbb{G}_l} \\ &= \sum_{l=1}^{m} f^l(x) \quad \text{car } \mathbb{G}_l \text{ est un RKHS} \\ &= f(x).\end{aligned}$$

Ainsi K est un noyau reproduisant et $\mathbb{G}$ est un RKHS.

3.3.4 Discussion

L'espace $\mathbb{G}$ est un RKHS et on peut lui appliquer le théorème de représentation classique (5). On remarque cependant que l'optimum du problème (3.11) n'est pas unique car il est atteint aussi pour des solutions dont la forme est plus simple que celle du *kernel basis* et, en ce

sens, l'intérêt de la pénalisation en norme 2 dans $\mathbb{G}$ est à relativiser. Notons toutefois que cette conclusion dépend de la construction du produit scalaire proposé (3.7) qui nous a semblé être la façon la plus naturelle de construire l'espace associé au *kernel basis*. Mais il peut être possible de trouver d'autres constructions qui permettent de générer un espace $\mathbb{G}$, qui n'est pas un RKHS et dont l'optimum est uniquement atteint pour une solution de type *kernel basis*. En conclusion cette étude théorique nous a conduit à nous intéresser à d'autres types de régularisation pour le *kernel basis*, notamment la pénalité LASSO (L_1) qui engendre de la parcimonie au sein du modèle.

3.4 Kernelisation du DRSVM

Nous expliquons dans cette partie, l'approche que nous avons choisie afin de kerneliser le problème DRSVM sous une configuration *kernel basis*.

3.4.1 Motivation

Dans la section précédente, nous avons proposé une possibilité de construction d'un espace $\mathbb{G}$ pour lequel on peut appliquer un théorème de représentation adapté au modèle *kernel basis*. Cependant notre étude révèle que $\mathbb{G}$ est, par construction, un RKHS car il possède un noyau reproduisant par rapport auquel il est également possible d'appliquer le théorème de représentation classique. En conséquence, il existe une solution plus simple que la solution *kernel basis*, qui atteint le même optimum. En ce sens la pénalisation en norme 2 via les RKHS ne semblant pas être la manière la plus pertinente d'introduire le *kernel basis*, nous nous sommes intéressés à la pénalisation LASSO. Remarquons tout d'abord que le modèle *kernel basis* est un modèle de dimension étendue. En effet le cardinal des variables intégrées dans un même noyau est de taille $m \times n$, où m désigne le nombre de noyaux et n le nombre de points d'apprentissage. Pour cette raison, une solution parcimonieuse est pertinente afin d'obtenir un modèle simple et interprétable. D'un point de vue théorique, la norme 1 n'étant pas une norme euclidienne, il est beaucoup plus difficile de dériver un théorème de représentation.

Notons néanmoins qu'une approche récente se propose de donner un cadre théorique pour la pénalisation LASSO dans le cas de la régression quantile kernelisée [Shi et al., 2014]. Aussi une généralisation intéressante du théorème de représentation pour la norme 1, qui remplace l'espace de Hilbert par un espace de Banach pour certains types de noyaux a été proposée par Song et al. [2013]. Toutefois la définition du noyau doit vérifier un certains nombre de propriétés qu'il semble difficile de vérifier dans la pratique. C'est pourquoi nous nous sommes orientés vers une autre approche dite *en base* ou *dictionnaire*. Cette dernière a été introduite dans le cadre de la kernelisation de la régression logistique [Zhu and Hastie, 2001]. L'idée consiste à

choisir a priori une solution de la forme :

$$f(\cdot) = \sum_{i \in V} \alpha_i K(x_i, \cdot) \quad \text{avec} \quad V \subset [1,n].$$

La solution se décompose sur V, un sous-ensemble des points d'apprentissage. Remarquons que cette méthode peut être rapprochée aux techniques d'ondelettes [Mallat, 1989] issues du traitement du signal. Nous avons insisté dans la section 3.2 sur le potentiel du *kernel basis* model (3.5) à générer des modèles interprétables, sous réserve qu'un processus de sélection de modèle adapté soit intégré dans le mécanisme d'apprentissage. Dans un souci d'éviter l'exclusion de variables pertinentes de la solution, nous avons opté pour une pénalité du type *elastic net* plutôt que LASSO, afin de construire un classifieur interprétable. Ces différentes exigences nous ont conduit à proposer une adaptation du problème DRSVM au cadre noyau. La présence du terme de pénalisation L_1 ne permet pas une kernelisation directe. Ainsi nous avons choisi une approche en base, dont le dictionnaire est généré à partir d'une famille de noyaux, dans le but de fusionner le problème DRSVM avec le modèle *kernel basis*.

3.4.2 DRSVM et approche *kernel basis*

L'algorithme DRSVM a été à l'origine proposé dans le cadre de modèles linéaires. Nous expliquons ici comment nous l'étendons au modèle *kernel basis*. Un moyen classique pour introduire la kernelisation consiste à utiliser la méthode du kernel trick mais qui ne peut être appliquée au problème DRSVM du au terme de régularisation L_1. Une manière alternative d'introduire la kernelisation, est d'appréhender les noyaux comme de nouvelles variables d'étude qui sont intégrées au sein d'un modèle global linéaire. Ce modèle peut alors être interprété comme une combinaison linéaire de fonctions noyaux. L'approche dictionnaire est une façon d'introduire les noyaux en construisant un dictionnaire D adapté à la kernelisation. Formellement un dictionnaire $D = (\phi_l)_{1 \leq l \leq p}$ peut se définir comme une collection de p fonctions ϕ définies sur $\mathscr{X}$ à valeur réelles. Le problème DRSVM adapté à l'approche dictionnaire s'écrit de la manière suivante :

$$\begin{cases} \min\limits_{\beta_0, f} & \sum\limits_{i=1}^{n} \max[0, 1 - y_i(\beta_0 + f(x_i))] + \dfrac{\lambda_2}{2}\Omega_2(f) + \lambda_1 \Omega_1(f) \\ \text{avec} & f(x) = \sum\limits_{l=1}^{p} \beta_l \phi(x), \quad \Omega_2(f) = \sum\limits_{l=1}^{p} \beta_l^2 \quad \text{et} \quad \Omega_1(f) = \sum\limits_{l=1}^{p} |\beta_l|. \end{cases}$$

Le choix $D = k(x_j, \cdot)_{1 \leq j \leq n}$ avec k un noyau et $(x_j)_{1 \leq j \leq n}$ les points d'apprentissage, correspond à une approche mono-noyau du problème DRSVM. Notons que si on considère $k = \sum\limits_{l=1}^{m} \beta_l k_l$ une somme pondérée de m noyaux, on adopte alors une approche MKL.

Pour introduire le modèle *kernel basis*, il est nécessaire d'utiliser un autre dictionnaire D' construit à partir d'une famille $(k_l)_{1\leq l\leq m}$ de m noyaux :

$$D' = (\psi_{jl})_{1\leq l\leq m}^{1\leq j\leq n} \text{ avec } \psi_{jl}(x) = k_l(x_j,x) \quad , \quad f(x) = \sum_{j=1}^{n}\sum_{l=1}^{m} B_{jl}k_l(x_j,x) \quad \text{et} \quad B \in \mathbb{R}^{nm}.$$

Le modèle issu du problème d'optimisation DRSVM étendu à l'architecture *kernel basis* par le biais du dictionnaire D' de taille $p = mn$ est appelé KB_DRSVM et prend la forme suivante :

$$\min_{\beta_0, B\in\mathbb{R}^{m\times n}} \sum_{i=1}^{n} \max\Big[0, 1 - y_i\big(\beta_0 + \sum_{j,l=1}^{n,m} B_{jl}k_l(x_j,x_i)\big)\Big] + \frac{\lambda_2}{2}\sum_{j,l=1}^{n,m} B_{jl}^2 + \lambda_1 \sum_{j,l=1}^{n,m} |B_{jl}|. \qquad (3.15)$$

Remarquons que l'on peut envisager la construction de D' à partir d'un famille complexe amalgamant des noyaux hétérogènes tels que des noyaux gaussiens avec des largeurs de bandes croissantes, des noyaux polynomiaux ou des noyaux d'Epanechnikov.

3.4.3 Chemin de régularisation et *kernel basis*

3.4.3.1 Résolution du problème *kernel basis*

Une fois que l'on a défini le dictionnaire D' correspondant au modèle *kernel basis*, on peut appliquer l'algorithme linéaire DRSVM sur les données transformées, via l'approche en base, et dériver un chemin de régularisation pour le modèle *kernel basis*. Pendant la résolution, nous adoptons la même stratégie pour l'initialisation et la procédure générale que dans le cas linéaire. L'algorithme de résolution est donc le même que celui présenté dans le chapitre précédent.

3.4.3.2 Notations et analyse du modèle

Après avoir achevé la construction du chemin, on reconstruit la matrice B et on peut étudier la valeur de ces coefficients afin d'analyser les interaction points / noyaux pertinents qui ont émergé à l'issue de la phase d'apprentissage.

L'extension du problème DRSVM à l'approche *kernel basis* peut générer une certaine confusion dans l'analyse du modèle en raison de la double dépendance dans la solution entre les variables kernalisées (les nouvelles variables d'étude) et les points d'observations associés (données initiales). Il est donc utile d'introduire des notations afin de lever les ambiguïtés et de décrire précisément la solution optimale. En particulier, il existe des situations pour lesquelles la signification des coefficients du modèle peut être contre-intuitive en comparaison de l'approche SVM. En effet, il est possible de rencontrer des cas où certains points x_i dont la composante de la sous-différentielle associée $\alpha_i = 0$ et tels que $B_{il} \neq 0$. Autrement dit, cela signifie qu'un point

x_i respectant la contrainte sur la marge (c'est à dire tel que $i \in \mathscr{L}$) et qui n'est donc par définition pas un vecteur support, peut avoir néanmoins une influence au sein du modèle final. Cela est susceptible de se produire lorsque des noyaux radiaux sont intégrés au sein de la famille de noyaux, car le KB_DRSVM (3.15) a tendance à sélectionner les points proches du centroïde de chacune des classes.

Nous conservons le nom de *vecteur support* pour désigner les points x_i situés sur la marge (c'est à dire pour $r_i \geq 0$ et $\alpha_i > 0$). Mais nous appelons *point actif* tout point x_i pour lequel il existe un noyau k^l tel que le coefficient $B_{jl} \neq 0$. En ce cas, le couple (i,l) est appelé *couple actif* (voir définition 8). L'ensemble des couples actifs modélise l'ensemble des synergies entre les points et les noyaux. Inversement, un couple (i,l) est dit *inactif* quand $B_{jl} \neq 0$ et qu'il n'y a pas de connexion entre le point x_i et le noyau k_l au sein de la solution.

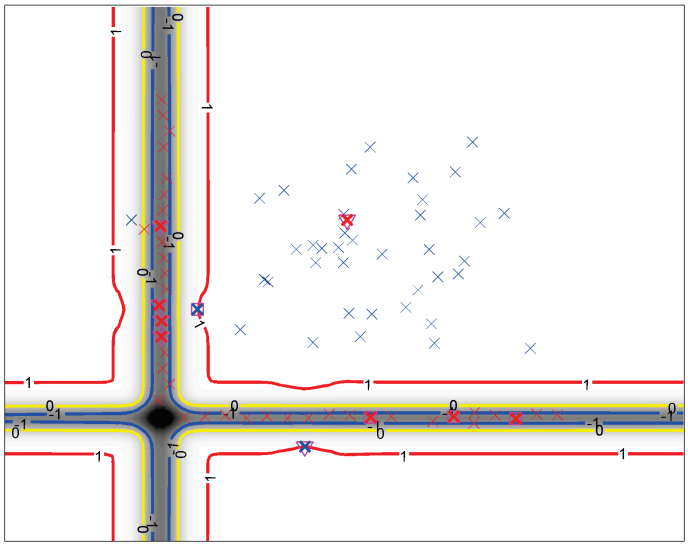


FIGURE 3.2 : Nous affichons la frontière de décision d'un modèle KB_DRSVM et avons entouré par des symboles les points actifs. Ce modèle peut être décrit par la données des points actifs (les points qui ont une influence dans la solution) et des points supports (les points situés sur la marge).

La figure 3.2 présente une illustration de visualisation d'un modèle *kernel basis* en 2D. On constate que le modèle est relativement parcimonieux dans la mesure où la grande majorité des points respectant strictement la contrainte sur la marge (ils ne sont pas vecteurs supports) ne sont pas des points actifs et donc ne participent pas au modèle. Mais parmi les points non vecteurs supports, il y en a néanmoins quelques-uns, situés aux centres de régions à forte densité, qui sont aussi des points actifs. Inversement, au sein de l'ensemble des points supports, situés à la périphérie, on a à la fois des points actifs et des points inactifs.

3.5 Expérimentations pour le *kernel basis*

Dans cette section, nous présentons des expérimentations menées à la fois sur des données synthétiques et des données réelles, afin de valider notre approche KB_DRSVM (3.15). En premier lieu, nous décrivons et analysons la nature de notre modèle à travers plusieurs jeux de données jouets. Enfin, nous présentons une phase d'expérimentation sur des données images, afin de montrer l'intérêt de l'approche pour l'interprétabilité du *kernel basis* sur des données réelles.

3.5.1 Analyse du comportement du modèle KB_DRSVM

La première expérimentation sur un jeu de données simulées vise à mettre en évidence l'influence des paramètres de régularisation vis-à-vis de la solution générée.

3.5.1.1 Description des données jouets : Toydata 1

Le jeu de données Toydata 1 est un problème de classification binaire, que nous avons généré afin d'illustrer la capacité KB_DRSVM à produire des couples actifs pertinents, dans l'hypothèse où la famille de noyaux utilisée pendant l'apprentissage $(k^l)_{1\leq l\leq m}$ est adaptée au problème. Le jeu de données Toydata 1 possède une topologie locale spécifique au sein de chaque classe et nous désirons obtenir un modèle *kernel basis*, dont la forme capture la nature de ce phénomène. Nous avons choisi un exemple en 2 dimensions, afin de pouvoir superposer directement la solution *kernel basis* avec les données d'apprentissage et d'analyser la forme du modèle.

Description du jeu de données : les données se composent de deux classes équilibrées. La classe « plus » est générée à partir d'une distribution gaussienne bidimensionnelle. La classe « moins » est l'union de deux bandes, qui sont générées selon des distributions gaussiennes unidimensionnelles (voir figure 3.3).

Famille de noyaux : le modèle *kernel basis* est généré à partir de $\mathscr{K} = (k^1, k^2, k^3)$, une famille de 3 noyaux gaussiens : k_1 un noyau gaussien isotrope, k_2 un noyau gaussien vertical (largeur de bande verticale élevée et une faible largeur de bande horizontale), k_3 un noyau gaussien horizontal (largeur de bande horizontale élevée et une faible largeur de bande verticale). La valeur du paramètre de largeur de bande σ associé est choisie de façon à décrire les 3 parties homogènes qui composent le Toydata 1. Le noyau k^1 est adapté à la classe « plus » et les noyaux k^2 et k^3 sont adaptés respectivement à la bande verticale et la bande horizontale.

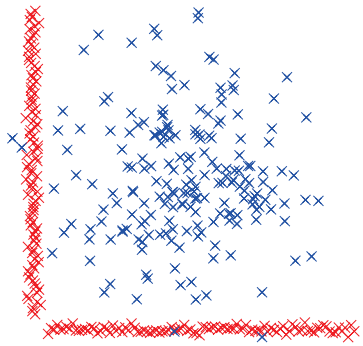


FIGURE 3.3 : Le jeu de données Toydata 1 bidimensionnel est composé de deux classes : la classe « plus » (affichée en bleu) est générée à partir d'une loi normale 2D et la classe « moins » (affichée en rouge) est générée par la réunion de 2 lois normales unidimensionnelles.

3.5.1.2 Différentes dynamiques de chemin

Avant d'analyser la nature de la solution du KB_DRSVM, nous souhaitons discuter de la relation entre la forme du chemin de régularisation et la valeur du paramètre λ_2 qui conditionne l'ordre d'apparition des événements. Au cours de la phase d'expérimentation, nous avons constaté que l'on peut globalement discerner 3 types de dynamiques de chemin :

1. si la valeur $\lambda_2 < \lambda_2^{\min} \approx 0$ le chemin obtenu est équivalent à un chemin de régularisation en norme 1,
2. si la valeur $\lambda_2 \in [\lambda_2^{\min}, \lambda_2^{\max}]$ une partie des solutions du chemin sont pénalisées en *elastic-net*,
3. si la valeur $\lambda_2 > \lambda_2^{\max}$ la forme du chemin est identique $\forall \lambda_2$ en termes d'événements.

La différence de comportement peut être observée à travers la visualisation des cardinaux associés aux ensembles de points ($\mathscr{R}$,$\mathscr{E}$ et $\mathscr{L}$) et de l'ensemble de variables $\mathscr{V}_\beta$ (voir figure 3.4).

Cas 1 : nous avons remarqué empiriquement qu'il existe une valeur $\lambda_2^{\min}$ proche de zéro tel que $\forall \lambda_2 < \lambda_2^{\min}$, les courbes des cardinaux évoluent de manière identique ainsi que la valeur des coefficients de la solution. Nous expliquons ce phénomène par le fait que le problème DRSVM devient dans cette situation extrêmement proche du problème du SVM L_1. Ainsi le chemin devient équivalent à un chemin de type LASSO.

Cas 2 : de façon similaire, il existe une valeur $\lambda_2^{\max}$ tel que $\forall \lambda_2 > \lambda_2^{\max}$, les courbes des cardinaux restent identiques. Mais plus on augmente la valeur du paramètre λ_2, plus la valeur, en moyenne sur le chemin, des coefficients de la solution est faible. Dans l'hypothèse où les classes sont équilibrées, il est possible de calculer explicitement la valeur λ_2^{max}. En effet, pour une valeur suffisamment grande de λ_2, on obtient un chemin pour lequel toutes les variables sont sélectionnées en priorité par rapport aux autres événements, jusqu'à ce que $\lambda_1 = 0$. La valeur λ_2^{max} correspond au cas critique, dans lequel deux événements surviennent de manière simultanée à la dernière itération : un point x_i transite dans $\mathscr{E}$ (c'est à dire $r_i = 0$) et $\lambda_1 = 0$. Afin de calculer cette valeur nous nous inspirons de la discussion à propos de l'influence de λ_2 sur le choix des événements (voir la figure 2.10 du chapitre précédent). Soit la variable C_j définie comme suit :

$$\forall j \in [1,p], \quad C_j = |\sum_{i=1}^{n} y_i x_{ij}|.$$

Supposons l'ordre des variables triées telles que : $C_1 > \cdots > C_j > \cdots > C_p$ alors on peut exprimer la valeur de λ_2^{max} de la manière suivante :

$$\lambda_2^{max} = \max_{i\in[1,n]} \Big(-y_i \times \textstyle\sum_{j=1}^{p} sign(\beta_j) x_{ij} \Big) + R_p$$

$$\text{avec} \quad \begin{cases} R_0 = 0 \\ R_k = R_{k-1} + (C_k - C_{k+1}) \times y_i \displaystyle\sum_{j=1}^{k} sign(\beta_j) x_{ij}, \quad \forall k \in [1,p]. \end{cases}$$

Le calcul s'inspire de la discussion à propos de l'influence de λ_2 sur le choix de l'événement à la première itération (voir figure 2.10). Dans le cas où l'on sélectionne uniquement des variables, il est facile d'exprimer la valeur du pas $\delta\lambda_1$ ainsi que les dérivées des différents paramètres à chaque itération et ainsi on peut calculer la valeur de λ_2^{max} pour laquelle le dernier événement correspond à l'annulation de λ_1 simultanée avec la transition d'un point dans l'ensemble $\mathscr{E}$.

Cas 3 : soit $\lambda_2 \in [\lambda_2^{\min}, \lambda_2^{\max}]$, alors les solutions du chemin sont pénalisées en *elastic-net*. Notons que la valeur de λ_2 est influente sur la longueur de la première partie du chemin, où la pénalisation sur la norme 1 est prépondérante par rapport à la pénalisation sur la norme 2.

3.5.1.3 Modèle en fin de chemin

Le chemin de régularisation parcourt l'ensemble des paramètres λ_1 en commençant depuis une valeur infinie (solution nulle) et s'arrête quand il n'y plus de pénalisation sur la norme 1, c'est-à-dire quand $\lambda_1 = 0$. En fin de chemin, le problème DRSVM (2.2) est uniquement régularisé en norme 2 par le biais du paramètre λ_2 et la solution est équivalente à une solution de type SVM. Rappelons que λ_2 a une influence inversement proportionnelle au paramètre

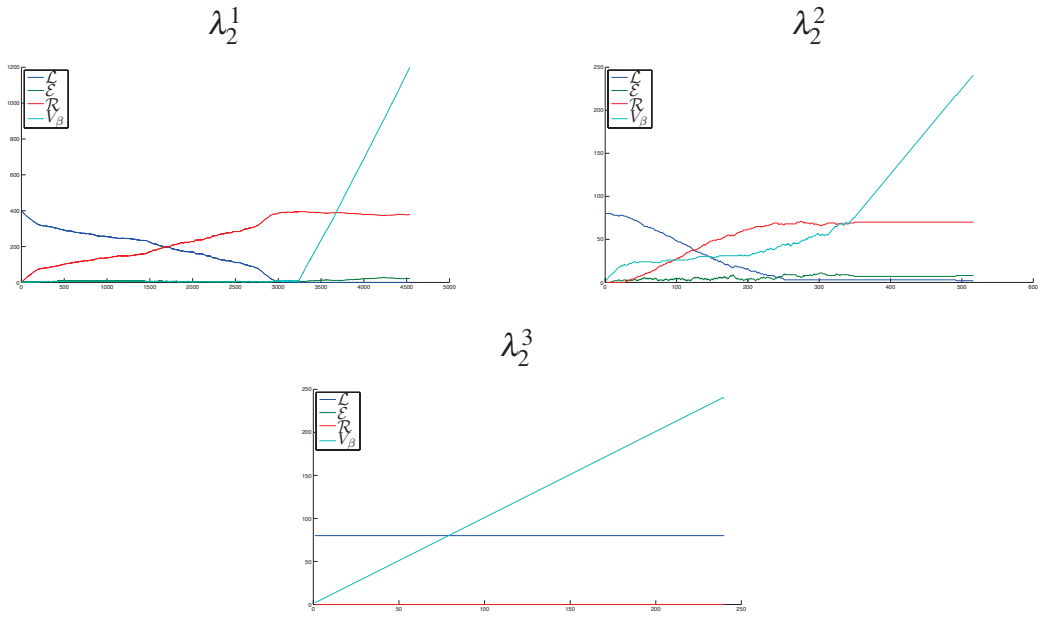


FIGURE 3.4 : La figure illustre la relation entre la forme du chemin et la valeur de λ_2. Nous avons affiché en fonction du nombre d'itérations, l'évolution des cardinaux des ensembles $\mathscr{R}$, $\mathscr{E}$, $\mathscr{L}$, $\mathscr{V}_\beta$ pour trois valeurs différentes de λ_2 : $\lambda_2^1 = 10^{-20}$ (figure en haut à gauche), $\lambda_2^2 = 1$ (figure en haut à droite) et $\lambda_2^3 = \lambda_2^{max}$ (figure en bas).

C du SVM classique. Notons que, indépendamment de la valeur de λ_2, la solution n'est pas parcimonieuse car $\lambda_1 = 0$ et donc toutes les variables sont actives à la dernière itération. Afin d'illustrer le comportement en fin de chemin, nous avons affiché sur la figure 3.5, la dernière solution calculée pour trois valeurs de λ_2 :

1. $\lambda_2^1 = 10^{-20}$: la valeur étant très faible, nous observons un phénomène de sur-apprentissage,
2. $\lambda_2^2 = 1$: nous obtenons une solution satisfaisante dont la frontière est régulière et sépare bien les données,
3. $\lambda_2^3 = \lambda_2^{max}$: la solution est fortement pénalisée, la frontière n'est pas très souple mais sépare bien les données.

3.5.1.4 Interprétabilité de la solution

La famille de noyaux $\mathscr{K}$ étant construite par hypothèse de manière à prendre en compte la topologie locale du jeu de données Toydata 1, nous souhaitons vérifier que la solution KB_DRSVM génère des couples actifs pertinents, c'est-à-dire tels que les coefficient $B_{il} \neq 0$

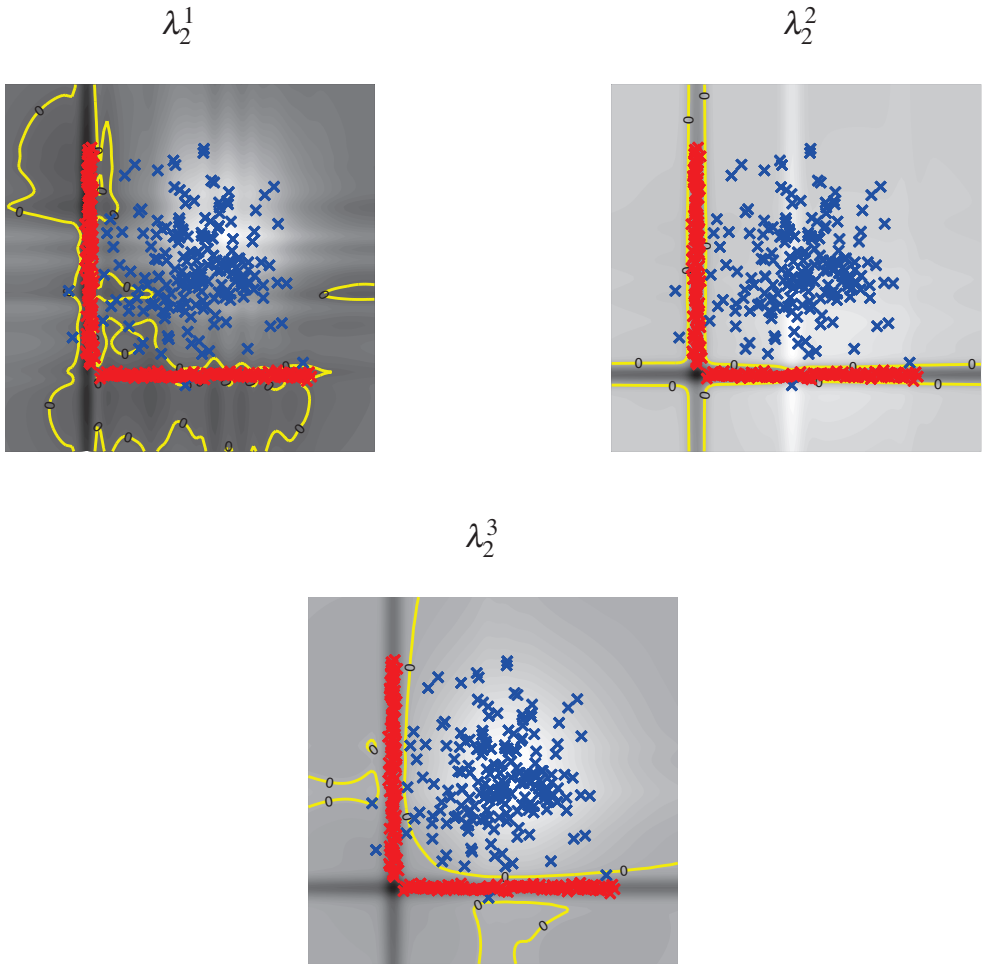


FIGURE 3.5 : Nous avons affiché la frontière de décision pour la dernière solution ($\lambda_1 = 0$), pour trois chemins correspondant à des valeurs différentes de λ_2 : 10^{-20} (figure en haut à gauche), 1 (figure en haut à droite) et λ_2^{max} (figure du bas). Ce modèle n'étant régularisé qu'en norme 2, la valeur du paramètre λ_2 a une grande influence sur la forme de la frontière de décision représentée en jaune.

soient associés à des points x_i appartenant à la zone locale pour laquelle le noyau k a été adapté. La simulation associée à la figure 3.6 a été effectuée avec une valeur $\lambda_2 = 10^{-10}$ afin de générer un chemin dont les solutions sont en moyenne parcimonieuses. La solution affichée correspond à la solution optimale de λ_1 sur le chemin de régularisation obtenue par validation croisée sur un ensemble de données test. L'analyse de la solution montre que le modèle peut être décrit uniquement à partir de 3 couples actifs $c = (c_1, c_2, c_3) : (c_1)$ une connexion entre un point situé à proximité du centre de la classe « plus » et k^1 (noyau gaussien isotrope), (c_2) une connexion entre un point proche au milieu de la bande verticale et k^2 (noyau gaussien avec une très grande largeur de bande verticale et très faible largeur de bande horizontale) et (c_3) une connexion entre un point proche du milieu de la bande horizontale et k^3 (noyau gaussien avec une très grande largeur de bande horizontale et très faible largeur de bande verticale). Comme les noyaux utilisés sont gaussiens, les points actifs sont très éloignés de la frontière et ne sont pas vecteurs supports.

Influence de λ_2 : nous avons effectué la même simulation, mais cette fois-ci pour $\lambda_2 = 1$. Le chemin de régularisation est globalement moins parcimonieux et sélectionne rapidement davantage de variables discriminantes, même si elles sont corrélées entre elles. Ainsi, la solution optimale sur le chemin est moins parcimonieuse que le modèle de la figure 3.6. On observe que les couples actifs restent pertinents comme nous pouvons le remarquer sur la figure 3.7. En effet, il y a plus de points actifs sur les bandes gaussiennes, mais le noyau actif associé est bien celui adapté à chaque bande.

Les résultats sur le jeu de donné Toydata 1 illustrent la capacité du modèle KB_DRSVM à réaliser une tâche de classification tout en générant simultanément des couples actifs cohérents avec la topologie sous-jacente locale des données.

3.5.2 Influence de la norme L_1 sur la forme du modèle

Le jeu de données Toydata 2 (voir figure 3.8) est une variante de Toydata 1, que nous avons généré pour mettre en évidence la nécessité de tempérer l'influence de λ_1 afin de préserver la topologie locale. Choisir une valeur de λ_1 élevée, peut conduire à sélectionner un modèle qui sépare bien les données, mais dont les couples actifs ne sont pas les plus pertinents, dans le sens où ils ne correspondent pas à la topologie locale des données.

Description du jeu de données : le jeu de données Toydata 2 est généré similairement à Toydata 1, à l'exception des bandes verticales et horizontales qui sont générées différemment. Les deux bandes de la classe « moins » sont un mélange de deux distributions gaussiennes. De plus nous avons ajouté des points de la classe « plus » (points de perturbation) au milieu de la distribution « plus » verticale pour créer un zone de confusion.

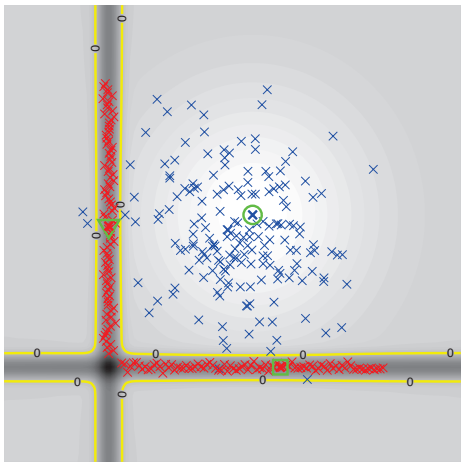


FIGURE 3.6 : Nous affichons la solution KB_DRSVM pour $\lambda_2 = 10^{-10}$ et pour la valeur optimale de λ_1 sur le chemin (obtenue par validation croisée). La frontière de décision est représentée en jaune. Pour chaque couple actif le point actif est représenté par un symbole vert dont la forme indique le noyau utilisé : un cercle pour k^1, un triangle pour k^2 et un carré pour k^3.

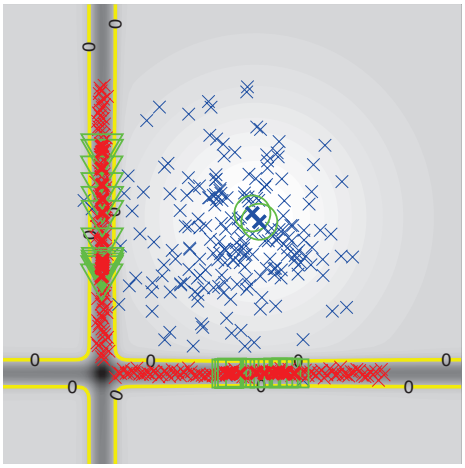


FIGURE 3.7 : Nous affichons la solution KB_DRSVM pour $\lambda_2 = 1$ et pour la valeur optimale de λ_1 sur le chemin (obtenue par validation croisée). La frontière de décision est représentée en jaune. Pour chaque couple actif le point actif est représenté par un symbole vert dont la forme indique le noyau utilisé : un cercle pour k^1, un triangle pour k^2 et un carré pour k^3. Nous observons que la solution sélectionne davantage de couples actifs pertinents que le modèle de la figure 3.6.

Famille noyau : la famille du noyau $\mathscr{K}$ est composée de cinq noyaux bidimensionnels (voir figure 3.9) : k_1 un noyau gaussien isotrope, k_2 un noyau gaussien vertical (largeur de bande

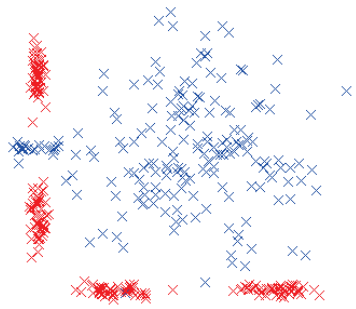


FIGURE 3.8 : Le jeu de données Toydata 2 est composé de deux classes comme Toydata 1, avec une zone de confusion : la classe « plus » (croix bleues) est générée à partir d'une loi normale 2D **et d'une bande gaussienne verticale** représentant une forme d'interaction. La classe « moins » (croix rouges) est générée par la réunion de 2 gaussiennes, générées chacune à partir de deux gaussiennes 1D discontinues horizontales et verticales.

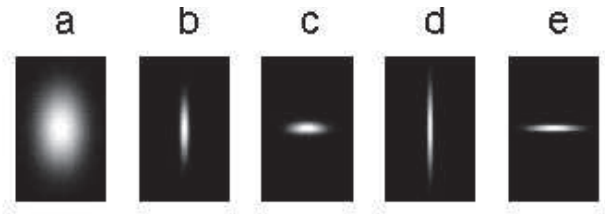


FIGURE 3.9 : Représentation de l'influence des noyaux : (a) k_1, (b) k_2, (c) k_3, (d) k_4 et (e) k_5. Nous avons appliqué les noyaux par rapport au centre et à un pavage de l'espace des données, nous affichons la valeur du noyau tel que des valeurs élevées et basses correspondent respectivement à des pixels clairs et sombres.

verticale élevée et une faible largeur de bande horizontale), k_3 un noyau gaussien horizontal (largeur de bande horizontale élevée et une faible largeur de bande verticale), k_4 un autre noyau gaussien vertical (largeur de bande verticale très élevée et très faible largeur de bande horizontale) et k_5 un autre noyau gaussien horizontal (largeur de bande horizontale très élevée et une très faible largeur de bande verticale).

3.5.2.1 Simulation

Les deux images de la figure 3.10 représentent la frontière de décision de deux modèles KB_DRSVM obtenus pour des valeurs différentes de λ_1 (mais avec la même valeur $\lambda_2 = 1$). Sur la figure de gauche, le modèle KB_DRSVM ne parvient pas à sélectionner les noyaux (k_2, k_4) afin de décrire la sous-distribution, et choisit à la place (k_3, k_5) dont les points actifs

associés se situent au milieu de chaque bande. Cette mauvaise sélection de couple actif est dû à un mauvais choix de λ_1 dont la trop grande valeur empêche l'activation de couples actifs associés à k_1 ou k_3. Mais lorsque que le paramètre λ_1 atteint une certaine valeur critique, les noyaux (k_2, k_4) sont alors intégrés dans la solution. De plus, nous observons que la valeur des couples actifs (k_3, k_5) commence à diminuer jusqu'à leur désélection. Cela conduit au modèle 3.10 (figure de droite), dont les couples actifs sont mieux adaptés aux données. En effet, nous pouvons observer qu'il y a des couples actifs qui associent k_3 aux points de perturbation issus de la classe « plus », k_2 à des points issus de chaque bande verticale (voir le zoom 3.11 sur la figure de droite de l'image 3.10). Les couples actifs restants associent k_1 à la classe « plus » et k_2 à chaque bande latérale.

A travers le jeu de données Toydata 2 nous avons illustré la nécessité de tempérer l'influence de la régularisation L_1 afin de préserver les topologies locales fines. Si la pénalisation est trop importante, il est possible d'obtenir un modèle qui a un taux de bonne classification satisfaisant, mais qui n'intègre pas les couples les plus pertinents. Cela s'explique par le fait que les couples d'intérêt n'ont pas toujours le meilleur pouvoir de discrimination parmi l'ensemble total des couples et fonctionnent par synergie avec d'autres.

3.5.3 KB_DRSVM et robustesse au bruit

3.5.3.1 Description des données checker

Il est possible d'introduire un a priori expert sur des données réelles et de construire une famille de noyaux $\mathscr{K}$ adaptée aux données. Néanmoins, il est possible que la famille soit imparfaite et contienne des éléments non adaptés aux données d'apprentissage. Dans cette configuration, nous souhaitons obtenir une solution capable de rejeter les noyaux pertinents, dont les variables kernelisées associées peuvent être assimilées à du bruit. Dans cette optique, l'influence de la pénalisation L_1 (au sein du terme de pénalisation elastic-net) est particulièrement utile de part sa grande robustesse au bruit.

Description du jeu de données : les données checker que nous générons appartiennent à $\mathbb{R}^4$: les deux premières dimensions sont un damier, c'est-à-dire un ensemble de 16 carrés adjacents générés chacun à partir d'une distribution uniforme bidimensionnelle (voir figure 3.12.a). Les deux dernières dimensions sont des variables de bruit générées selon une distribution gaussienne bidimensionnelle.

$\lambda_1 = 10^{-1}$ $\lambda_1 = 1$

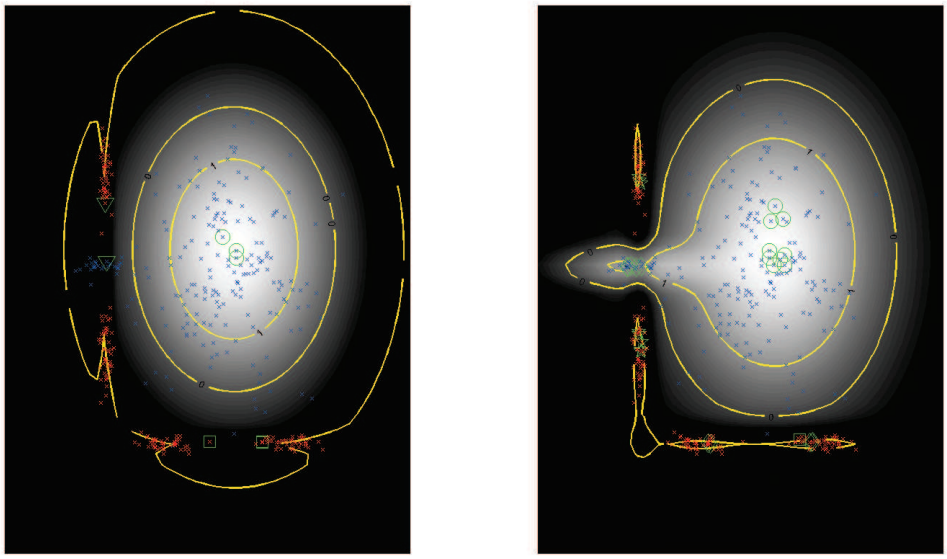


FIGURE 3.10 : Nous avons affiché deux solutions KB_DRSVM sur le jeu de données Toy-data 2 pour deux valeurs de λ_1 et avec $\lambda_2 = 1$. La frontière et les marges sont représentées en jaune. Pour chaque couple actif, le point actif associé est représenté par un symbole vert dont la forme indique le noyau utilisé : un cercle pour k^1, une étoile pour k^2 et un diamant pour k^3, un triangle pour k_4 et un carré pour k_5. La figure de gauche correspond à un modèle qui présente un de taux de classification satisfaisant mais qui ne parvient pas à sélectionner assez finement les noyaux pour prendre en compte la topologie locale des données. Le modèle de droite réussi à créer des couples actifs pertinents qui sont adaptés aux données ($\lambda_1 = 1$) mais est moins parcimonieux que le modèle de gauche ($\lambda_1 = 10^{-1}$).

Famille de noyaux : la famille de noyaux est construite à partir d'un noyau gaussien k que l'on projette sur toutes les combinaisons de deux dimensions. Ainsi on génère une famille $\mathscr{K}$ de 6 noyaux : 1 noyau pertinent k_r (la projection de k sur les deux premières dimensions) et 5 noyaux non pertinents car ils incluent au moins une dimension de bruit et sont donc inadaptés pour discriminer les données checker.

Simulation : afin d'évaluer l'influence de la pénalisation elastic-net sur la capacité de la solution *kernel basis* à rejeter les noyaux non-discriminants, nous avons effectué plusieurs simulations correspondant à des choix de paramètre λ_2 différents. Nous avons ensuite sélectionné le modèle associé à la valeur optimale de λ_1 sur le chemin et relevé pour chaque modèle le nombre de couples actifs pertinents (associés au noyau k_r).

Afin de visualiser la forme de la solution KB_DRSVM, nous projetons les données sur les

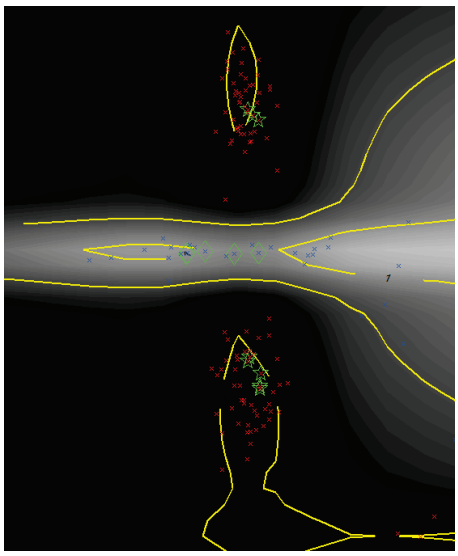


FIGURE 3.11 : Zoom sur la bande verticale de la figure 3.10 (droite).

deux premières dimensions (variables d'intérêt) et affichons sur la figure 3.12 les 3 modèles KB_DRSVM correspondant à des valeurs différentes de λ_2. La frontière de décision est affichée en jaune, les points actifs sont repérés par des cercles jaunes. On observe que la régularisation L_1 a tendance à générer un seul couple actif pour chaque case du damier, et qui est situé à proximité de son centre. La pénalisation L_2 tempère ce phénomène en sélectionnant progressivement les points en direction de la périphérie de chaque case. Pour chaque simulation, le nombre de variables pertinentes et non-pertinentes est reporté en tableau 3.1. Nous constatons que le KB_DRSVM tend à sélectionner les variables pertinentes qui sont corrélées et à les rejeter. Puis, quand la valeur du paramètre λ_2 est trop élevée, toutes les variables sont activées (même celles non-pertinentes) ce qui conduit à une légère diminution du taux de bonne classification en test.

λ_2	10^{-2}	10^{-1}	1
Nb de couples actifs (noyau pertinent (k_r)) :	21	69	193
Nb de couples actifs (noyau non pertinent) :	21	38	127
λ_1	1.87	1.48	0.90
taux de bonne classification en test :	94%	95%	94%

TABLE 3.1 : Résultats sur le jeu de données checker, de trois modèles KB_DRSVM optimaux sur le chemin, pour trois valeurs différentes de $\lambda_2 = \{10^{-2}, 10^{-1}, 1\}$. Afin d'évaluer la capacité de notre modèle à ne pas trop sélectionner de mauvais noyaux, nous affichons la proportion des couples actifs associés respectivement aux deux types de noyaux : pertinent et non pertinent.

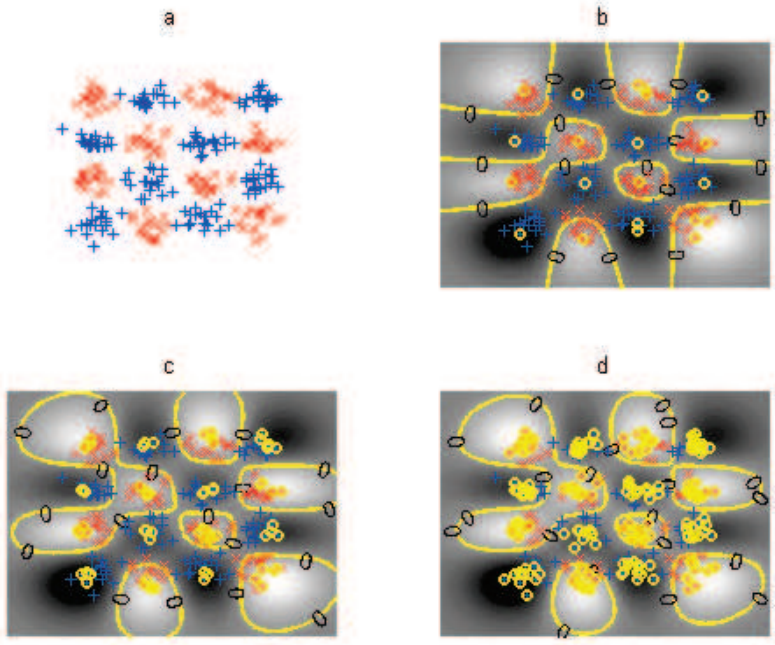


FIGURE 3.12 : Les données checker, projetées sur les 2 variables pertinentes (damier), sont affichées sur la figure (a). Puis nous affichons la frontière de décision restreinte aux 2 variables pertinentes de 3 modèles optimaux (λ_1 est optimisé sur le chemin de régularisation), pour des valeurs différentes de λ_2 : (b) $\lambda_2 = 10^{-2}$, (c) $\lambda_2 = 10^{-1}$ et (d) $\lambda_2 = 1$. Nous observons qu'une augmentation de la valeur de λ_2 conduit à sélectionner davantage de points périphériques en tant que points actifs.

3.5.4 Application à des données images

Pour compléter les expérimentations effectuées sur des données synthétiques, la dernière simulation concerne des données réelles issues d'une base d'images.

3.5.4.1 Description des données

La base de données MNIST [1] (Mixed National Institute of Standards and Technology) est une base de données d'images en libre accès et très utilisée au sein de la communauté Machine Learning. Elle se compose d'une base de données d'apprentissage de 60000 images et d'une base de test de 10000 images. Les images de taille de 28×28 pixels correspondent à des chiffres (0-9) tracés par différents utilisateurs. Les données MNIST définissent un problème de classification à 10 classes où l'objectif consiste à prédire pour une nouvelle image, le chiffre tracé par l'utilisateur.

1. `http://yann.lecun.com/exdb/mnist/`

3.5.4.2 Motivation du choix de la base de données

Le choix de la base d'images MNIST pour démontrer la capacité de notre modèle à construire un classifieur interprétable a été motivé par plusieurs raisons :

- Nous avons conjecturé que les images de la base, de part la nature de leur variabilité spatiale, sont adéquates avec notre volonté d'adapter localement les noyaux aux données. En outre, il existe plusieurs manières d'écrire certains chiffres et il peut être intéressant de rechercher un modèle *kernel basis* qui met en exergue ce phénomène.
- Nous avons jusqu'à présent étudié des exemples où la représentation de la frontière de décision en 2 dimensions était possible, ce qui permet d'interpréter le modèle via l'analyse de ses couples actifs. Mais, dans la situation où la dimension des données est supérieure, il devient compliqué, voire impossible, d'afficher la frontière de décision, et de mettre en relation directe les données d'apprentissage et les couples actifs de la solution. Afin de contourner cette difficulté, nous avons choisi d'utiliser les images, car leur visualisation offre une possibilité de représentation.
- Nous pouvons construire une famille de noyaux adaptée aux données, en jouant nous-mêmes le rôle d'expert et d'inclure notre a priori sur la définition des noyaux.

3.5.4.3 Choix de la famille de noyaux

Il existe de nombreuses possibilités de construction des noyaux spécifiques sur les images, nous pouvons construire des noyaux sophistiqués basés sur des caractéristiques d'image, telles que les contours. Nous avons choisi une approche permettant d'avoir une représentation pertinente des couples actifs pour pouvoir analyser efficacement la forme de la solution. Pour cela nous choisissons d'adopter une approche spatiale directe en balayant chaque image I avec une fenêtre W de largeur w et la longueur h, translatée (verticalement et horizontalement) par un pas de t pixels (voir la figure 3.13). Puis, nous construisons la famille du noyau en associant un noyau polynomial unique à chaque fenêtre. Un couple actif peut être assimilé à un lien entre une image I et une fenêtre W. Notons que l'utilisation d'un noyau polynomial, plutôt qu'un noyau radial, est liée à la méthode de représentation des couples actifs que l'on a adoptée pour l'interprétation, comme nous allons le préciser dans le paragraphe suivant.

Représentation des couples actifs : une fois la phase d'apprentissage réalisée, nous pouvons visualiser les couples actifs en tant que superposition de I et W (voir la figure 3.13). L'analyse des différentes connexions donne des informations utiles pouvant être utilisées pour comprendre comment le classificateur fonctionne. Pendant la phase d'expérimentation, nous avons testé plusieurs types de noyaux : gaussien et polynomial. Nous avons observé des performances légèrement inférieures pour le noyau gaussien. De plus, nous avons constaté que le

noyau gaussien a tendance à associer à certains chiffres des fenêtres qui n'ont aucun recouvrement avec la partie informative de l'image (pixels blancs). En fait, le noyau gaussien interprète le fond de l'image dont les pixels sont nuls comme de l'information. Bien que cohérent d'un point de vue apprentissage statistique, cela est contradictoire pour l'interprétation des résultats puisque le noyau retient l'absence d'information comme caractéristique discriminante. Ainsi, nous avons choisi d'utiliser un noyau polynomial, car il génère des couples actifs seulement s'il y a un chevauchement entre les pixels des chiffres I et des fenêtres W, ce qui facilite l'interprétation de la visualisation.

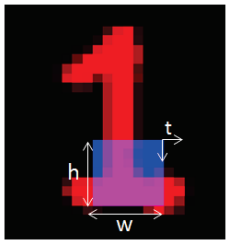


FIGURE 3.13 : Nous proposons une représentation des couples actifs, en affichant simultanément l'image du chiffre (point actif) en rouge et la fenêtre W (associée au noyau actif) en bleu.

3.5.4.4 Description du protocole

Nous avons construit le modèle de séparation multi-classes, à partir de 45 classifieurs binaires avec une stratégie 1 contre 1. La sortie de chaque séparateur est probabilisée selon la méthode de Platt [1999]. Pour tous les classifieurs, on définit un ensemble d'apprentissage et de validation (utilisé pour régler la valeur de λ_1 par validation croisée) de taille 1000 chacun. Pour chaque classifieur, nous parcourons l'intégralité du chemin, afin d'optimiser la valeur de λ_1. Au cours de cette simulation, nous nous concentrons sur l'intelligibilité du modèle KB_DRSVM, nous choisissons $\lambda_2 = 10^{-9}$ afin d'induire de la parcimonie et donc un modèle simple.

L'architecture de la famille du noyau utilisée est la même pour tous les classifieurs. Une étude préliminaire nous a conduit à choisir un noyau polynomial de degré 3 calculé à partir d'une fenêtre carrée de côté $w = h = 7$ pixels et un pas de translation $t = 3$ pixels. Dans cette configuration 49 fenêtres sont extraites de chaque image et la famille de noyaux a 49 éléments. Notons qu'un pas de translation $t = 1$ pixel est utile, car il génère un modèle plus souple qui a une meilleure performance de classification. Mais cela augmente drastiquement la taille de la famille de noyaux et conduit à des limitations en terme de mémoire pour stocker la matrice de Gram associée au *kernel basis* de taille $n \times nm$ (n et m désignent respectivement le nombre de points et de noyaux). Remarquons que la taille de la fenêtre a une grande influence sur le modèle : utiliser de petites fenêtres conduit à utiliser une grande famille de noyaux, mais qui peuvent échouer à prendre en compte la variabilité spatiale des données.

Analyse des résultats : nous avons affiché la matrice de confusion sur le tableau 3.2. Le taux global d'erreur de classification du modèle KB_DRSVM en multi-classe est de 5,25 %, en deçà de l'état de l'art [Ciresan et al., 2012]. Les résultats sont satisfaisants dans la mesure où nous avons appris sur une petite base d'images, et que notre objectif est de construire un modèle suffisamment fiable, mais qui soit avant tout explicatif et capable d'intégrer au sein de sa formulation des informations utiles pour l'interprétation.

Prediction / Vérité	0	1	2	3	4	5	6	7	8	9
0 (980)	**951**	0	1	0	0	17	7	1	1	2
1 (1135)	0	**1116**	4	2	0	0	4	0	9	0
2 (1032)	1	5	**967**	5	12	4	7	*11*	15	5
3 1010	0	3	16	**946**	1	*22*	0	5	4	13
4 (982)	2	1	2	0	**931**	0	8	3	6	*29*
5 (892)	8	6	1	*19*	8	**832**	6	1	5	6
6 (958)	9	4	3	2	11	10	**915**	1	3	0
7 (1028)	0	15	*24*	6	2	0	0	**967**	3	11
8 (974)	4	1	6	19	5	13	6	8	**905**	7
9 (1009)	2	4	3	6	*22*	7	1	11	8	**945**

TABLE 3.2 : Ce tableau représente la matrice de confusion du classifieur multi-classe. Nous construisons 45 classifieurs binaires (stratégie un contre un) dont les sorties sont probabilisées. L'affectation de classe est effectuée selon le vote le plus fort.

Analyse des couples actifs : ici, nous nous focalisons sur la forme du classificateur binaire associé à la simulation 4 contre 7, afin d'illustrer l'intelligibilité du modèle KB_DRSVM. La figure 3.14 représente les couples actifs associés aux plus forts coefficients issus du classificateur. Les exemples affichés montrent que les couples sont localisés sur des zones discriminantes de chaque classe, mettant en évidence des formes caractéristiques de chaque chiffres. Parmi ces points actifs nous pouvons noter que :

(a) indique que certains utilisateurs tracent le chiffre 4 avec une boucle,
(f) indique que parmi les utilisateurs qui écrivent le chiffre 4 sans boucle, certains tracent un angle aigu,
(d) indique que certains utilisateurs écrivent le chiffre 7 avec un angle aigu,
(g) indique que certains utilisateurs tracent horizontalement la seconde barre du chiffre 7.

Le modèle KB_DRSVM établit automatiquement les connexions entre les images et les noyaux, nous pouvons donner une interprétation aux couples actifs à l'aide de la méthode de visualisation (voir 3.14) et comprendre comment fonctionne le classifieur. Pour chaque classi-

fieur, l'analyse des couples actifs permet de mettre en évidence les zones pertinentes associées aux images caractéristiques de chaque classe.

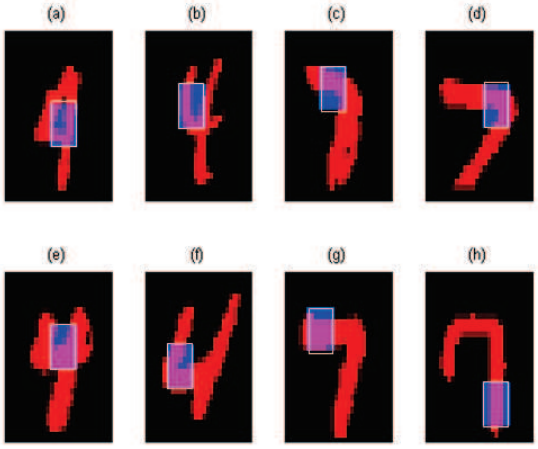


FIGURE 3.14 : Nous affichons les couples actifs (voir 3.13), dont les coefficients sont les plus élevés au sein du modèle optimal 4 contre 7 (l'un des 45 classificateurs binaires générés). L'analyses des différents choix de connexions permet de faire ressortir certains comportements d'utilisateurs et d'illustrer ainsi la capacité d'interprétation de notre approche.

3.5.4.5 Discussion

Nous avons appliqué notre modèle sur les données MNIST, une base d'images pour laquelle il est simple de mettre en œuvre notre protocole, afin d'extraire de l'interprétabilité de la solution : construction de noyaux adaptés, apprentissage et analyse. L'approche est prometteuse, car elle permet d'obtenir des performances correctes tout en ayant un aspect interprétable. En outre, la visualisation des couples actifs permet d'extraire de l'information pertinente sur les données, comme par exemple la manière d'écrire de certains utilisateurs. De plus, l'interprétabilité du modèle permet aussi d'avoir des intuitions sur le choix des noyaux. En effet, pendant la phase de réglage de la fenêtre du noyau, nous avons analysé les solutions obtenues pour différentes tailles w par visualisation des couples actifs et observé l'interaction entre la base d'image et la famille de noyaux. Cette analyse nous a permis d'avoir une intuition sur la taille de la fenêtre à appliquer sur les images. Cependant le *kernel basis* est un modèle à grandes dimensions et nous avons été confrontés à des limitations en terme de stockage de mémoire. Ainsi, nous n'avons pas pu utiliser une famille de noyau aussi fine que nous le voulions qui auraient été plus à même de capter les singularités spatiales de quelques images. De plus nous avons été contraints d'utiliser seulement $1/6$ de la base d'apprentissage, car le nombre de noyaux générés pour chaque image est un facteur limitant dans cette application.

3.6 Conclusion

Dans ce chapitre, nous nous sommes interrogés sur la manière d'intégrer au sein de l'algorithme linéaire DRSVM des informations de topologie et de structure sur les données. La méthode des machines à noyaux est particulièrement appropriée à cette problématique dans la mesure où elle permet de changer l'espace de représentation des données.

L'étude des noyaux a conduit à nous focaliser sur un modèle spécifique de machine à noyau appelé *kernel basis* dont la forme nous a semblé particulièrement adaptée à la génération de modèles interprétables. Nous avons cherché à donner un cadre formel au modèle *kernel basis* via les RKHS. L'analyse montre que l'utilisation d'une pénalisation uniquement en norme 2 n'est pas pertinente pour le *kernel basis*. Le problème DRSVM inclue une pénalisation *elastic net* qui ne permet d'utiliser le kernel trick. A la place, nous avons adopté une approche par dictionnaire, afin de *kerneliser* le problème DRSVM (voir équation 3.15). Le cadre théorique qui a été défini nous permet d'appliquer le même algorithme de chemin robuste proposé dans le chapitre 2 afin de résoudre le DRSVM dans une configuration de type *kernel basis*.

Finalement, nous avons mené une phase d'expérimentation sur des données réelles, afin d'analyser le comportement de la méthode et d'illustrer sa capacité à induire de l'interprétabilité au sein de la solution. Nous avons en particulier appliqué notre modèle sur des données images et nous avons réalisé le processus total, c'est-à-dire le choix du noyau, l'apprentissage modèle du DRSVM, et l'analyse du modèle par l'interprétation de la solution.

Conclusion

Bilan des travaux

Les travaux de thèse présentés dans ce manuscrit sont focalisés sur des manières d'induire de l'interprétabilité au sein d'un modèle de classification. Dans cette optique, nous avons commencé par étudier un choix approprié pour la fonction de coût et la régularisation et avons opté respectivement pour la fonction charnière pour sa simplicité car elle n'implique pas de paramètre supplémentaire, et pour une pénalité *elastic-net* car elle permet de faire de la sélection de variables tout en corrigeant certains défauts inhérents au LASSO.

Cela nous a conduit à nous intéresser à un problème de classification appelé DRSVM, et plus particulièrement à un algorithme construisant un chemin de régularisation à partir d'un problème équivalent. Dans certaines configurations de données, nous avons constaté une certaine instabilité sur le chemin dont l'analyse des causes nous a révélé que la résolution du chemin de régularisation est inadaptée. Nous avons alors étudié le problème initial non-différentiable grâce à la théorie de la sous-différentielle afin de dériver les équations d'optimalité. Cette analyse nous a permis de comprendre quel paramètre choisir afin de construire le chemin de régularisation. Nous avons alors proposé un chemin alternatif par rapport à λ_1 le paramètre régularisation en norme 1 et montré qu'il est linéaire par morceaux.

Cependant le problème DRSVM étant un problème linéaire, nous sommes limités à un unique choix dans la représentation des observations. Or, il est possible que le format des données d'origine ne prenne pas en compte la richesse de leur topologie et de leur structure. Dans une perspective d'interprétabilité, il nous a paru cohérent de générer un modèle qui englobe ces informations. Aussi, l'approche noyau nous a semblé pertinente dans la mesure où elle introduit des mesures de similarité permettant de changer l'espace de représentation. L'étude des noyaux, no us a mené à nous intéresser plus particulièrement à une forme de modèle appelé *kernel basis* qui permet, de part sa structure, de générer des modèles où l'influence des noyaux est pondérée localement. Dans un premier temps nous avons mené une analyse via la théorie des RKHS afin de déterminer la pénalisation appropriée pour ce modèle. La conclusion de cette étude, nous a montré que la pénalisation uniquement en norme 2 n'est pas pertinente pour le *kernel basis*.

Nous avons alors recherché comment intégrer le modèle *kernel basis* au sein du problème DRSVM. L'introduction de la kernelisation via le *kernel trick* n'est pas adaptée ici en raison de la présence du terme de régularisation en norme 1. Aussi, nous avons opté pour une approche par dictionnaire que nous avons construit de manière à générer la forme *kernel basis* au sein du DRSVM. Puis nous avons validé notre fusion du DRSVM à travers plusieurs ensembles de données jouets avec pour chaque simulation une double exigence : nous avons vérifié que notre modèle effectue la tâche de classification de façon satisfaisante et nous avons analysé la forme de solution pour montrer qu'elle génère au son sein des éléments interprétables permettant d'expliquer les observations. Enfin, nous avons choisi de traiter la base d'images MNIST afin de valider notre approche sur des données réelles. La visualisation des images, nous a permis d'une part, de jouer le rôle d'expert et d'introduire un a priori dans le choix des noyaux à utiliser, et d'autre part, d'analyser la forme de la solution et d'illustrer l'interprétabilité de notre modèle.

Discussion

Dimensionnalité du kernel basis

Un défaut inhérent à la structure du *kernel basis* est d'être un modèle à haute dimension. Plus précisément, si n désigne le nombre de points d'apprentissage et m le nombre de noyaux, il est nécessaire d'optimiser $n \times m$ coefficients. Pendant la phase d'expérimentation sur les données images MNIST, nous avons rencontré des limitations techniques en terme de mémoire, du à la dimensionnalité du *kernel basis*. L'élément le plus volumineux à stocker est la matrice *kernel basis* K définie de la façon suivante :

$$K = (k^l(x_i, x_j))_{1 \leq i,j \leq n, 1 \leq l \leq m} \quad . \tag{3.16}$$

On peut observer que la taille de la matrice K est $n^2 m$. En conséquence, cela nous a contraint à réduire la base d'apprentissage, entraînant une diminution de la variabilité des données et donc de la représentativité de la base. De plus, nous avons restreint la richesse de notre famille de noyaux, ce qui a pour effet d'obtenir un modèle moins flexible. Ainsi, ces deux limitations ont eu pour effet de dégrader les performances en classifications de notre modèle.

Choix des noyaux

Notre méthode ne permet pas de créer ex-nihilo de l'information intelligible pour expliquer les observations. En effet, elle nécessite qu'en amont un a priori expert soit intégré lors de la construction de la famille de noyaux afin d'adapter la modèle *kernel basis* au données d'observation. Notons que les noyaux sont parfois considérés comme des boites noires et moins

interprétables que les modèles linéaire. Cette vision est justifiée dans l'hypothèse où n'avons pas un a priori expert sur les noyaux et que nous ne maîtrisons pas leur construction du point de vue de l'interprétabilité.

Exploiter pleinement l'interprétabilité

Pendant la phase d'expérimentation, nous nous sommes rendus compte de la difficulté à mettre en exergue les informations contenues dans le modèle. La raison principale est que notre méthode nécessite trois étapes : définition des la famille de noyaux via un a priori expert, phase d'apprentissage et analyse via une connaissance experte de la forme de la solution afin de déduire des éléments explicatifs. Ainsi notre méthode n'est pas adaptée à la recherche en aveugle d'éléments interprétables au sein du classifieur car le choix des noyaux conditionnent fortement la forme de la solution finale. De plus, la connaissance de l'expert en début (a priori noyau) et fin (analyse de la forme de la solution) de processus est essentielle afin d'axer la recherche d'éléments explicatifs au sein du modèle.

Perspectives

Résolution en parallèle

Afin de pouvoir gérer les problèmes de mémoire, il semble prometteur d'utiliser des techniques de parallélisation afin de réduire la taille de la matrice K. Pendant la construction nous n'avons pas besoin de toute l'information contenue dans K et il est possible envisager de répartir la matrice K (voir (3.16)) sur plusieurs machines et de les faire communiquer uniquement quand la transmission de l'information est nécessaire. La résolution du DRSVM serait adaptée facilement à une telle procédure.

Evaluer l'intérêt d'autres régularisations

Dans le cadre de cette thèse, nous nous sommes intéressés à la pénalisation *elastic net* mais il peut être aussi intéressant d'explorer d'autres pistes pour la régularisation. Dans une perspective d'interprétabilité, la norme CAP (Composite Absolue Penalty) nous a semblé pertinente dans la mesure où elle permet de prendre en compte des informations de hiérarchie sur la famille de noyaux au sein de la solution [Zhao et al., 2009]. D'autre part, nous avons étudié la norme MCP pour sa capacité à réduire le biais du LASSO pour les modèles à coefficients élevés [Zhang, 2010]. De plus nous avons mené une analyse du problème SCM MCP via la

sous-différentielle de Clark et il semble théoriquement possible de construire un chemin de régularisation (voir annexe A). Une autre pénalité analogue et potentiellement intéressante est celle du SCAD (*smoothly clipped absolute deviation*, Zhang et al. [2006]).

Proposer un double-chemin de régularisation

Afin de faciliter les réglages de paramètre, nous avons imaginé générer l'intégralité du double chemin, autrement dit de construire une surface de régularisation. En analysant le problème DRSVM via la théorie de la sous-différentielle, nous avons remarqué que pour chaque itération, deux événements (a) et (b) sont susceptibles de se produire avant tous les autres et que leur ordre d'apparition dépend explicitement de la valeur du paramètre λ_2 (le paramètre de régularisation associé à la norme 2). Ainsi en calculant les conditions de transition pour (a) et (b) à chaque itération, on obtient un arbre, qui représente le maillage associé à la surface de régularisation. Notons néanmoins que cette solution est coûteuse en terme calculatoire puisqu'à chaque étape on génère deux embranchements différents.

Annexe

A Analyse du SVM MCP via la sous-différentielle

Nous avons étudié d'autres pénalités afin de construire le chemin la régularisation. En particulier la pénalisation MCP (*minimax concave penalty*, Zhang [2010]) qui permet de débiaiser les modèles dans le cas où les coefficients sont importants, mais qui est non convexe. Nous avons alors posé le problème SVM MCP, c'est à dire la combinaison de la fonction charnière avec la pénalisation MCP et nous l'avons analysé sous le prisme de la théorie de la sous-différentielle afin d'établir s'il est possible de construire un chemin. Le problème étant non convexe, il est nécessaire de partir de la définition de la sous-différentielle de Clarke (voir Rockafellar and Wets [2009] et l'équation (A.3)) afin de dériver les conditions d'optimalité. La conclusion de cette étude suggère qu'il est possible de construire un chemin par rapport à la régularisation L_1 pour le MCP. Néanmoins nous avons rencontré des problèmes pour initialiser le chemin et nous n'avons pas retenu pour l'instant cette solution.

Le problème que nous cherchons à résoudre est le suivant :

$$\min_{\beta_0,\beta} J(\beta_0,\beta) \quad \text{avec} \quad J(\beta_0,\beta) = \sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^{p} \rho|\beta_j| \tag{A.1}$$

et :

$$\rho(u) = \begin{cases} u - \dfrac{u^2}{2\gamma\lambda} & \text{si } u < \gamma\lambda \\ \dfrac{\gamma\lambda}{2} & \text{si } u \geq \gamma\lambda\,. \end{cases} \tag{A.2}$$

La fonction $\rho(|\cdot|)$ est la composée de la fonction $|\cdot|$ qui est convexe mais non différentiable en zéro, avec la fonction $\rho(\cdot)$ qui est convexe mais différentiable en zéro, on obtient alors une fonction non convexe et non différentiable en zéro. Il s'ensuit que non seulement nous ne pouvons pas définir le gradient de cette fonction, mais que nous ne pouvons pas non plus utiliser le théorème d'optimalité classique du sous-gradient (voir théorème 1), qui est défini uniquement

pour les fonctions convexes. En revanche nous allons montrer que nous pouvons utiliser une notion qui généralise le concept de dérivée et qui s'appelle la différentielle de Clarke. Cette différentielle directionnelle dans la direction d est définie pour toute fonction localement lipschitzienne et se formalise de la façon suivante :

$$D_c f(\beta, d) = \limsup_{\delta \to \beta, \varepsilon \to 0} \frac{f(\delta + \varepsilon d) - f(\delta)}{\varepsilon} \quad . \tag{A.3}$$

Cette définition quelque peu abstraite, permet d'introduire un théorème qui est utilisé pour trouver les conditions d'optimalité nécessaires à la construction du chemin de régularisation par rapport à λ. Ce théorème s'énonce de la manière suivante :

Théorème 6 (Théorème d'optimalité pour les fonctions non-convexes)**.** *Soit f une fonction localement lipschitzienne, tel que f puisse se décomposer sous la forme $f = f_1 + f_2$, avec f_1 une fonction non-convexe différentiable et f_2 une fonction convexe non-différentiable, alors :*

$$\beta^* \textit{ est un minimum local } \Rightarrow -\nabla_\beta f_1(\beta^*) \in \partial_\beta f_2(\beta^*).$$

Ici J est une fonction non-différentiable non-convexe mais localement lipschitzienne et qui s'écrit sous la forme : $J = J_1 + J_2$ avec :

$$\begin{aligned} J_1(\beta_0, \beta) &= -\frac{1}{2\gamma} \sum_{j=1}^{p} \beta_j^2 \mathbf{1}_{[-\gamma\lambda < \beta_j < \gamma\lambda]} + (\frac{\gamma\lambda^2}{2} - \lambda \sum_{j=1}^{p} |\beta_j|) \mathbf{1}_{[|\beta_j| \geq \gamma\lambda]} \\ J_2(\beta_0, \beta) &= \sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \left\|\beta\right\|_1 . \end{aligned} \tag{A.4}$$

La fonction J_1 est différentiable mais non-convexe et J_1 est non-différentiable en zéro mais convexe, nous pouvons donc appliquer le théorème afin de déterminer l'optimum de J.

Calculons le gradient de J_1 par rapport à β_j :

$$\begin{aligned} \nabla_{\beta_j} J_1(\beta_0, \beta) &= -\frac{1}{\gamma} \beta_j \mathbf{1}_{[|\beta_j| < \gamma\lambda]} + (0 - \lambda sign(\beta_j)) \mathbf{1}_{[|\beta_j| \geq \gamma\lambda]} \\ &= -\frac{1}{\gamma} \beta_j \mathbf{1}_{[|\beta_j| < \gamma\lambda]} - \lambda sign(\beta_j) \mathbf{1}_{[|\beta_j| \geq \gamma\lambda]}. \end{aligned} \tag{A.5}$$

Afin de calculer le sous gradient de J_2 par rapport à β_j, nous introduisons la notation r pour désigner le résidu défini comme suit :

$$r_i = 1 - y_i(\beta_0 + \beta^T x_i). \tag{A.6}$$

Ainsi que les ensembles de points (I_0,I_α,I_1) définis de la manière suivante :

– $I_0 = \{i, r_i < 0\}$
– $I_\alpha = \{i, r_i = 0\}$
– $I_1 = \{i, r_i > 0\}$

Et aussi les ensembles de variables suivants :

– $\mathscr{V}_\beta = \{j, \beta_j \neq 0\}$
– $\mathscr{V}_{UNP} = \{j, \beta_j < \gamma\lambda\}$
– $\mathscr{V}_P = \{j, \beta_j \geq \gamma\lambda\}$
– $\mathscr{V}_0 = \{j, \beta_j = 0\}$

Les ensembles $\mathscr{V}_P$ et $\mathscr{V}_{UNP}$ correspondent respectivement aux variables pénalisées et dépénalisées

Calculons le sous gradient de J_2 par rapport à β_j :

$$\begin{aligned} \partial_{\beta_j} J_2 &= \partial_{\beta_j} \{ \sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \left\| \beta \right\|_1 \} \\ &= -\sum_{i=1}^{n} \alpha_i y_i x_{ij} + \lambda \Gamma_j, \end{aligned} \tag{A.7}$$

avec $(\alpha_i)_{1 \leq i \leq n}$ et $(\Gamma_j)_{1 \leq j \leq p}$ vérifiant :

$$\begin{aligned} &\alpha_i = 0 \text{ si } i \in I_0 \\ &\alpha_i \in [0,1] \text{ si } i \in I_\alpha \\ &\alpha_i = 1 \text{ si } i \in I_1 \end{aligned}$$

$$\begin{aligned} &\Gamma_j = sign(\beta_j) \text{ si } j \in \mathscr{V}_\beta \\ &\Gamma_j = [-1,1] \text{ si } j \in \mathscr{V}_0 \end{aligned}$$

Pour obtenir les équations d'optimalité on utilise le théorème suscité (voir 6), c'est à dire : $-\nabla_{\beta_j} J_1 \in \partial_{\beta_j} J_2$. Deux cas sont à séparer, le cas où β_j est nul et le cas β_j est différent de zéro. En effet si β_j est non nul J_2 est différentiable et son sous gradient est réduit à son gradient ce qui implique les égalités suivantes :

$$-(-\frac{1}{\gamma}\beta_j \mathbf{1}_{[|\beta_j| < \gamma\lambda]} - \lambda sign(\beta_j) \mathbf{1}_{[|\beta_j| \geq \gamma\lambda]}) = -\sum_{i=1}^{n} \alpha_i y_i x_{ij} + \lambda sign(\beta_j) \text{ , } \forall j \in \mathscr{V}_\beta.$$

Il faut encore séparer les cas suivants : $|\beta| < \gamma\lambda$ et $|\beta| \geq \gamma\lambda$, car les équations changent à cause de la présence de la fonction indicatrice. Ainsi on a les équations suivantes :

$$\begin{aligned} -\frac{1}{\gamma}\beta_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij} &= -\lambda sign(\beta_j) \text{ , } && \forall j \in \mathscr{V}_P \\ \sum_{i=1}^{n} \alpha_i y_i x_{ij} &= 0 \text{ , } && \forall j \in \mathscr{V}_{UNP}. \end{aligned} \tag{A.8}$$

Si $\beta_j = 0$ on a l'équation suivante :

$$-\sum_{i=1}^{n} \alpha_i y_i x_{ij} + \lambda \Gamma_j = 0 \, , \forall j \in \mathscr{V}_0 \text{ avec } \Gamma_j \in [-1,1]. \tag{A.9}$$

Pour obtenir le gradient de J par rapport à β_0 nous procédons de manière similaire. Comme J_1 ne fait pas intervenir β_0 son gradient est nul et en utilisant la sous différentielle de J_2 on obtient l'équation suivante :

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{A.10}$$

Comme les équations d'optimalité font intervenir l'ensemble (I_α), un ensemble de points respectant une égalité définie par rapport à la fonction r, la solution du problème doit vérifier les équations suivantes :

$$\forall i \in I_\alpha \, , \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} = 0. \tag{A.11}$$

Une brève analyse montre que l'initialisation se passe de façon identique que pour le DRSVM. C'est durant l'étape principale que l'algorithme fonctionne différemment. L'analyse des équations [A.8, A.9, A.10,A.11] montre que les paramètres système (β, β_0, α, Γ) évoluent de façon linéaire par rapport à λ. En effet nous avons $|I_\alpha| + |\mathscr{V}_\beta| + 1$ équations, pour le même nombre d'équations. Nous construisons le système dérivé des paramètres par rapport à λ :

$$\forall j \in \mathscr{V}_P \, , -\frac{1}{\gamma}\frac{\Delta\beta_j}{\Delta\lambda} - \sum_{i \in I_\alpha} \frac{\Delta\alpha_i}{\Delta\lambda} y_i x_{ij} = -sign(\beta_j) \tag{A.12}$$

$$\forall j \in \mathscr{V}_{UNP} \, , \sum_{i \in I_\alpha} \frac{\Delta\alpha_i}{\Delta\lambda} y_i x_{ij} = 0 \tag{A.13}$$

$$\sum_{i \in I_\alpha} \frac{\Delta\alpha_i}{\Delta\lambda} y_i = 0 \tag{A.14}$$

$$\forall i \in I_\alpha \, , \frac{\Delta\beta_0}{\Delta\lambda} + \sum_{j \in \mathscr{V}_\beta} \frac{\Delta\beta_j}{\Delta\lambda} x_{ij} = 0. \tag{A.15}$$

Nous définissons M la matrice associée au système dérivé de la manière suivante :

$$M[\alpha_{I_\alpha}, \beta_{\mathscr{V}_P}, \beta_{\mathscr{V}_{UNP}}, \beta_0]^T = \begin{pmatrix} -sign(\beta_{\mathscr{V}_P}) \\ 0 \end{pmatrix}. \tag{A.16}$$

Les notations : α_{I_α}, $\beta_{\mathscr{V}_P}$ et $\beta_{\mathscr{V}_{UNP}}$ signifient que nous extrayons respectivement les variables sur les ensembles d'indices $\mathscr{V}_P$,$\mathscr{V}_{UNP}$ et I_α. Nous pouvons donc construire le chemin de régularisation en inversant la matrice M pour calculer la dérivée des différents paramètres par

rapport à λ puis détecter les différents événements susceptibles de venir modifier le système d'équations.

Par rapport au DRSVM il y a deux nouveaux événements à considérer : une variable est dépénalisée ou une variable est repénalisée ce qui revient à dire que la variable en question touche en valeur absolue la valeur $\gamma\lambda$. Il faut faire attention quand on calcule la condition associée à ces événements car le seuil n'est pas statique au cours du chemin puisqu'il dépend de λ. Nous détaillons ici le calcul afin d'obtenir $\delta_\lambda = \lambda^{(2)} - \lambda^{(1)}$ pour que $\beta_j^{(2)} = \gamma\lambda^{(2)}$ sachant que l'on connait le triplet ($\lambda^{(1)}$,$\beta_j^{(1)}$, $\frac{\Delta\beta_j}{\Delta\lambda}$). On a alors :

$$\begin{aligned} \beta_j^{(2)} &= \gamma\lambda^{(2)} \\ \beta_j^{(1)} + \delta_\lambda \frac{\Delta\beta_j}{\Delta\lambda} &= \gamma\lambda^{(1)} + \gamma\delta_\lambda \\ \delta_\lambda &= -\frac{\gamma\lambda^{(1)} - \beta_j^{(1)}}{\gamma - \frac{\Delta\beta_j}{\Delta\lambda}}. \end{aligned}$$

Le même travail pour l'autre borne $\beta_j^{(2)} = -\gamma\lambda^{(2)}$ implique l'équation suivante :

$$\delta_\lambda = -\frac{-\gamma\lambda^{(1)} - \beta_j^{(1)}}{-\gamma - \frac{\Delta\beta_j}{\Delta\lambda}}.$$

A priori nous ne savons pas quelle borne β_j atteint mais comme nous construisons le chemin de régularisation tel que λ diminue, cela implique $\delta_\lambda < 0$. Il faut donc sélectionner $\delta\lambda$ de la manière suivante :

$$\delta_\lambda = \min\left(-\frac{\gamma\lambda^1 - \beta_j^1}{\gamma - \frac{\Delta\beta_j}{\Delta\lambda}}, -\frac{-\gamma\lambda^1 - \beta_j^1}{-\gamma - \frac{\Delta\beta_j}{\Delta\lambda}} \right).$$

Les deux événements se différencient au niveau des ensembles de variables j sur lesquels on va calculer δ_λ. En effet dans le cas de dépénalisation on utilise l'ensemble $\mathscr{V}_P$ et dans le cas d'une repénalisation on utilise l'ensemble $\mathscr{V}_{UNP}$.

L'analyse montre que le fonctionnement du du SVM MCP est similaire au DRSVM (à l'exception de la nature de certains événements) et il semble théoriquement possible de construire un chemin de régularisation. Mais nous avons rencontré des difficultés lors de l'initialisation qui nous empêchent de continuer la construction du chemin. Ainsi nous avons écarté provisoirement la solution SVM MCP en attendant de trouver un moyen d'initialiser le chemin.

Bibliographie

Massih-Reza Amini. *Apprentissage machine : de la théorie à la pratique*. Editions Eyrolles, 2015.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, pages 337–404, 1950.

Pat Langley Avrim Blum. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2) :245—-271, 1997.

Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual information analysis : a comprehensive study. *Journal of Cryptology*, 24(2) :269–291, 2011.

Arjan B Berkelaar, Kees Roos, and Tamás Terlaky. *The optimal set and optimal partition approach to linear and quadratic programming*. Springer, 1997.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2) :121–167, 1998.

Stéphane Canu, Xavier Mary, and Alain Rakotomamonjy. Functional learning through kernels. Technical report, LITIS Rouen, december 2002.

Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

Antoine Cornuéjols and Laurent Miclet. *Apprentissage artificiel : concepts et algorithmes*. Eyrolles, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3) : 273–297, 1995.

ILOG CPLEX. High-performance software for mathematical programming and optimization, 2005.

Christine De Mol, Ernesto De Vito, and Lorenzo Rosasco. Elastic-net regularization in learning theory. *Journal of Complexity*, 25(2) :201–230, 2009.

David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization. *Proceedings of the National Academy of Sciences*, 100(5) : 2197–2202, 2003.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2) :407–499, 2004.

Michael Eichler. Causal inference in time series analysis. *Causality : statistical perspectives and applications*, pages 327–352, 2012.

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3) :37, 1996.

Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2) :302–332, 2007.

Jerome H Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1) :55–77, 1997.

Tomas Gal. *Postoptimal analyses, parametric programming and related topics*. Walter de Gruyter, 1995.

Thomas Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1) :49–58, 2003.

Alireza Ghaffari-Hadigheh, Habib Ghaffari-Hadigheh, and Tamás Terlaky. Bi-parametric optimal partition invariancy sensitivity analysis in linear optimization. *Central European Journal of Operations Research*, 16(2) :215–238, 2008.

Mehmet Gönen and Ethem Alpaydin. Localized multiple kernel learning. In *Proceedings of the 25th international conference on Machine learning*, pages 352–359. ACM, 2008.

Clive William John Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3) :424–438, 1969.

Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx : Matlab software for disciplined convex programming, 2008.

Vincent Guigue, Alain Rakotomamonjy, and Stéphane Canu. Kernel basis pursuit. In *Machine Learning : ECML 2005*, pages 146–157. Springer, 2005.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

Isabelle Guyon, Constantin Aliferis, and André Elisseeff. Causal feature selection. Technical report, 2007.

Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *The Journal of Machine Learning Research*, 5 :1391–1415, 2004.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning : data mining, inference and prediction. *The Mathematical Intelligencer*, 27 (2) :83–85, 2005.

Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC Press, 1990.

I Heller. Sensitivity analysis in linear programming. In *LRP Seminar, Logistics Research Project (The George Washington University)*, 1954.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms I : Part 1 : Fundamentals*, volume 305. Springer Science & Business Media, 1996.

Shunsuke Ihara. *Information theory for continuous systems*, volume 2. Singapore : World Scientific, 1993.

Benjamin Jansen. Sensitivity analysis in quadratic programming. In *Interior Point Techniques in Optimization*, pages 57–69. Springer, 1997.

Ian Jolliffe. *Principal Component Analysis*. Encyclopedia of Statistics in Behavioral Science, 2005.

W. Alan Nicewander Joseph Lee Rodgers. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42 (1) :59–66, 1988.

Masayuki Karasuyama and Ichiro Takeuchi. Suboptimal solution path algorithm for support vector machine. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 473–480. Omnipress, 2011.

Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Kernels for graphs. *Kernel methods in computational biology*, 39(1) :101–113, 2004.

Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data : an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1) :82–95, 1971.

Sun Yuan Kung. *Kernel methods and machine learning*. Cambridge University Press, 2014.

Antoine Lachaud, David Mercier, Stephane Canu, and Frederic Suard. A robust regularization

path for the doubly regularized support vector machine. In *Proceedings of ESANN*, pages 313–318, 2014.

Gert RG Lanckriet, Tijl De Bie, Nello Cristianini, Michael I Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16) :2626–2635, 2004.

Darrin P Lewis, Tony Jebara, and William Stafford Noble. Nonstationary kernel combination. In *Proceedings of the 23rd international conference on Machine learning*, pages 553–560. ACM, 2006.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2 :419–444, 2002.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, number 14 in 1, pages 281–297. Oakland, CA, USA., 1967.

Stephane G Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7) :674–693, 1989.

James Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, pages 415–446, 1909.

Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and KR Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41–48. IEEE, 1999a.

Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. *Advances in neural information processing systems*, 11(1) :536–542, 1999b.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

Michael R Osborne, Brett Presnell, and Berwin A Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis-Institute of Mathematics and its Applications*, 20(3) :389–404, 2000.

Neelamadhab Padhy, Dr Mishra, Rasmita Panigrahi, et al. The survey of data mining applications and feature scope. *arXiv preprint arXiv :1211.5723*, 2012.

Mee Young Park and Trevor Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 69(4) :

659–677, 2007.

Paul Pavlidis, Jason Weston, Jinsong Cai, and William Noble Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the fifth annual international conference on Computational biology*, pages 249–255. ACM, 2001.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. More effciency in multiple kernel learning. In *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML)*, 2007.

R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Roman Rosipal, Leonard J Trejo, and Bryan Matthews. Kernel pls-svc for linear and nonlinear classification. In *ICML*, pages 640–647, 2003.

Saharon Rosset and Ji Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, pages 1012–1030, 2007.

Stefan Ruzika and Margaret M Wiecek. Approximation methods in multiobjective programming. *Journal of optimization theory and applications*, 126(3) :473–501, 2005.

Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *Computational learning theory*, pages 416–426. Springer, 2001.

Simon J. Sheather. Density estimation. *Statistical Science*, 19 :588–597, 2004.

Lei Shi, Xiaolin Huang, Zheng Tian, and Johan AK Suykens. Quantile regression with ℓ 1—regularization and gaussian kernels. *Advances in Computational Mathematics*, 40(2) :517–551, 2014.

Galit Shmueli. To explain or to predict ? *Statistical Science*, pages 289–310, 2010.

Bernard Walter Silverman. *Density estimation for statistics and data analysis*. School of Mathematics University of Bath, 1986.

Guohui Song, Haizhang Zhang, and Fred J Hickernell. Reproducing kernel banach spaces with the ℓ1 norm. *Applied and Computational Harmonic Analysis*, 34(1) :96–116, 2013.

Sören Sonnenburg, Gunnar Raetsch, and Christin Schaefer. A general and efficient multiple kernel learning algorithm. In *Advances in Neural Information Processing Systems 18*, pages 1273–1280, 2005.

Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

Frédéric Suard and David Mercier. Using kernel basis with relevance vector machine for feature selection. In *Artificial Neural Networks–ICANN 2009*, pages 255–264. Springer, 2009.

Marie Szafranski and Yves Grandvalet. Keops : Kernels organized into pyramids. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 8262–8266. IEEE, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Vladimir N Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 606–613. IEEE, 2009.

Pascal Vincent and Yoshua Bengio. Kernel matching pursuit. *Machine Learning*, 48(1-3) : 165–187, 2002.

Li Wang, Ji Zhu, and Hui Zou. The doubly regularized support vector machine. *Statistica Sinica*, 16(2) :589, 2006.

Li Wang, Ji Zhu, and Hui Zou. Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics*, 24(3) :412–419, 2008.

Waloddi Weibull. A statistical distribution function of wide applicability. *Journal of applied mechanics*, pages 293–297, 1951.

Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1) :1–37, 2008.

Gui-Bo Ye, Yifei Chen, and Xiaohui Xie. Efficient variable selection in support vector machines via the alternating direction method of multipliers. In *International Conference on Artificial Intelligence and Statistics*, pages 832–840, 2011.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 68(1) :49–67, 2006.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, pages 894–942, 2010.

Hao Helen Zhang, Jeongyoun Ahn, Xiaodong Lin, and Cheolwoo Park. Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, 22(1) :88–95, 2006.

Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped

and hierarchical variable selection. *The Annals of Statistics*, pages 3468–3497, 2009.

Ji Zhu and Trevor Hastie. Kernel logistic regression and the import vector machine. In *Advances in neural information processing systems*, pages 1081–1088, 2001.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. *Advances in neural information processing systems*, 16(1) :49–56, 2004.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(2) :301–320, 2005.