Sélection de modèle par chemin de régularisation pour les machines à vecteurs support à coût quadratique

Rémi Bonidal

UNIVERSITÉ DE LORRAINE

Département de formation doctorale en informatique **École doctorale IAEM Lorraine**

Collégium Sciences et Technologies

Sélection de modèle par chemin de régularisation pour les machines à vecteurs support à coût quadratique

THÈSE

présentée et soutenue publiquement le 19 juin 2013

pour l'obtention du

Doctorat de l'Université de Lorraine

(spécialité informatique)

par

Rémi Bonidal

Composition du jury

Président :	Le président	du jury
Rapporteurs :	Massih-Reza AMINI	PR, Université Joseph Fourier, Grenoble
	Olivier TEYTAUD	CR INRIA, Laboratoire de Recherche en Informatique, Orsay
Examinateurs :	Didier GALMICHE	PR, Université de Lorraine, Nancy
	Liva RALAIVOLA	PR, Aix-Marseille Université, Marseille
Directeurs de thèse :	Yann GUERMEUR	DR CNRS, LORIA, Nancy
	Samy TINDEL	PR, Université de Lorraine, Nancy

Équipe ABC
LORIA
UMR 7503 - 54506 Vandœuvre-lès-Nancy Cedex

Équipe PS
IECN
UMR 7502 - 54506 Vandœuvre-lès-Nancy Cedex

Mis en page avec la classe thloria.

Remerciements

En premier lieu, j'exprime ma profonde gratitude et mes sincères remerciements aux membres du jury :

- à Massih-Reza Amini et à Olivier Teytaud qui m'ont fait l'honneur d'accepter d'être les rapporteurs de cette thèse,
- à Liva Ralaivola qui m'a reçu à Marseille et avec qui j'ai pu avoir le plaisir d'échanger,
- à Didier Galmiche pour avoir été mon référent interne et pour sa disponibilité,
- à Samy Tindel, mon codirecteur de thèse, pour ses conseils et sa vision critique et avisée,
- à Yann Guermeur, mon directeur de thèse, pour son attention, sa rigueur scientifique et sa passion.

Je tiens ensuite à remercier les personnes avec lesquelles j'ai travaillé en étroite collaboration au cours de cette thèse : Fabienne Thomarat et Fabien Lauer, membres de l'équipe ABC, qui m'ont fait profiter de leurs connaissances ; et Julien Bonnel, de l'ENSTA Bretagne, pour son amabilité.

Enfin, mes pensées vont également à ceux qui me sont chers -famille, collègues et amis- sans t'oublier, toi lecteur, qui justifie par nature la rédaction d'un tel document.

Je dédie ce manuscrit
à ceux qui le liront

Table des matières

Table des figures

Liste des tableaux

Introduction générale et contributions

Les travaux présentés dans ce manuscrit de thèse portent sur la sélection de modèle pour les machines à vecteurs support (Support Vector Machines, SVM) [104] à coût quadratique mettant en œuvre une discrimination. Ils abordent successivement le cas bi-classe et le cas multi-classe (Multi-class Support Vector Machines, M-SVM). Il s'agit précisément de choisir les valeurs des hyperparamètres du problème de l'apprentissage de ces machines de manière à optimiser les performances, en l'occurrence le taux de reconnaissance du classifieur retenu.

L'objectif est de fournir un ensemble de méthodes réalisant automatiquement ce choix, qui soient efficaces au sens où elles fournissent un bon compromis entre la qualité du classifieur et le temps nécessaire à son obtention. La première référence à dépasser est l'approche naïve : une recherche exhaustive sur une grille. Les hyperparamètres des SVM sont le coefficient de régularisation et le noyau. Le choix de la valeur du coefficient de régularisation permet de réaliser un compromis entre la simplicité du classifieur et son adéquation aux données d'apprentissage. Le noyau fournit la mesure de similarité entre exemples au cœur du principe inférentiel. Ce travail de thèse a pris le parti d'aborder la sélection de modèle par le biais du parcours du chemin de régularisation [90], c'est-à-dire en balayant l'ensemble des solutions du problème d'apprentissage pour une plage de valeurs du coefficient de régularisation.

Contributions

Nous avons apporté trois contributions principales au domaine d'étude.

Choix de la valeur du coefficient de régularisation pour la ℓ_2-SVM
Après avoir redémontré les résultats relatifs au parcours du chemin de régularisation de la ℓ_1-SVM [32], en précisant au passage le lemme 2 de [66], nous proposons un algorithme efficace pour le parcours du chemin de la ℓ_2-SVM [32]. Cet algorithme n'est pas une extension directe de celui de la ℓ_1-SVM, puisque l'évolution des multiplicateurs de Lagrange n'est plus linéaire par morceaux. Il s'appuie sur un lien établi entre la ℓ_2-SVM et la SVM des moindres carrés (Least Squares SVM, LS-SVM) [99]. Nous mettons en évidence le fait qu'il permet une intégration particulièrement efficace d'un critère de sélection de modèle aux qualités démontrées : une approximation de l'erreur de la procédure de validation croisée leave-one-out dérivée de la "borne Span" [105]. L'inconvénient majeur de cette approximation réside dans son coût en temps de calcul. Nous établissons que lorsque le rang de la matrice de Gram est faible devant le nombre d'exemples,

les complexités du parcours du chemin et de la sélection de modèle sont linéaires en le nombre d'exemples. Des résultats expérimentaux illustrent la pertinence de l'approche préconisée. Il est connu que la variance du critère de sélection de modèle est un facteur qui doit être pris en compte (voir par exemple [22]). Cette observation nous a conduits à proposer deux nouvelles approximations de l'erreur en généralisation, l'une associée à la procédure de validation croisée à V pas, l'autre à celle du bootstrap .632+ [41], c'est-à-dire à des estimateurs à plus faible variance que l'erreur de validation croisée leave-one-out.

Choix de la valeur du coefficient de régularisation pour la M-SVM^2
Reprenant la démarche appliquée dans le cas bi-classe, cette étude débute par l'introduction d'une nouvelle M-SVM des moindres carrés obtenue comme extension de la M-SVM de Lee, Lin et Wahba (LLW-M-SVM) [78], la "Least Squares Multi-class SVM" (LS-M-SVM). Nous démontrons à nouveau que l'apprentissage de cette machine se réduit à la résolution d'un système linéaire et que l'erreur de la procédure de validation croisée leave-one-out s'obtient sans avoir à effectuer cette procédure. Les liens entre la LLW-M-SVM et la M-SVM^2 [63] nous permettent alors d'étendre l'algorithme de parcours du chemin de régularisation de la ℓ_2-SVM à sa généralisation multi-classe. Nous proposons ensuite une extension à la M-SVM^2 de la "borne Span" en nous fondant sur la borne Rayon-Marge multi-classe [63]. Des considérations liées au temps de calcul nous incitent à rechercher des approximations, ce qui soulève des problèmes techniques propres au cas multi-classe. Nous en présentons deux dont l'utilité pratique est établie par des résultats expérimentaux.

Alignement régularisé de noyaux centrés
Cette contribution porte sur l'autre volet de la sélection de modèle pour les SVM : le choix du noyau (plus précisément sa paramétrisation). Notre méthode se fonde sur le principe de maximisation de l'alignement noyau/cible [35], dans sa version centrée [30]. Elle l'étend à travers l'introduction d'un terme de régularisation. Nous l'appliquons à la recherche des valeurs des paramètres d'un noyau gaussien elliptique, ce qui en fait également une méthode de sélection de variables douce. Comme on pouvait l'espérer, la pénalisation ℓ_1 s'avère efficace pour produire une solution parcimonieuse. Les résultats expérimentaux obtenus sur des données jouet et du monde réel illustrent l'efficacité de la méthode, qui fait ainsi converger harmonieusement sélection de modèle et sélection de variables.

Plan

Le manuscrit débute par un chapitre d'état de l'art introduisant les machines à vecteurs support pour la discrimination ainsi que les principales méthodes de sélection de modèle qui leur sont attachées (chapitre 1). Le reste du plan suit la trame des contributions. Les chapitres 2 et 3, qui traitent de la sélection de modèle par parcours du chemin de régularisation, constituent le cœur du mémoire. S'y succèdent les contributions relevant du calcul des dichotomies (chapitre 2) et du calcul des polytomies (chapitre 3). Le dernier chapitre est dédié au choix des valeurs des

paramètres du noyau par maximisation régularisée de l'alignement noyau/cible centré. Enfin, nous attirons l'attention sur une contribution technique ayant permis l'obtention des résultats expérimentaux des chapitres 3 et 4. L'annexe A décrit un algorithme d'optimisation de la borne Rayon-Marge multi-classe par descente en gradient.

Chapitre 1

Machines à vecteurs support pour la discrimination

« This one's tricky. You have to use imaginary numbers, like eleventeen ... »
—Calvin and Hobbes, Bill Watterson

Dans ce chapitre nous présentons tout d'abord le problème de la discrimination. Les Machines à Vecteurs Support (SVM) sont ensuite présentées en deux étapes : les algorithmes des SVM bi-classes puis l'utilisation de SVM pour réaliser des polytomies. Une étude bibliographique des différentes méthodes usuelles de sélection de modèle pour ces SVM conclut le chapitre. Ce chapitre introduit les notations qui seront utilisées tout au long du manuscrit.

1.1 Cadre théorique

Hypothèses : Nous considérons des problèmes de discrimination à Q catégories. Chaque objet est représenté par sa description $x \in \mathcal{X}$ et l'ensemble $\mathcal{Y}$ des catégories y peut être identifié à l'ensemble des indices de catégories $[\![1, Q]\!]$. Nous supposons que $(\mathcal{X}, \mathcal{A})$ et $(\mathcal{Y}, \mathcal{B})$ sont des espaces mesurables et que le lien entre les descriptions et les catégories est caractérisé par une mesure de probabilité P sur l'espace mesurable $(\mathcal{X} \times \mathcal{Y}, \mathcal{A} \otimes \mathcal{B})$. (X, Y) est un couple aléatoire à valeurs dans $\mathcal{X} \times \mathcal{Y}$, distribué selon P. Cette mesure de probabilité P est inconnue et on souhaite inférer de la connaissance à son sujet afin d'être en mesure de classer des objets non étiquetés. Pour cela, on suppose disposer d'un m-échantillon $D_m = ((X_i, Y_i))_{1 \leqslant i \leqslant m}$ constitué de copies indépendantes de (X, Y).

Approche : Afin de représenter la connaissance sur P acquise à partir du m-échantillon D_m, nous cherchons une fonction f sur l'ensemble de description vers l'ensemble des catégories dans une famille de fonctions $\mathcal{F}$. Pour mesurer la manière dont cette fonction rend compte de P, nous introduisons une fonction de perte.

Définition 1 (Fonction de perte canonique pour la classification). *La fonction de perte canonique pour la classification est l'indicatrice ℓ définie par*

$$\forall \left(y, y'\right) \in \mathcal{Y}^2,\ \ell\left(y, y'\right) = 1 - \delta_{y,y'}$$

avec δ le symbole de Kronecker.

Le risque se définit comme l'espérance de la fonction de perte :

Définition 2 (Risque). *Pour f de $\mathcal{X}$ vers $\mathcal{Y}$, le risque $R(f)$ est l'espérance de la fonction de perte*

$$R(f) = \mathbb{E}_{(X,Y)\sim P}\left[\ell\left(Y, f\left(X\right)\right)\right].$$

Lorsque la nature de la mesure de probabilité est évidente, elle ne sera pas mentionnée dans l'espérance. Le risque fournit une mesure de l'adéquation de la fonction au problème considéré. Un classifieur présentant un risque minimum est appelé le classifieur de Bayes.

Définition 3 (Classifieur de Bayes). *Parmi toutes les fonctions sur $\mathcal{X}$ vers $\mathcal{Y}$ celle minimisant le risque est appelée* classifieur de Bayes *:*

$$s = \operatorname*{argmin}_f R(f).$$

Le classifieur de Bayes est la meilleure fonction qu'il est possible de trouver pour effectuer la discrimination. Dans le cadre où nous nous plaçons, la minimisation de cette fonctionelle est impossible puisque la distribution P est inconnue. Toutefois, nous avons accès à la mesure aléatoire P_m (distribution empirique) définie par

$$P_m = \frac{1}{m}\sum_{i=1}^{m} \delta_{(X_i,Y_i)}$$

avec $\delta_{(X_i,Y_i)}$ une masse ponctuelle en (X_i, Y_i). Cette mesure aléatoire permet la définition du risque empirique.

Définition 4 (Risque empirique). *Pour f de $\mathcal{X}$ vers $\mathcal{Y}$, le risque empirique $R_m(f)$ est l'espérance de la fonction de perte respectivement à la distribution empirique, soit*

$$R_m(f) = \mathbb{E}_{(X,Y)\sim P_m}\left[\ell\left(Y, f\left(X\right)\right)\right] = \frac{1}{m}\sum_{i=1}^{m} \ell\left(Y_i, f\left(X_i\right)\right).$$

La minimisation empirique du risque est le principe inductif de base pour lequel on sait que, pour certaines classes de fonctions, on dispose d'un résultat de convergence presque sûrement uniforme [1] [104]. L'apprentissage s'appuie donc sur une classe de fonctions $\mathcal{F}$ donnée. Il est

1. Ce résultat est une extension du théorème de Glivenko-Cantelli.

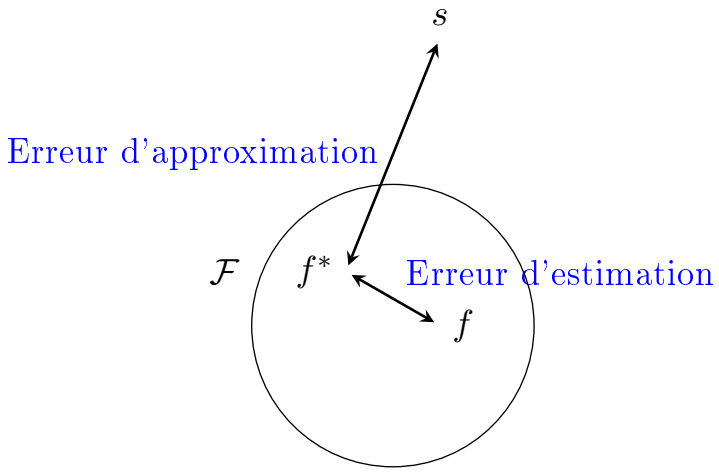


FIGURE 1.1 – Décomposition du risque en erreur d'estimation et erreur d'approximation

important de noter que l'on ne fait pas l'hypothèse que le classifieur de Bayes appartient à $\mathcal{F}$. Dans cette classe de fonction, le meilleur choix possible est

$$f^* = \operatorname*{arginf}_{f \in \mathcal{F}} R(f).$$

La fonction f obtenue par minimisation empirique du risque peut être arbitrairement loin du classifieur de Bayes. En effet, l'écart de risque entre cette fonction et le classifieur de Bayes peut s'écrire comme la somme de l'erreur d'approximation et de l'erreur d'estimation. La première correspond à l'écart de risque entre la meilleure des fonctions (f^*) de la classe et le classifieur de Bayes (s). La seconde quantifie l'écart entre la meilleure fonction (f^*) et celle fournie par le principe inductif. Ces notions d'erreurs d'estimation et d'approximation sont à mettre en lien avec le dilemme biais-variance [91].

La figure 1.1 les illustre. Elle met en évidence le compromis existant entre ces deux quantités : plus $\mathcal{F}$ est riche, plus l'erreur d'approximation devrait être faible, mais plus l'erreur d'estimation risque d'être grande (comme par exemple le sur-apprentissage), et inversement.

Toutefois, les résultats théoriques obtenus sur la minimisation empirique du risque permettent d'aller plus loin. En effet, à taille d'échantillon fixée, pour les classifieurs à marge, le risque est borné presque sûrement [58]

$$R(f) \leqslant R_{\gamma,m}(f) + \Phi(m, d, \gamma, \delta)$$

avec $R_{\gamma,m}(f)$ le risque empirique à marge (un majorant du risque empirique) et Φ un terme de contrôle croissant avec la complexité de la classe de fonctions considérée. Lorsque l'on travaille sur de petits échantillons, le risque empirique peut être faible devant Φ. Afin de minimiser le risque, il semble alors naturel de chercher une fonction qui soit en même temps bien adaptée aux données et suffisamment *douce*.

Cette approche est notamment utilisée pour les machines à vecteurs support. La fonctionelle (appelée fonction objectif) à minimiser n'est plus seulement le risque empirique mais la somme d'un risque empirique convexifié et d'un terme de pénalisation. Cette formulation du problème d'apprentissage rentre dans le cadre de la régularisation à la Tikhonov [102]. La pénalisation choisie est inversement proportionelle à la *simplicité* de la fonction. L'importance relative de ces deux grandeurs est contrôlée par l'intermédiaire du coefficient de régularisation.

Les sections 2 et 3 sont dédiées à la recherche d'un *bon* compromis entre l'adéquation aux données et la simplicité de la solution, c'est-à-dire au **choix du coefficient de régularisation**.

Dans le cadre de la sélection de modèle, nous cherchons à faire dépendre cette valeur des données. C'est cette approche que nous traiterons dans nos méthodes de sélection de modèles par parcours du chemin de régularisation.

1.2 Problèmes d'apprentissage des machines à vecteurs support bi-classes

Nous commençons par définir l'espace fonctionnel sur lequel opèrent les machines à vecteurs support bi-classes puis présentons les problèmes d'apprentissage de ces machines.

1.2.1 Espace fonctionnel

Nous considérons un problème de discrimination binaire (calcul d'une dichotomie), pour lequel l'ensemble des catégories $\mathcal{Y}$ est identifiable à $\{-1, +1\}$.

L'approche consiste en deux étapes :

- rechercher une fonction g, le classifieur, à valeurs réelles (d'une famille de fonctions $\mathcal{G}$) minimisant un risque empirique convexifié et pénalisé,
- construire la règle de décision d_g à partir de g.

Cette règle de décision d_g est définie par :

$$\begin{cases} g(x) < 0 \Longleftrightarrow d_g(x) = -1 \\ g(x) = 0 \Longleftrightarrow d_g(x) = * \\ g(x) > 0 \Longleftrightarrow d_g(x) = 1 \end{cases} .$$

L'introduction de la classe artificielle $*$ permet de prendre en compte les cas d'indétermination. Par abus de langage, le terme classifieur pourra être utilisé pour ces deux fonctions quand le contexte ne prête pas à confusion. En conséquence, l'exemple $(x, y) \in \mathcal{X} \times \mathcal{Y}$ est bien classé par d_g si et seulement si $yg(x) > 0$. Pour choisir la fonction g nous disposons de $d_m = ((x_i, y_i))_{1 \leqslant i \leqslant m}$ une réalisation de D_m.

Dans la suite de ce chapitre, $\mathcal{G}$ est instanciée par la classe $\mathcal{H}$ des fonctions calculées par une SVM. Cette classe de fonctions, construite à partir d'une fonction de type positif à valeurs réelles κ [13] et de l'espace de Hilbert à noyau reproduisant (Reproducing Kernel Hilbert Space, noté

RKHS) correspondant [1] $(\mathbf{H}_\kappa, \langle\cdot,\cdot\rangle_\kappa)$, est $\mathcal{H} = (\mathbf{H}_\kappa, \langle\cdot,\cdot\rangle_\kappa) + \{1\}$ avec $\{1\}$ l'espace des fonctions à valeurs constantes sur $\mathcal{X}$. La propriété de reproduisance permet d'écrire les fonctions h de $\mathcal{H}$ comme des fonctions affines sur ce RKHS :

$$\forall h \in \mathcal{H}, \exists \left(\bar{h}, b\right) \in \mathbf{H}_\kappa \times \mathbb{R} \;/\; \forall x \in \mathcal{X},\;\; h\left(x\right) = \bar{h}\left(x\right) + b = \langle \bar{h}, \kappa_x \rangle_\kappa + b$$

avec $\kappa_x = \kappa\left(x, \cdot\right) \in \mathbf{H}_\kappa$. Nous notons $K \in \mathcal{M}_{m,m}\left(\mathbb{R}\right)$ la matrice de Gram de terme général $K_{i,j} = \kappa\left(x_i, x_j\right)$ pour i et j entre 1 et m. Dans la littérature, les notions d'espace de représentation (*feature space*) et d'application de représentation (*feature map*) sont couramment utilisées pour décrire les machines à vecteurs support. Pour un noyau donné, il n'y a pas unicité de l'espace de représentation et nous choisissons de travailler avec le RKHS. Nous préférons la fonction $\bar{h}$ du RKHS à la fonction $\langle w, \cdot\rangle_\kappa$ (communément rencontrée dans la littérature) afin d'éviter une référence explicite à un espace de représentation spécifique.

1.2.2 ℓ_1-SVM

En 1992, les auteurs de [15] présentent l'essentiel des concepts des machines à vecteurs support. [32] complète [15] en introduisant pour la première fois le nom de machine à (réseau de) vecteurs support et la notion de marge douce. En raison de ses bonnes performances et de ses fondements théoriques, la ℓ_1-SVM a retenu l'attention de la communauté d'apprentissage. Son apprentissage s'effectue en résolvant un problème de programmation quadratique convexe, dont la définition est la suivante :

Définition 5 (ℓ_1-SVM)**.** *Soit $\mathcal{X}$ un ensemble non vide. Soit κ une fonction de type positif sur $\mathcal{X}^2$ à valeurs réelles. Soit $\mathbf{H}_\kappa$ et $\mathcal{H}_\kappa$ les deux classes de fonction induite par κ décrite ci-dessus. Soit $P_{\mathbf{H}_\kappa}$ l'opérateur de projection orthogonale de $\mathcal{H}_\kappa$ sur $\mathbf{H}_\kappa$. Pour $m \in \mathbb{N}^*$, Soit $d_m = ((x_i, y_i))_{1 \leqslant i \leqslant m} \in (\mathcal{X} \times \mathcal{Y})^m$ l'ensemble d'apprentissage. Une ℓ_1-SVM munie du noyau κ est un modèle discriminant à grande marge entraîné par résolution d'un problème de programmation quadratique convexe de la forme*

Problème 1 (Problème d'apprentissage d'une ℓ_1-SVM, formulation primale)**.**

$$\min_{h,\xi} J_{1,p}\left(h, \xi\right) = \|\xi\|_1 + \frac{\lambda}{2}\left\|P_{\mathbf{H}_\kappa} h\right\|_\kappa^2$$

$$s.c. \begin{cases} \forall i \in [\![1, m]\!],\;\; y_i h\left(x_i\right) \geqslant 1 - \xi_i \\ \forall i \in [\![1, m]\!],\;\; \xi_i \geqslant 0 \end{cases}$$

où $\lambda \in \mathbb{R}_+^*$ le coefficient de régularisation et $\xi \in \mathbb{R}^m$ le vecteur des variables d'écart.

Par la suite, puisque $P_{\mathbf{H}_\kappa} h = \bar{h}$, nous n'utilisons plus l'opérateur de projection mais directement $\bar{h}$. Le vecteur $\xi = (\xi_i)_{1 \leqslant i \leqslant m} \in \mathbb{R}_+^m$ permet la relaxation des contraintes de bon classement.

1. L'existence d'un RKHS pour lequel κ est un noyau reproduisant est prouvée par le théorème de Moore-Aronszajn [8].

Il est possible de reformuler le problème 1 sans les contraintes puisqu'à l'optimum nous avons : $\xi_i = (1 - y_i h(x_i))_+$ avec $(\cdot)_+$ la fonction égale à l'identité sur $\mathbb{R}_+$ et retournant 0 ailleurs.

Plutôt que de résoudre directement le problème 1, on préfère ordinairement résoudre son dual. Celui-ci s'obtient par application de la dualité lagrangienne. Soit $\beta = (\beta_i)_{1 \leqslant i \leqslant m} \in \mathbb{R}_+^m$ le vecteur des multiplicateurs de Lagrange associés aux contraintes de bon classement et $\theta = (\theta_i)_{1 \leqslant i \leqslant m} \in \mathbb{R}_+^m$ le vecteur des multiplicateurs de Lagrange associés à la positivité des variables d'écart. Dans la suite, nous noterons respectivement $\xi(\lambda), \beta(\lambda)$ et $\theta(\lambda)$ les valeurs optimales de ξ, β et θ, afin d'expliciter leur dépendance à λ et nous utiliserons h_λ (respectivement $\bar{h}_\lambda$ et b_λ) pour designer la valeur optimale de h (respectivement la valeur optimale de $\bar{h}$ et b). La fonction lagrangienne du problème 1 est :

$$L_1\left(\bar{h}, b, \xi, \beta, \theta\right) = 1_m^T \xi + \frac{\lambda}{2}\|\bar{h}\|_\kappa^2 - \sum_{i=1}^m \beta_i \left[y_i\left(\bar{h}(x_i) + b\right) - 1 + \xi_i\right] - \theta^T \xi.$$

A l'optimum, le gradient de la fonction lagrangienne par rapport aux variables primales est nul. L'indice du gradient correspond à la variable par rapport à laquelle nous prenons le gradient. De

$$\begin{aligned}
\nabla_{\bar{h}} L_1\left(\bar{h}, b, \xi, \beta, \theta\right) &= \lambda \bar{h} - \sum_{i=1}^m \beta_i y_i \frac{\partial \bar{h}(x_i)}{\partial \bar{h}} \\
&= \lambda \bar{h} - \sum_{i=1}^m \beta_i y_i \frac{\partial \langle \bar{h}, \kappa_{x_i} \rangle}{\partial \bar{h}} \\
&= \lambda \bar{h} - \sum_{i=1}^m \beta_i y_i \kappa_{x_i}
\end{aligned}$$

de

$$\frac{\partial}{\partial b} L_1\left(\bar{h}, b, \xi, \beta, \theta\right) = -\sum_{i=1}^m \beta_i y_i$$

et de

$$\nabla_\xi L_1\left(\bar{h}, b, \xi, \beta, \theta\right) = 1_m - \beta - \theta$$

on déduit :

$$\bar{h}_\lambda = \frac{1}{\lambda} \sum_{i=1}^m \beta_i(\lambda) y_i \kappa_{x_i}, \tag{1.1}$$

$$\sum_{i=1}^m \beta_i(\lambda) y_i = 0 \tag{1.2}$$

et

$$1_m - \beta(\lambda) - \theta(\lambda) = 0_m. \tag{1.3}$$

La substitution des équations (1.1) à (1.3) dans le lagrangien permet d'obtenir, à une multi-

plication par λ près, l'expression de la fonction objectif du dual, soit :

$$J_{1,d}(\beta) = -\frac{1}{2}\beta^T H\beta + \lambda 1_m^T \beta,$$

où H est la matrice de $\mathcal{M}_{m,m}(\mathbb{R})$ de terme général

$$\forall (i,j) \in [\![1,m]\!]^2,\ h_{ij} = y_i y_j \kappa(x_i, x_j).$$

Les contraintes du dual se déduisent de (1.2) et (1.3), en tenant compte de la positivité des multiplicateurs de Lagrange. En posant $y = (y_i)_{1 \leqslant i \leqslant m}$, on obtient en définitive :

Problème 2 (Problème d'apprentissage d'une ℓ_1-SVM exprimée avec λ, formulation duale)**.**

$$\max_{\beta} J_{1,d}(\beta) = -\frac{1}{2}\beta^T H\beta + \lambda 1_m^T \beta$$

$$s.c.\ \begin{cases} \forall i \in [\![1,m]\!],\ \ 0 \leqslant \beta_i \leqslant 1 \\ y^T\beta = 0 \end{cases}.$$

Naturellement, ce problème dual est équivalent à celui obtenu en écrivant la fonction objectif sous sa forme standard : $\frac{1}{2}\left\|\bar{h}\right\|_\kappa^2 + C\sum_{i=1}^m \xi_i$ avec $C = \frac{1}{\lambda}$.

Problème 3 (Problème d'apprentissage d'une ℓ_1-SVM exprimée avec C, formulation duale)**.**

$$\max_{\alpha} \left\{ -\frac{1}{2}\alpha^T H\alpha + 1_m^T \alpha \right\}$$

$$s.c.\ \begin{cases} \forall i \in [\![1,m]\!],\ \ 0 \leqslant \alpha_i \leqslant C \\ y^T\alpha = 0 \end{cases}.$$

Le lien entre les deux vecteurs de multiplicateurs de Lagrange est simplement $\beta = \lambda\alpha$. Comme dans le cas du vecteur β, nous notons $\alpha(\lambda)$ la valeur optimale de α. Pour une valeur de λ donnée, nous avons à l'optimum

$$\forall x \in \mathcal{X},\ \ h_\lambda(x) = \langle \bar{h}_\lambda, \kappa_x \rangle_\kappa + b_\lambda = \sum_{i=1}^m \alpha_i(\lambda) y_i \kappa(x_i, x) + \alpha_0(\lambda) = \frac{1}{\lambda}\left(\sum_{i=1}^m \beta_i(\lambda) y_i \kappa(x_i, x) + \beta_0(\lambda) \right). \tag{1.4}$$

Les conditions de Kuhn-Tucker complémentaires s'expriment sous la forme :

$$\forall i \in [\![1,m]\!],\ \begin{cases} \beta_i(\lambda)\left(y_i h_\lambda(x_i) - 1 + \xi_i(\lambda)\right) = 0 \\ \theta_i(\lambda)\,\xi_i(\lambda) = 0 \end{cases}.$$

Compte tenu de l'équation (1.3), les conditions de Kuhn-Tucker vérifiées par les exemples pour lesquels $\beta_i(\lambda) \in]0,1[$ permettent de déterminer la valeur de $b_\lambda = \alpha_0(\lambda) = \frac{1}{\lambda}\beta_0(\lambda)$.

1.2.3 ℓ_2-SVM

La ℓ_2-SVM se distingue de la ℓ_1-SVM par sa fonction objectif qui fait intervenir le carré de la norme ℓ_2 sur le vecteur ξ des variables d'écart au lieu de la norme ℓ_1. Nous commençons par ré-établir l'équivalence entre l'apprentissage de cette machine et l'apprentissage d'une machine à marge dure en posant un premier changement de noyau (cf. problème 5). Cette formulation à marge dure est une caractéristique importante de la machine de norme 2. En fait, il s'agit même de sa caractéristique principale. En effet, on dispose d'une borne sur l'erreur de validation croisée *leave-one-out* de la machine à marge dure. Une fois la formulation à marge dure obtenue, nous effectuons un changement de variable nécessaire pour faciliter les calculs permettant d'obtenir les multiplicateurs de Lagrange lors du parcours du chemin de régularisation. Pour conserver une formulation à marge dure malgré le changement de variable, nous changeons à nouveau de noyau (problème 6). Nous conservons cette formulation pour le reste du manuscrit.

L'apprentissage d'une ℓ_2-SVM correspond à la résolution d'un problème de programmation quadratique très proche dans sa formulation de celui de l'apprentissage d'une ℓ_1-SVM. Toutefois, la positivité des variables d'écart ne fait plus l'objet de contraintes. On en comprend facilement la raison, dans la mesure où une variable d'écart négative (pour un point bien classé) aurait pour seule conséquence l'augmentation de la valeur de la fonction objectif (sans améliorer la satisfaction des contraintes de bon classement). On retrouve naturellement ce résultat en appliquant la dualité lagrangienne aux problèmes avec et sans cette contrainte puisqu'ils sont associés au même problème dual. Cette remarque nous permet d'écrire la formulation primale de l'apprentissage d'une ℓ_2-SVM.

Problème 4 (Problème d'apprentissage d'une ℓ_2-SVM, formulation primale)**.**

$$\min_{h,\xi} J_{2,p}(h,\xi) = \|\xi\|_2^2 + \frac{\lambda}{2}\left\|\bar{h}\right\|_\kappa^2$$

$$s.c.\ \forall i \in [\![1,m]\!],\ \ y_i h(x_i) \geqslant 1 - \xi_i.$$

Définissons alors le vecteur $\beta = (\beta_i)_{1\leqslant i\leqslant m}$ des multiplicateurs de Lagrange des contraintes de bon classement. De manière analogue à la convention utilisée dans la section 1.2.2, le vecteur $\beta(\lambda)$ (respectivement $\xi(\lambda)$) correspond au vecteur β (respectivement au vecteur ξ des variables d'écart) optimal pour le problème défini par le coefficient de régularisation λ. Le lagrangien du problème 4 s'écrit :

$$L_2\left(\bar{h}, b, \xi, \beta\right) = \sum_{i=1}^{m} \xi_i^2 + \frac{\lambda}{2}\|\bar{h}\|_\kappa^2 - \sum_{i=1}^{m} \beta_i [y_i\left(\langle \bar{h}, \kappa_{x_i}\rangle_\kappa + b\right) - 1 + \xi_i].$$

A l'optimum, le gradient du lagrangien par rapport aux variables primales est égal au vecteur nul. De

$$\frac{\partial}{\partial b} L_2\left(\bar{h}, b, \xi, \beta\right) = -y^T\beta$$

nous obtenons $y^T\beta(\lambda) = 0$ qui formera la contrainte égalité de la formulation duale. De

$$\nabla_\xi L_2\left(\bar{h}, b, \xi, \beta\right) = 2\xi - \beta$$

et

$$\nabla_{\bar{h}} L_2\left(\bar{h}, b, \xi, \beta\right) = \lambda\bar{h} - \sum_{i=1}^{m} \beta_i y_i \kappa_{x_i}$$

on déduit :

$$2\xi(\lambda) = \beta(\lambda) \tag{1.5}$$

et

$$\bar{h}_\lambda = \frac{1}{\lambda}\sum_{i=1}^{m} y_i \beta_i(\lambda)\,\kappa_{x_i}.$$

Par substitution dans l'expression du lagrangien, nous obtenons la fonction objectif de la formulation duale du problème d'apprentissage :

$$J_{2,d}(\beta) = -\frac{1}{4}\beta^T I_m \beta - \frac{1}{2\lambda}\beta^T H\beta + 1_m^T\beta.$$

Le hessien H est celui déjà défini dans la section 1.2.2. En regroupant les deux termes quadratiques, nous retrouvons la formulation duale d'une machine à marge dure en posant :

$$\forall(i,j) \in [\![1,m]\!]^2, \quad \bar{\kappa}_\lambda(x_i, x_j) = \frac{1}{\lambda}\kappa(x_i, x_j) + \frac{1}{2}\delta_{i,j}$$

avec δ le symbole de Kronecker. En posant $\bar{H}(\lambda)$ le hessien associé à $\bar{\kappa}_\lambda$, le problème devient :

Problème 5 ([Problème d'apprentissage d'une ℓ_2-SVM, formulation duale)**.**

$$\max_\beta J_{2,d}(\beta) = -\frac{1}{2}\beta^T\bar{H}(\lambda)\beta + 1_m^T\beta$$

$$s.c. \begin{cases} \forall i \in [\![1,m]\!], \quad \beta_i \geqslant 0 \\ y^T\beta = 0 \end{cases}.$$

De plus, compte tenu de (1.5) les conditions complémentaires de Kuhn-Tucker se reformulent comme suit :

$$\forall i \in [\![1,m]\!], \ \beta_i(\lambda)\left[y_i h_\lambda(x_i) - 1 + \frac{1}{2}\beta_i(\lambda)\right] = 0.$$

Dans la suite, nous allons effectuer un changement de variable qui facilitera l'écriture en forme fermée. Comme dans le cas de la ℓ_1-SVM, nous posons $\alpha(\lambda) = \frac{1}{\lambda}\beta(\lambda)$. Ce changement de variable permet de conserver la formulation à marge dure à condition d'effectuer un nouveau changement de noyau. On pose

$$\forall(i,j) \in [\![1,m]\!]^2, \kappa_\lambda(x_i, x_j) = \kappa(x_i, x_j) + \frac{\lambda}{2}\delta_{i,j}. \tag{1.6}$$

Il est important de remarquer que κ_λ ne satisfait pas la définition d'un noyau, puisqu'il n'est défini que sur la base d'apprentissage. Toutefois, par facilité de langage, nous l'appellerons noyau dans la suite du manuscrit. En notant $H(\lambda)$ le hessien associé à κ_λ, nous obtenons alors :

Problème 6 (Problème d'apprentissage d'une ℓ_2-SVM après changement de noyau, formulation duale)**.**

$$\max_{\alpha} J_{2,d}(\alpha) = -\frac{1}{2}\alpha^T H(\lambda)\alpha + 1_m^T\alpha$$

$$s.c. \begin{cases} \forall i \in [\![1,m]\!], \ \ \alpha_i \geqslant 0 \\ y^T\alpha = 0 \end{cases}.$$

La nouvelle fonction objectif est égale à $\frac{1}{\lambda}$ fois la précédente. Pour toute la suite du document, dans le cas bi-classe, on travaillera avec les variables duales du problème 6. Les équations importantes obtenues pour le problème 6 sont les suivantes :

$$\xi = \frac{1}{2}\lambda\alpha, \tag{1.7}$$

$$h_\lambda = \sum_{i=1}^{m} y_i\alpha_i(\lambda)\kappa_{x_i} + \alpha_0(\lambda). \tag{1.8}$$

avec $\alpha_0(\lambda) = b_\lambda$. La notation $\alpha_0(\lambda)$ sera particulièrement utile pour faciliter la lecture plus loin dans le manuscrit [1].

1.2.4 SVM des moindres carrés

Une SVM des moindres carrés [99] (notée LS-SVM pour Least Squares SVM) est une SVM dont le problème d'apprentissage fait intervenir la même fonction objectif que la ℓ_2-SVM, mais pour laquelle les contraintes de bon classement sont des contraintes égalités : les variables d'écart peuvent être négatives. Le problème d'apprentissage d'une LS-SVM est donc le suivant :

Problème 7 (Problème d'apprentissage d'une LS-SVM, formulation primale)**.**

$$\min_{h,\xi} \left\{ \|\xi\|_2^2 + \frac{\lambda}{2}\left\|\bar{h}\right\|_\kappa^2 \right\}$$

$$s.c. \ \ \forall i \in [\![1,m]\!], \ y_i h(x_i) = 1 - \xi_i.$$

Suykens et Vandewalle ont montré que l'écriture du lagrangien et l'application des conditions de Kuhn-Tucker permettent de réduire la résolution du problème 7 à la résolution du système linéaire suivant (équation (20) de [99]) :

$$\begin{pmatrix} 0 & y^T \\ y & H + \frac{\lambda}{2}I_m \end{pmatrix} \begin{pmatrix} \alpha_0(\lambda) \\ \alpha(\lambda) \end{pmatrix} = \begin{pmatrix} 0 \\ 1_m \end{pmatrix}.$$

1. Notamment pendant le calcul du parcours du chemin de régularisation et de l'approximation de l'erreur de validation croisée leave-one-out par les spans.

Cette machine joue un rôle important dans l'analyse du parcours du chemin de régularisation de la ℓ_2-SVM.

1.3 Problèmes d'apprentissage des machines à vecteurs support multi-classes

Nous nous plaçons dans le cadre de la discrimination à Q catégories avec $Q \in \mathbb{N} \setminus [\![0, 2]\!]$. Soit une classe $\mathcal{G}$ de fonctions de $\mathcal{X}$ vers $\mathbb{R}^Q$. A chaque classifieur g de $\mathcal{G}$ correspond la règle de décision d_g définie par :

$$\forall x \in \mathcal{X}, \begin{cases} \text{Si } \exists k \in [\![1, Q]\!] : g_k(x) > \max_{l \neq k} g_l(x), \text{ alors } d_g(x) = k \\ \text{sinon } d_g(x) = * \end{cases} \tag{1.9}$$

avec $*$ la catégorie artificielle introduite dans la section 1.2.1. Dans un premier temps, nous présentons les approches de combinaisons de SVM bi-classes avant de décrire les machines à vecteurs support multi-classes (M-SVM).

1.3.1 La combinaison de SVM bi-classes : un premier pas vers les SVM multi-classes

Les méthodes de décomposition permettent d'aborder un problème de discrimination à plusieurs catégories ($Q \geqslant 3$) comme une combinaison de problèmes de calcul de dichotomies. Nous ne traitons ici que des deux principales méthodes de décomposition et laissons de côté, par exemple, les méthodes à codes correcteurs d'erreurs.

1.3.1.1 Approches "un contre tous"

L'approche "un contre tous" est la plus simple et la plus ancienne des méthodes de décomposition. Elle consiste à utiliser un classifieur binaire (à valeurs réelles) par catégorie. Le k-ème classifieur est destiné à distinguer la catégorie d'indice k de toutes les autres. Pour affecter un exemple, on le présente donc à Q classifieurs, et la décision s'obtient en application du principe *winner-takes-all* : l'étiquette retenue est celle associée au classifieur ayant renvoyé la valeur la plus élevée. On cite ordinairement comme plus anciens travaux évoquant l'emploi de cette stratégie avec des SVM [92] (voir aussi [103]). Dans [89], les auteurs soutiennent la thèse selon laquelle cette approche, aussi simple soit-elle, lorsqu'elle est mise en œuvre avec des SVM correctement paramétrées, obtient des performances qui ne sont pas significativement inférieures à celles des autres méthodes. Il convient cependant de souligner qu'elle implique d'effectuer des apprentissages aux répartitions entre catégories très déséquilibrées, ce qui soulève souvent des difficultés pratiques.

1.3.1.2 Approches "un contre un"

Une autre méthode de décomposition, tout aussi intuitive, est la méthode "un contre un"[48]. Ordinairement attribuée à Knerr et ses coauteurs [73], elle consiste à utiliser un classifieur par couple de catégories. Le classifieur indicé par le couple (k, l) (avec $1 \leqslant k < l \leqslant Q$), est destiné à distinguer la catégorie d'indice k de celle d'indice l. Pour affecter un exemple, on le présente donc à C_Q^2 classifieurs, et la décision s'obtient habituellement en effectuant un vote majoritaire (*max-wins voting*).

Une extension des travaux de Friedman est présentée dans [67]. Les auteurs considèrent également des classifieurs binaires estimant les probabilités a posteriori des classes $P(k \mid x \in \{k, l\})$, $1 \leqslant k < l \leqslant Q$, et la synthèse des sorties s'effectue au moyen d'une méthode nommée "couplage par paire" (*pairwise coupling*), de manière à produire à nouveau des estimations des probabilités a posteriori $P(k \mid x)$ pour l'ensemble des Q catégories. Le modèle probabiliste sous-jacent est celui de Bradley-Terry [16], et la détermination de la valeur des paramètres s'obtient par application du principe de maximum de vraisemblance. Notons que cette solution, qui impose de post-traiter les sorties des SVM, semble en contradiction avec le principe même de ces machines : rechercher la frontière de décision optimale (de Bayes) et non une estimation des probabilités a posteriori. Et pourtant, les résultats expérimentaux fournis sont bons. De nombreux auteurs ont proposé des améliorations de la méthode de base, portant soit sur la façon de réaliser le couplage par paire [83, 79], soit sur l'estimation des probabilités conditionnelles $P(k \mid x, x \in \{k, l\})$. Pour produire des estimations des probabilités a posteriori d'un couple de catégories à partir des sorties de la SVM correspondante, Platt [85] propose d'utiliser un modèle paramétrique s'adaptant directement à la probabilité conditionnelle de la classe "positive" sachant la sortie de la SVM. Il considère spécifiquement le cas où ce modèle est une sigmoïde à deux paramètres A et B :

$$P\left(+1 \mid \tilde{h}(x)\right) = \frac{1}{1 + \exp\left(A\tilde{h}(x) + B\right)}.$$

Les valeurs de A et B sont déterminées par un apprentissage appliquant le principe de maximum de vraisemblance [80]. Nous utilisons cette approche dans l'annexe B comme référence pour la comparaison de l'estimation des probabilités a posteriori.

Au-delà des arguments théoriques ou empiriques avancés pour justifier l'emploi de méthodes de décomposition, une considération pratique explique leurs bonnes performances. Les experts spécifiant le problème d'apprentissage et effectuant la collecte des données ont souvent à l'esprit le schéma canonique dans lequel les catégories se trouvent être aussi indépendantes que possible les unes des autres (voir par exemple les catégories de l'annexe B). Les membres de la classe 1 sont aussi différents de ceux de la classe 2 que de ceux de la classe 3. Les cas pratiques échappant à ce schéma sont rares.

1.3.2 Modèle générique de M-SVM

Les SVM bi-classes s'interprètent facilement de manière géométrique grâce à la notion d'hyperplan de marge maximale. Hélas, dans le cadre multi-classe cette notion ne s'étend pas de manière triviale. C'est sans doute une raison pour laquelle plusieurs M-SVM [110, 33, 78, 63] ont été introduites sans qu'aucune ne s'impose. Dans un objectif de sélection de modèle, les propriétés de la ℓ_2-SVM sont très intéressantes. Malheureusement, l'extension naïve de cette machine ne permet pas de les conserver. La section 2.4.1.4 de [57] détaille ce point. La M-SVM^2 [63] conserve cette propriété d'équivalence à une machine à marge dure : elle est aux M-SVM ce que la ℓ_2-SVM est aux SVM. Nous présentons un modèle générique permettant d'englober les extensions "utiles" de la ℓ_2-SVM ainsi que les extensions de la ℓ_1-SVM. Pour cela nous nous appuyons sur [61] et introduisons une formulation générique de l'apprentissage des M-SVM.

Toutes les M-SVM sont des machines à noyau qui opèrent sur une unique classe de fonctions à valeurs dans $\mathbb{R}^Q$. Historiquement, cette classe de fonctions $\mathbf{H}_\kappa^Q \oplus \{1\}^Q$ est construite à partir de $\mathbf{H}_\kappa$. Plutôt que d'utiliser cette formulation classique, nous nous appuyons sur la définition des RKHS de fonctions à valeurs vectorielles introduite par Wahba [106] afin de justifier le choix du pénalisateur utilisé.

Définition 6 (RKHS de fonctions à valeurs dans $\mathbb{R}^Q$ [106])**.** *Soit $\tilde{\kappa}$ une fonction de type positif sur $(\mathcal{X} \times [\![1, Q]\!])^2$ à valeurs réelles. Pour chaque (x, k) de $\mathcal{X} \times [\![1, Q]\!]$, la fonction $\tilde{\kappa}_{x,k}^{(Q)}(\cdot)$ de $\mathcal{X}$ dans $\mathbb{R}^Q$ est définie par*

$$\tilde{\kappa}_{x,k}^{(Q)}(\cdot) = \left(\tilde{\kappa}\left((x,k),(\cdot,l)\right)\right)_{1 \leqslant l \leqslant Q}.$$

Le RKHS $(\mathbf{H}_{\tilde{\kappa}^{(Q)}}, \langle \cdot, \cdot \rangle_{\tilde{\kappa}^{(Q)}})$ consiste en la variété linéaire de toutes les combinaisons linéaires finies de $\tilde{\kappa}_{x,k}^{(Q)}(\cdot)$ pour (x, k) dans $\mathcal{X} \times [\![1, Q]\!]$ et de sa fermeture au sens du produit scalaire

$$\forall \left(x, x'\right) \in \mathcal{X}^2, \forall \left(k, l\right) \in [\![1, Q]\!]^2, \langle \tilde{\kappa}_{x,k}^{(Q)}, \tilde{\kappa}_{x',l}^{(Q)} \rangle_{\tilde{\kappa}^{(Q)}} = \tilde{\kappa}\left((x,k),\left(x',l\right)\right).$$

Nous définissons à présent le RKHS de fonctions à valeurs vectorielles qui est à la base des M-SVM.

Définition 7 (RKHS $\mathbf{H}_{\kappa,Q}$)**.** *Soit κ une fonction de type positif à valeurs réelles sur $\mathcal{X}^2$. En utilisant les notations de la définition 6, le RKHS de fonction à valeurs vectorielles à la base d'une M-SVM à Q catégories munie du noyau κ, $(\mathbf{H}_{\kappa,Q}, \langle \cdot, \cdot \rangle_{\kappa,Q})$, est le RKHS $(\mathbf{H}_{\tilde{\kappa}^{(Q)}}, \langle \cdot, \cdot \rangle_{\tilde{\kappa}^{(Q)}})$ correspondant au choix du noyau $\tilde{\kappa}$:*

$$\forall \left(x, x'\right) \in \mathcal{X}^2, \ \forall \left(k, l\right) \in [\![1, Q]\!]^2, \ \ \tilde{\kappa}\left((x,k),\left(x',l\right)\right) = \delta_{k,l}\kappa\left(x, x'\right)$$

avec δ le symbole de Kronecker.

La fonction $\tilde{\kappa}$ de la définition 7 satisfait effectivement les hypothèses de la définition 6. En effet $\tilde{\kappa}$ est un noyau sur $(\mathcal{X} \times [\![1, Q]\!])^2$ comme produit tensoriel d'un noyau sur $\mathcal{X}^2$ et d'un noyau sur $[\![1, Q]\!]^2$ [94]. Par un raisonnement sur des suites de Cauchy, il est possible de montrer que

cette définition de $\mathbf{H}_{\kappa,Q}$ peut être reformulée en fonction de $\mathbf{H}_\kappa$ et donc retrouver la classe de fonctions utilisée de manière classique pour les M-SVM :

Proposition 1 (Caractérisation alternative de $\mathbf{H}_{\kappa,Q}$). *Soit κ une fonction de type positif à valeurs réelles sur $\mathcal{X}^2$ et soit $(\mathbf{H}_\kappa, \langle\cdot,\cdot\rangle_\kappa)$ le RKHS associé, alors, $\mathbf{H}_{\kappa,Q} = \mathbf{H}_\kappa^Q$. De plus le produit scalaire de $\mathbf{H}_{\kappa,Q}$ peut être exprimé comme une fonction du produit scalaire de $\mathbf{H}_\kappa$:*

$$\forall \left(\bar{h}, \bar{h}'\right) \in \mathbf{H}_{\kappa,Q}^2,\ \bar{h} = \left(\bar{h}_k\right)_{1\leqslant k\leqslant Q},\ \bar{h}' = \left(\bar{h}'_k\right)_{1\leqslant k\leqslant Q},\ \ \langle \bar{h}, \bar{h}'\rangle_{\kappa,Q} = \sum_{k=1}^{Q} \langle \bar{h}_k, \bar{h}'_k\rangle_\kappa.$$

De manière analogue au cadre fonctionnel du cas bi-classe, la classe de fonctions sur laquelle opèrent les M-SVM est obtenue par somme directe :

Définition 8 (Classe de fonctions $\mathcal{H}_{\kappa,Q}$). *Soit κ une fonction de type positif à valeurs réelles sur $\mathcal{X}^2$ et soit $\mathbf{H}_{\kappa,Q}$ le RKHS de fonction à valeurs dans $\mathbb{R}^Q$ associé à κ d'après la définition 7. Soit $\{1\}$ l'espace des fonctions à valeurs constantes sur $\mathcal{X}$. La classe de fonctions à la base des M-SVM à Q catégories dont le noyau est κ est*

$$\mathcal{H}_{\kappa,Q} = \mathbf{H}_{\kappa,Q} \oplus \{1\}^Q = \left(\mathbf{H}_\kappa \oplus \{1\}\right)^Q.$$

Les fonctions de $\mathcal{H}_{\kappa,Q}$ peuvent être vues comme des fonctions affines multivariées sur $\mathbf{H}_{\kappa,Q}$. En effet, par la propriété de reproduisance,

$$\forall h \in \mathcal{H}_{\kappa,Q},\ \forall x \in \mathcal{X},\ \ h(x) = \bar{h}(x) + b = \left(\langle \bar{h}_k, \kappa_x\rangle_{\kappa,Q} + b_k\right)_{1\leqslant k\leqslant Q},$$

où la fonction $\bar{h} = \left(\bar{h}_k\right)_{1\leqslant k\leqslant Q}$ est un élément de $\mathbf{H}_\kappa^Q$ et $b = (b_k)_{1\leqslant k\leqslant Q} \in \mathbb{R}^Q$. Rappelons que l'ensemble des exemples utilisés pour l'apprentissage est $d_m = ((x_i, y_i))_{1\leqslant i\leqslant m} \in (\mathcal{X}\times\mathcal{Y})^m$. Soit $\mathbb{R}^{Qm}(d_m)$ le sous-ensemble de $\mathbb{R}^{Qm}$ constitué des vecteurs $v = (v_t)_{1\leqslant t\leqslant Qm}$ satisfaisant :

$$\left(v_{(i-1)Q+y_i}\right)_{1\leqslant i\leqslant m} = 0_m. \tag{1.10}$$

De manière analogue, nous définissons $\mathbb{R}_+^{Qm}(d_m) = \mathbb{R}^{Qm}(d_m) \cap \mathbb{R}_+^{Qm}$. Pour garder les notations simples, les composantes des vecteurs de $\mathbb{R}^{Qm}(d_m)$ sont écrites avec deux indices, c'est-à-dire, v_{ik} à la place de $v_{(i-1)Q+k}$, pour i dans $[\![1,m]\!]$ et k dans $\mathcal{Y}$. En conséquence, (1.10) se simplifie en $(v_{iy_i})_{1\leqslant i\leqslant m} = 0_m$. De ce fait, les variables associées au couple d'indices (i, y_i) sont des variables artificielles.

Pour n dans $\mathbb{N}^*$, soit $\mathcal{M}_{n,n}(\mathbb{R})$ l'algèbre des matrices de taille $n\times n$ sur $\mathbb{R}$. Nous notons $\mathcal{M}_{Qm,Qm}(d_m)$ le sous-ensemble de $\mathcal{M}_{Qm,Qm}(\mathbb{R})$ constitué des matrices $M = (m_{tu})_{1\leqslant t,u\leqslant Qm}$ satisfaisant :

$$\forall j \in [\![1,m]\!],\ \ \left(m_{t,(j-1)Q+y_j}\right)_{1\leqslant t\leqslant Qm} = 0_{Qm}.$$

Par même souci de simplicité, les matrices de $\mathcal{M}_{Qm,Qm}(d_m)$ sont écrites avec quatre indices, c'est-à-dire, $m_{ik,jl}$ au lieu de $m_{(i-1)Q+k,(j-1)Q+l}$, pour (i,j) dans $[\![1,m]\!]^2$ et (k,l) dans $[\![1,Q]\!]^2$.

Avec ces définitions et notations à disposition, le modèle générique de M-SVM proposé dans [61] se définit ainsi :

Définition 9 (Modèle générique de M-SVM). *Soit $\mathcal{X}$ un ensemble non vide et $Q \in \mathbb{N} \setminus [\![0,2]\!]$. Soit κ une fonction de type positif à valeurs réelles sur $\mathcal{X}^2$. Soit $\mathbf{H}_{\kappa,Q}$ et $\mathcal{H}_{\kappa,Q}$ les deux classes de fonctions induites par κ d'après les définitions 7 et 8. Soit $P_{\mathbf{H}_{\kappa,Q}}$ l'opérateur de projection orthogonale de $\mathcal{H}_{\kappa,Q}$ sur $\mathbf{H}_{\kappa,Q}$. Pour $m \in \mathbb{N}^*$, soit $d_m = ((x_i, y_i))_{1 \leqslant i \leqslant m} \in (\mathcal{X} \times [\![1,Q]\!])^m$ et $\xi \in \mathbb{R}^{Qm}(d_m)$. Une M-SVM à Q catégories munie du noyau κ et de l'ensemble d'apprentissage d_m est un modèle discriminant à grande marge entraîné en résolvant un problème de programmation quadratique convexe de la forme*

Problème 8 (Problème d'apprentissage d'une M-SVM, formulation primale).

$$\min_{h,\xi} \left\{ \|M\xi\|_p^p + \lambda \|P_{\kappa,Q} h\|_{\kappa,Q}^2 \right\}$$

$$s.c. \begin{cases} \forall i \in [\![1,m]\!],\ \forall k \in [\![1,Q]\!] \setminus \{y_i\},\ \ K_1 h_{y_i}(x_i) - h_k(x_i) \geqslant K_2 - \xi_{ik} \\ \forall i \in [\![1,m]\!],\ \forall (k,l) \in ([\![1,Q]\!] \setminus \{y_i\})^2,\ \ K_3(\xi_{ik} - \xi_{il}) = 0 \\ \forall i \in [\![1,m]\!],\ \forall k \in [\![1,Q]\!] \setminus \{y_i\},\ \ (2-p)\,\xi_{ik} \geqslant 0 \\ (1-K_1)\sum_{k=1}^{Q} h_k = 0 \end{cases}$$

où $\lambda \in \mathbb{R}_+^*$, $M \in \mathcal{M}_{Qm,Qm}(d_m)$ est une matrice de rang $(Q-1)\,m$, $p \in \{1,2\}$, $(K_1, K_3) \in \{0,1\}^2$, et $K_2 \in \mathbb{R}_+^*$. Si $p = 1$, alors M est une matrice diagonale.

L'expression de la formulation duale, dont l'obtention est présentée en détail dans [61], nécessite l'introduction des matrices $N^{(-1)}$ et $\tilde{N}$. Elles sont construites à partir de M et ne sont explicitées, ici, que pour la M-SVM^2 qui est la machine d'intérêt dans ce manuscrit. De plus, l'expression de la matrice $H(K_1)$ de la fonction objectif du modèle générique ne sera pas donnée, seul le hessien de la M-SVM de Lee, lin et Wahba (LLW-M-SVM), donc par voie de conséquence de la M-SVM^2, le sera.

Soit $\beta \in \mathbb{R}_+^{Qm}(d_m)$ le vecteur des multiplicateurs de Lagrange associés aux contraintes de bon classement et soit $\tilde{\beta} = \left(\sum_{k \neq y_i} \beta_{ik}\right)_{1 \leqslant i \leqslant m}$. Le problème dual de 8 est :

Problème 9 (Problème d'apprentissage de la M-SVM à marge douce, formulation duale).

$$\max_{\beta} J_{M\text{-}SVM,d}(\beta)$$

$$s.c. \begin{cases} \forall i \in [\![1,m]\!],\ \forall k \in [\![1,Q]\!] \setminus \{y_i\},\ \ 0 \leqslant (1-K_3)(2-p)\,\beta_{ik} \leqslant (2-p)\,m_{ik,ik} \\ \forall i \in [\![1,m]\!],\ \ 0 \leqslant K_3(2-p)\sum_{k \neq y_i} \beta_{ik} \leqslant (2-p)\sum_{k \neq y_i} m_{ik,ik} \\ \forall i \in [\![1,m]\!],\ \forall k \in [\![1,Q]\!] \setminus \{y_i\},\ \ (p-1)\,\beta_{ik} \geqslant 0 \\ \forall k \in [\![1,Q-1]\!],\ \ \sum_{i=1}^{m}\sum_{l=1}^{Q} \left\{ K_1 \delta_{y_i,k} + (1-K_1)\,\frac{1}{Q} - \delta_{k,l} \right\} \beta_{il} = 0 \end{cases}$$

où

$$J_{M\text{-}SVM,d}(\beta) =$$

$$-\frac{1}{4}\left\{\beta^T\left(\frac{1}{\lambda}H(K_1)+(1-K_3)(p-1)N^{(-1)}\right)\beta+K_3(p-1)\tilde{\beta}^T\tilde{N}^{-1}\tilde{\beta}\right\}+K_2 1_{Qm}^T\beta. \quad (1.11)$$

Les quatre M-SVM principales s'expriment comme des instances de ce modèle générique 8. Les valeurs correspondantes de ces hyperparamètres sont indiquées dans le tableau 1.1.

M-SVM	M	p	K_1	K_2	K_3
WW-M-SVM	$I_{Qm}(d_m)$	1	1	1	0
CS-M-SVM	$\frac{1}{Q-1}I_{Qm}(d_m)$	1	1	1	1
LLW-M-SVM	$I_{Qm}(d_m)$	1	0	$\frac{1}{Q-1}$	0
M-SVM^2	$M^{(2)}$	2	0	$\frac{1}{Q-1}$	0

TABLE 1.1 – Paramétrage du modèle générique pour les quatre M-SVM principales. Les 3 premières sont celles de Weston et Watkins (WW), Crammer et Singer (CS), et Lee, Lin, et Wahba (LLW).

Dans ce tableau la matrice $M^{(2)}$, l'instance de M correspondant à la M-SVM^2, est de terme général

$$m^{(2)}_{ik,jl} = (1-\delta_{k,y_i})(1-\delta_{l,y_j})\left(\delta_{k,l}+\frac{\sqrt{Q}-1}{Q-1}\right)\delta_{i,j}. \quad (1.12)$$

En imposant $\beta_{i,y_i} = 0$, les valeurs des lignes et des colonnes qui étaient à zéro deviennent indéterminées. Par simplicité de notations, nous choisissons de les fixer égaux au terme général de la matrice. Nous choisissons de travailler avec ces notations compactes. La matrice $N^{(2)} = {M^{(2)}}^T M^{(2)}$ est alors de terme général

$$(\delta_{k,l}+1)\,\delta_{i,j}$$

qui s'écrit aussi

$$N^{(2)} = I_m \otimes \left(I_Q + 1_Q 1_Q^T\right)$$

avec $\otimes$ le produit de Kronecker.

La matrice $\bar{N}^{(2)}$ est définie par $\bar{N}^{(2)} = I_m \otimes \left(I_{Q-1} + 1_{Q-1}1_{Q-1}^T\right)$. Puisque les valeurs propres de $I_{Q-1} + 1_{Q-1}1_{Q-1}^T$ sont 1 et Q, $\bar{N}^{(2)}$ est inversible et son inverse est ${\bar{N}^{(2)}}^{-1} = I_m \otimes \left(I_{Q-1} - \frac{1}{Q}1_{Q-1}1_{Q-1}^T\right)$. La matrice ${N^{(2)}}^{(-1)}$, qui est l'instance de $N^{(-1)}$ pour la M-SVM^2, est construite en rajoutant des lignes et des colonnes à ${\bar{N}^{(2)}}^{-1}$ de manière à avoir ${N^{(2)}}^{(-1)}$ égale à $I_m \otimes \left(I_{Q-1} - \frac{1}{Q}1_{Q-1}1_{Q-1}^T\right)$. Par souci de simplicité de notation puisqu'il ne sera traité que de la M-SVM^2 dans ce manuscrit, la matrice $M^{(2)}$ sera remplacée par M.

Dans le cadre de la sélection de modèle, notamment pour l'utilisation de la borne Rayon-

Marge, la formulation à marge dure est nécessaire.

Définition 10 (M-SVM à marge douce et à marge dure)**.** *Une M-SVM entraînée sous la contrainte d'une adéquation complète aux données ($\xi = 0_{Qm}$) est dite à* marge dure*. Dans le cas contraire, elle est dite à* marge douce*.*

La formulation duale du problème d'apprentissage du modèle générique de M-SVM à marge dure est :

Problème 10 (Problème d'apprentissage de la M-SVM à marge dure, formulation duale)**.**

$$\max_{\beta} \left\{ -\frac{1}{4\lambda}\beta^T H \beta + K_2 1_{Qm}^T \beta \right\}$$

$$s.c. \begin{cases} \forall i \in [\![1, m]\!],\ \forall k \in [\![1, Q]\!] \setminus \{y_i\},\ \ \beta_{ik} \geqslant 0 \\ \forall k \in [\![1, Q-1]\!],\ \ \sum_{i=1}^{m} \sum_{l=1}^{Q} \left\{ K_1 \delta_{y_i,k} + (1 - K_1)\frac{1}{Q} - \delta_{k,l} \right\} \beta_{il} = 0 \end{cases} .$$

Comme nous l'avons vu dans la sous-section 1.2.3, la formulation duale de la ℓ_2-SVM correspond à la formulation duale d'une machine à marge dure "à un changement de noyau près" [1]. Ainsi son erreur de validation croisée leave-one-out peut être bornée par la borne Rayon-Marge [104]. La littérature fournit une telle machine pour le cas multi-classe.

1.3.3 M-SVM^2

La M-SVM^2 a précisément été conçue afin d'obtenir une machine dont la formulation duale du problème d'apprentissage soit équivalente à celle du problème d'apprentissage d'une machine à marge dure.

Proposition 2. *La formulation duale du problème d'apprentissage d'une M-SVM^2 est identique à la formulation duale du problème d'apprentissage d'une LLW-M-SVM à marge dure (à un changement de noyau près).*

Puisque nous allons étudier la sélection de modèle pour la M-SVM^2, il est intéressant de détailler ce point.

Démonstration. La réécriture du problème 10 correspondant à la LLW-M-SVM à marge dure est :

Problème 11 (Problème d'apprentissage d'une LLW-M-SVM, formulation duale)**.**

$$\max_{\beta} \left\{ -\frac{1}{4}\beta^T H \beta + \frac{1}{Q-1} 1_{Qm}^T \beta \right\}$$

$$s.c. \begin{cases} \forall i \in [\![1, m]\!],\ \forall k \in [\![1, Q]\!] \setminus \{y_i\},\ \ \beta_{ik} \geqslant 0 \\ \forall k \in [\![1, Q-1]\!],\ \ \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\frac{1}{Q} - \delta_{k,l} \right) \beta_{il} = 0 \end{cases}$$

1. Nous utiliserons couramment cet abus de langage.

avec pour terme général de la matrice Hessienne H :

$$h_{ik,jl} = \left(\delta_{k,l} - \frac{1}{Q}\right) \kappa\left(x_i, x_j\right).$$

La réécriture du problème 9 correspondant à la M-SVM^2 est :

Problème 12 (Problème d'apprentissage d'une M-SVM^2, formulation dual)**.**

$$\max_{\beta} \left\{ -\frac{1}{4\lambda}\beta^T H \beta - \frac{1}{4}\beta^T N^{(-1)} \beta + \frac{1}{Q-1} 1_{Qm}^T \beta \right\}$$

$$s.c. \begin{cases} \forall i \in [\![1, m]\!],\ \forall k \in [\![1, Q]\!] \setminus \{y_i\},\ \ \beta_{ik} \geqslant 0 \\ \forall k \in [\![1, Q-1]\!],\ \ \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\frac{1}{Q} - \delta_{k,l}\right) \beta_{il} = 0 \end{cases}$$

avec H la matrice du problème 11 et N la matrice de terme général

$$n_{ik,jl} = \left(\delta_{k,l} + 1\right) \delta_{i,j}.$$

A la lumière de la fin de la section 1.3.2, on remarque que $n_{ik,jl}^{(-1)}$ est égal à $h_{ik,jl}$ si $\kappa\left(x_i, x_j\right)$ est remplacé par $\delta_{i,j}$. En effet, si nous définissons ainsi le noyau κ' :

$$\forall \left(i, j\right) \in [\![1, m]\!]^2,\ \ \kappa'\left(x_i, x_j\right) = \lambda \delta_{i,j}, \tag{1.13}$$

nous obtenons alors

$$\frac{1}{4\lambda}\beta^T H \beta + \frac{1}{4}\beta^T N^{(-1)} \beta = \frac{1}{4\lambda}\beta^T H'' \beta,$$

où H'' est la matrice déduite de H en remplaçant le noyau κ par le noyau $\kappa'' = \kappa + \kappa'$. Il en résulte que les problèmes 11 et 12 sont identiques à un changement de noyau prés, ce qui conclut la preuve. □

Une équation, valable pour la LLW-M-SVM à marge dure, qui sera utile dans le chapitre 3 est :

$$\|\bar{h}^*\|_{\kappa,Q}^2 = \sum_{k=1}^{Q} \|\bar{h}_k^*\|_{\kappa}^2 = \beta\left(\lambda\right)^T H \beta\left(\lambda\right) = \frac{1}{Q-1} 1_{Qm}^T \beta\left(\lambda\right) \tag{1.14}$$

avec $\left(h_k^*\right)_{1 \leqslant k \leqslant Q} = \left(\frac{1}{\lambda} \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\frac{1}{Q} + \delta_{k,l}\right) \beta\left(\lambda\right)_{il} \kappa_{x_i} + b_k\right)_{1 \leqslant k \leqslant Q}$ la solution optimale du problème 11. L'exposant * dénotera toujours la solution optimale du problème d'intérêt lorsque sa dépendance à λ ne porte pas à confusion. Ce changement de noyau est le même que celui du cas bi-classe [1] (1.6), et donc nous posons, pour le cas multi-classe :

$$\forall \left(i, j\right) \in [\![1, m]\!]^2, \kappa_{\lambda}\left(x_i, x_j\right) = \kappa\left(x_i, x_j\right) + \lambda \delta_{i,j}.$$

1. Penser au changement de variable $\lambda' = \frac{\lambda}{2}$

Pour la LLW-M-SVM à marge dure, la sélection des hyperparamètres est possible à l'aide de la borne Rayon-Marge multi-classe. Puisqu'une formulation à marge dure existe pour la M-SVM^2, la même méthode de sélection est applicable (voir annexe A).

De manière analogue au changement de variable du cas bi-classe, nous introduisons $\alpha = \frac{1}{2\lambda}\beta$ de manière à ce que λ n'intervienne plus qu'une seule fois dans la fonction objectif [1]. L'apparition du facteur un demi dans le changement de variable, par rapport au cas bi-classe, est due au choix d'utiliser la formule du modèle générique ne faisant pas intervenir de coefficient un demi devant le pénalisateur. La formulation duale du problème d'apprentissage de la M-SVM^2 s'écrit alors :

Problème 13 (Problème d'apprentissage de la M-SVM^2, formulation duale)**.**

$$\max_{\alpha} J_{M\text{-}SVM^2,d}(\alpha)$$

$$s.c. \begin{cases} \forall i \in [\![1,m]\!],\ \forall k \in [\![1,Q]\!] \setminus \{y_i\},\ \ \alpha_{ik} \geqslant 0 \\ \forall k \in [\![1,Q-1]\!],\ \ \sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q}-\delta_{k,l}\right)\alpha_{il} = 0 \end{cases}$$

avec

$$J_{M\text{-}SVM^2,d}(\alpha) = -\frac{1}{2}\alpha^T H(\lambda)\,\alpha + \frac{1}{Q-1}1_{Qm}^T\alpha,$$

où $H(\lambda)$ *est la matrice de* $\mathcal{M}_{Qm,Qm}(\mathbb{R})$ *de terme général*

$$h_{ik,jl}(\lambda) = \left(\delta_{k,l} - \frac{1}{Q}\right)(\kappa(x_i,x_j) + \lambda\delta_{i,j}).$$

La sous-section suivante est dédiée à la question de la sélection de modèle pour les SVM.

1.4 Sélection de modèle appliquée aux SVM

Pour les machines à vecteurs support, la sélection de modèle correspond au choix du noyau et à la détermination de la valeur des hyperparamètres, qui sont de deux types : le coefficient de régularisation λ et les paramètres du noyau. L'estimation de l'erreur est un premier pas vers la sélection de modèle. Nous présentons, dans un premier temps, les méthodes usuelles d'estimation de l'erreur en généralisation. Les solutions dédiées à la sélection de modèle pour les SVM bi-classes sont ensuite détaillées. La dernière partie présente des résultats spécifiques aux M-SVM.

1.4.1 Méthodes classiques d'estimation de l'erreur en généralisation

Les trois principales méthodes de sélection de modèle sont le *hold-out*, la validation croisée à V pas et le bootstrap. Les critères de la statistique classique, tel que AIC [1] et BIC [95], ne sont pas habituellement appliqués aux SVM [7].

1. Cette fonction objectif est multipliée $1/2\lambda$

1.4.1.1 Validation *hold-out*

La procédure de validation *hold-out* consiste à définir un ensemble d'apprentissage et un ensemble de validation. Le premier ensemble sert à entraîner le classifieur alors que le second permet d'évaluer les performances en généralisation. Cette procédure n'est intéressante que lorsque beaucoup d'exemples sont à disposition. En effet, son principal défaut est de ne pas tirer le meilleur parti des données. Une possibilité pour cela consiste à faire *tourner* le jeu de validation : il s'agit de la validation croisée.

1.4.1.2 Validation croisée à V pas

Le principe de la validation croisée à V pas consiste à partitionner l'ensemble d'apprentissage en V parts approximativement égales. L'apprentissage se fait sur $V-1$ parts alors que le test se fait sur celle restante. Ce processus est répété pour chaque part. Le taux d'erreur global est appelé l'erreur de validation croisée à V pas que nous notons $\hat{Err}^{(cv,V/m)}$. L'erreur de validation croisée à V pas est couramment utilisée dans une optique de sélection de modèle [97, 18]. En pratique, V est souvent fixé à 5, 7 ou 10. Dans ces conditions, cet estimateur présente une faible variance mais un biais élevé [12].

Lorsque $V = m$, cette procédure porte le nom de validation croisée *leave-one-out*. En nous inspirant de la notation de [41], nous notons $\hat{Err}^{(cv,1)}$ la fréquence d'erreur de validation croisée leave-one-out. Cet estimateur possède la particularité d'être presque sans biais [104] :

Théorème 1 (Théorème 10.8 de [104])**.** *L'estimateur de la probabilité d'erreur fourni par la validation croisée* leave-one-out *est presque sans biais au sens où*

$$\mathbb{E}\left(p_{erreur}^{m-1}\right) = \mathbb{E}\left(\hat{Err}^{(cv,1)}\right) \tag{1.15}$$

où p_{erreur}^{m-1} correspond à la probabilité pour que le classifieur entraîné sur un échantillon de taille $m-1$ commette une erreur.

Cette absence de biais, au prix d'une variance élevée, en fait un bon outil de sélection de modèle. Pour des raisons pratiques, la validation à petit nombre de pas est plus souvent utilisée car la complexité de la procédure de validation croisée leave-one-out est au pire pour les SVM en $O\left(m^4\right)$. Nous verrons par la suite que l'utilisation de bornes ou d'estimations de cette quantité permet de diminuer cette complexité.

1.4.1.3 Bootstrap .632+

La procédure de bootstrap .632+ permet d'obtenir un estimateur de l'erreur en généralisation possédant en même temps un faible biais et une faible variance [41]. Cette procédure s'étend au cas multi-classe mais, par souci de simplicité, ne sera détaillée ici que dans le cas bi-classe. Elle se découpe en deux parties :

- le calcul d'un estimateur appelé leave-one-out bootstrap, qui est biaisé par valeur supérieure,
- la correction de cet estimateur permettant d'obtenir l'estimateur de bootstrap .632+.

Leave-one-out bootstrap : Soit B m-échantillons obtenus par re-échantillonage de l'ensemble d'apprentissage. Nous appelons échantillons de validation les échantillons complémentaires de chacun de ces B échantillons. L'estimateur de bootstrap leave-one-out correspond au calcul de la moyenne des erreurs de validation d'un classifieur entraîné sur ces B m-échantillons. Pour cette sous-section, nous notons $D = (Z_i)_{1 \leqslant i \leqslant m}$ l'échantillon d'apprentissage avec $Z = (X, Y)$ le couple aléatoire distribué selon P. Soit f_D la règle de décision apprise sur le m-échantillon d'apprentissage D.

L'erreur en généralisation de cette règle de décision f_D par rapport à la distribution P est :

$$R(f_D) = \mathbb{E}_{Z \sim P}[\mathbb{1}_{\{Y \neq f_D(X)\}}] \tag{1.16}$$

où D (et donc f_D) est fixé et $\mathbb{1}$ la fonction indicatrice. $R(f_D)$ est donc l'erreur en généralisation conditionnelle à l'ensemble d'apprentissage D. Le taux d'erreur apparent (aussi nommé taux d'erreur en resubstitution) est :

$$R_m(f_D) = \mathbb{E}_{Z \sim \hat{P}}[\mathbb{1}_{\{Y \neq f_D(X)\}}] = \frac{1}{m} \sum_{p=1}^{m} \mathbb{1}_{\{Y_p \neq f_D(X_p)\}} \tag{1.17}$$

avec $\hat{P}$ la distribution empirique qui associe une probabilité de $1/m$ à chacune des observations.

La méthode du bootstrap se base sur des *échantillons de bootstrap*. Nous notons D^* un tel m-échantillon qui suit la loi $\hat{P}$. Pour la méthode du *bootstrap leave-one-out*, de la même manière que pour la validation croisée *leave-one-out*, le classifieur est testé sur des données n'ayant pas servi à l'apprentissage. Soit $D^{*(p)}$ est un échantillon de bootstrap tiré suivant la distribution empirique $\hat{P}^{(p)}$, distribution uniforme sur $\{Z_j \ : \ 1 \leqslant j \neq p \leqslant m\}$. L'estimateur de *bootstrap leave-one-out* est alors défini par :

$$\hat{Err}^{(1)} = \frac{1}{m} \sum_{p=1}^{m} \mathbb{E}_{D^{*(p)} \sim \hat{P}^{(p)m}}[\mathbb{1}_{\{Y_p \neq f_{D^{*(p)}}(X_p)\}}] \tag{1.18}$$

L'estimation de $\mathbb{E}_{D^{*(p)} \sim \hat{P}^{(p)m}}[\mathbb{1}_{\{Y_p \neq f_{D^{*(p)}}(X_p)\}}]$ est obtenue en tirant avec remise B m-échantillons dans $\{Z_1, \cdots, Z_m\}$. Soient $D^{*1}, \cdots, D^{*B}$ ces B m-échantillons. L'absence de Z_p dans le b-ème échantillon de bootstrap s'écrit

$$\mathbb{1}_{\{Z_p \notin D^{*b}\}}$$

ce qui donne

$$\hat{Err}^{(1)} = \frac{1}{m}\sum_{p=1}^{m}\frac{\sum_b \mathbb{1}_{\{Z_p \notin D^{*b} \wedge Y_p \neq f_{D^{*b}}(X_p)\}}}{\sum_b \mathbb{1}_{\{Z_p \notin D^{*b}\}}}. \tag{1.19}$$

De part sa complexité, le bootstrap n'est pas couramment utilisé pour les SVM. Dans [6], le nombre de réplications a été fixé à 1000 entraînant un temps de calcul beaucoup plus important que pour une validation croisée à V pas.

L'estimateur de bootstrap .632+ Il a été montré dans [43] que l'estimateur de bootstrap leave-one-out est biaisé par valeur inférieure. Pour pallier ce défaut, l'estimateur de bootstrap .632 est défini par :

$$\hat{Err}^{(.632)} = 0.368 R_m\left(f_D\right) + 0.632\hat{Err}^{(1)}.$$

Le coefficient 0.632 provient du fait que les échantillons de bootstrap comportent en moyenne $0.632m$ points différents. Cet estimateur corrige, dans le cas général, le biais du leave-one-out bootstrap, mais se retrouve biaisé dans le cas d'un sur-apprentissage. L'estimateur .632+ tend à corriger ce biais en augmentant la pondération de $\hat{Err}^{(1)}$ lorsque le sur-apprentissage est important. Cette quantité de sur-apprentissage est calculée à partir d'un coefficient d'*erreur non informative* qui correspond au cas où X et Y sont indépendants. D'après [41], un estimateur de cette quantité est

$$\hat{\tau} = \hat{p}_1\left(1-\hat{q}_1\right) + \left(1-\hat{p}_1\right)\hat{q}_1$$

avec $\hat{p}_1$ la proportion des y_i égaux à 1 et $\hat{q}_1$ celle des prédictions $f_d\left(x_i\right)$ égales à 1. Le taux de sur-apprentissage relatif est

$$\hat{R} = \frac{\hat{Err}^{(1)} - R_m\left(f_D\right)}{\hat{\tau} - R_m\left(f_D\right)}.$$

Avec cette définition, $\hat{R}$ peut ne pas appartenir à $[0,1]$, ce qui amène une définition modifiée de $\hat{Err}^{(1)}$ et $\hat{R}$:

$$\hat{Err}^{(1)'} = min\left(\hat{Err}^{(1)}, \hat{\tau}\right)$$

et

$$\hat{R}' = \begin{cases} \hat{\tau} \text{ si } \hat{Err}^{(1)} > R_m\left(f_D\right) \text{ ou } \hat{\tau} > R_m\left(f_D\right) \\ 0 \text{ sinon} \end{cases}.$$

L'estimateur de *bootstrap* .632+ est défini par

$$\hat{Err}^{(.632+)} = \hat{Err}^{(.632)} + \left(\hat{Err}^{(1)'} - R_m\left(f_D\right)\right)\frac{.368 \cdot .632\hat{R}'}{1 - .368\hat{R}'}. \tag{1.20}$$

Par définition, le calcul de $\hat{Err}^{(.632+)}$ demande à peine plus de calcul que celui de $\hat{Err}^{(1)}$.

1.4.2 Méthodes dédiées aux SVM bi-classes

Cette section présente les principales méthodes existantes de sélection de modèle pour les SVM bi-classes. Ces méthodes utilisent l'architecture des SVM ainsi que le fait que le classifieur est l'optimum du problème d'apprentissage.

1.4.2.1 Bornes sur l'erreur de validation croisée *leave-one-out*

Comme nous l'avons vu, la procédure de validation croisée leave-one-out est très coûteuse en opérations. Nous présentons ici des bornes supérieures sur cette erreur utilisant la structure des machines à vecteurs support. La liste n'est pas exhaustive et n'est introduite qu'afin de présenter les bases des développements à venir.

Le comptage des vecteurs support : La méthode la plus simple pour borner l'erreur de validation croisée leave-one-out pour une machine à marge dure est de partir de la constatation que seuls les vecteurs support jouent un rôle. En effet, enlever un exemple dont le multiplicateur de Lagrange est égal à zéro ne change pas la fonction de décision. La plus simple des bornes non triviales est donc

$$\hat{Err}^{(cv,1)} \leqslant \frac{N_{SV}}{m}$$

avec N_{SV} le nombre de vecteurs support de la machine entraînée avec tous les exemples.

La borne Rayon-Marge : Dans le chapitre 10 de [104], Vapnik propose une borne sur l'erreur de validation croisée leave-one-out de la SVM à marge dure.

Théorème 2 (D'après les sections 10.3 et 10.4 de [104])**.** *Considérons une SVM à marge dure entraînée sur d_m. Soit $\gamma = \left\|\bar{h}\right\|_\kappa^{-1}$ sa marge et R le rayon de la plus petite boule de $\mathbf{H}_\kappa$ contenant l'ensemble $\{\kappa_{x_i} : 1 \leqslant i \leqslant m\}$. Alors,*

$$\hat{Err}^{(cv,1)} = \frac{1}{m}\sum_{p=1}^{m} \mathbb{1}_{\{y_p h^p(x_p) \leqslant 0\}} \leqslant \frac{4}{m}\frac{R^2}{\gamma^2}$$

avec h^p la fonction calculée par la SVM entraînée sur $d_m \setminus \{(x_p, y_p)\}$.

Afin d'appliquer cette borne à la ℓ_2-SVM, il faut considérer le bon espace de représentation, c'est-à-dire le "RKHS induit par κ_λ" [1], qui dépend de λ. En conséquence, le rayon doit être recalculé pour chaque valeur de λ, ce qui revient à résoudre une série de problèmes de programmation quadratique.

La borne Span (*Span bound* en anglais) : La *Span bound* est une borne de l'erreur de validation croisée leave-one-out développée par Vapnik et Chapelle dans [105] en utilisant le concept de sous-espace vectoriel engendré par les vecteurs support. Seuls les résultats concernant

1. Il s'agit encore d'un abus de langage puisque κ_λ n'est pas un noyau.

la SVM à marge dure, et donc la ℓ_2-SVM, sont présentés ici bien qu'une formulation existe pour la ℓ_1-SVM.

Définition 11. *Soit Λ_p le sous-espace engendré par la combinaison linéaire contrainte des images des vecteurs support dans l'espace de représentation, à l'exception du vecteur support x_p :*

$$\Lambda_p = \left\{ \sum_{i \neq p,\ \alpha_i > 0} \tau_i \kappa_{x_i}, \sum_{i \neq p} \tau_i = 1 \right\}$$

et soit S_p la distance de cet espace à κ_{x_p} .

Nous utilisons la formulation et la notation de Chapelle dans [27]. La valeur maximale des S_p est appelée le S-span S. Nous avons alors :

Théorème 3 (Borne *Span* [27] et Théorème 10 de [105])**.** *Considérons une SVM à marge dure entraînée sur d_m. Soit $\gamma = \left\|\bar{h}\right\|_\kappa^{-1}$ sa marge, S le S-span et R le rayon de la plus petite boule de $\mathbf{H}_\kappa$ contenant l'ensemble $\{\kappa_{x_i} : 1 \leqslant i \leqslant m\}$. Alors,*

$$\hat{Err}^{(cv,1)} = \frac{1}{m} \sum_{p=1}^{m} \mathbb{1}_{\{y_p h^p(x_p) \leqslant 0\}} \leqslant \frac{2}{m} \frac{SR}{\gamma^2}$$

avec h^p la fonction calculée par la SVM entraînée sur $d_m \setminus \{(x_p, y_p)\}$.

La démonstration de cette borne est très proche de la démonstration de la borne Rayon-Marge, ce qui explique leurs formes similaires. La borne Span est plus fine dans la mesure où le S-span est toujours plus petit que le diamètre.

En raison de sa complexité de calcul, cette borne (exacte) sur l'erreur de validation croisée leave-one-out n'est pas utilisée en pratique. On lui préfère généralement une approximation de cette erreur basée sur la notion d'espace engendré défini pour la borne Span.

1.4.2.2 Approximation de l'erreur de validation croisée leave-one-out par les spans

Nous détaillons à présent une approximation de l'erreur de validation croisée leave-one-out. Cette approximation [105] se base sur l'hypothèse que l'ensemble des vecteurs support ne change pas pendant la procédure de validation croisée. Les notations utilisent explicitement λ afin de rappeler que ce résultat est aussi bien valable pour la machine à marge douce que pour la machine marge dure (qui correspond à $\lambda \to 0$).

Théorème 4 (Théorème 3 de [105])**.** *Soit une ℓ_1-SVM entraînée sur d_m. Sous l'hypothèse que l'ensemble des vecteurs support reste constant pendant la procédure de validation croisée, l'égalité suivante est vérifiée :*

$$\forall p \in [\![1, m]\!],\ \ y_p \left(h_\lambda(x_p) - h_\lambda^p(x_p) \right) = \alpha_p(\lambda) \, S_p(\lambda)^2$$

avec $S_p(\lambda)$ la distance entre κ_{x_p} et l'ensemble $\Lambda_p(\lambda)$ défini par

$$\Lambda_p(\lambda) = \left\{ \sum_{j:\alpha_j(\lambda)\in(0,\lambda^{-1})\wedge j\neq p} \tau_j \kappa_{x_j}, \sum_{j:\alpha_j(\lambda)\in(0,\lambda^{-1})\wedge j\neq p} \tau_j = 1 \right\} .$$

Cette définition de $\Lambda_p(\lambda)$ est légèrement différente de celle présentée dans [105], mais elles sont identiques sous l'hypothèse d'invariance des vecteurs support.

Corollaire 1 (Corollaire 1 dans [105]). *Sous l'hypothèse du théorème 4,*

$$\hat{Err}^{(cv,1)} = \frac{1}{m}\sum_{p=1}^{m} \mathbb{1}_{\left\{y_p h_\lambda^p(x_p)\leqslant 0\right\}} = \frac{1}{m}\sum_{p=1}^{m} \mathbb{1}_{\left\{\alpha_p(\lambda)S_p(\lambda)^2 - y_p h_\lambda(x_p)\geqslant 0\right\}}. \tag{1.21}$$

Le terme de droite de l'équation (1.21) est nommé l'approximation de l'erreur de validation croisée leave-one-out (ou simplement approximation de l'erreur de test par abus de langage) et nous le notons $\hat{Err}_{sv}^{(cv,1)}$, l'indice sv faisant référence à l'hypothèse d'invariance des vecteurs support.

Lorsque ce calcul est implanté de manière naïve, sa complexité est du même ordre de grandeur que celle de la procédure de validation croisée. Puisque nous nous intéressons de manière privilégiée aux machines à coût quadratique, la suite de l'étude se concentre sur les machines à marge dure. Ce cas est traité dans [27]. Dans ce cadre de travail, l'équation 1.21 se simplifie en

$$\frac{1}{m}\sum_{p=1}^{m} \mathbb{1}_{\{y_p h^p(x_p)\leqslant 0\}} = \frac{1}{m}\sum_{p=1}^{m} \mathbb{1}_{\left\{\alpha_p S_p^2 - 1\geqslant 0\right\}} .$$

Les auteurs fournissent un résultat pratique pour calculer la valeur des spans S_p en se basant sur une reformulation algébrique :

$$\forall p \in [\![1,m]\!],\ \ S_p^2 = \min_{\tau}\max_{\mu} \left(\Phi(x_p) - \sum_{j:\alpha_j>0\wedge j\neq p} \tau_j \kappa_{x_j} \right)^2 + 2\mu \left(\sum_{j:\alpha_j>0\wedge j\neq p} \tau_j - 1 \right)$$

avec μ le multiplicateur de Lagrange associé à la contrainte $\sum \tau_j = 1$. Soit $\mathcal{E}$ l'ensemble des indices des vecteurs support, soit $m_\mathcal{E}$ son cardinal et soit $K_{\mathcal{E},\mathcal{E}}$ la sous-matrice de K composée uniquement des lignes et colonnes d'indices dans $\mathcal{E}$. Soit $\bar{K}$ la matrice donnée par :

$$\bar{K} = \begin{pmatrix} K_{\mathcal{E},\mathcal{E}} & 1_{m_\mathcal{E}} \\ 1_{m_\mathcal{E}}^T & 0 \end{pmatrix}.$$

En posant $\bar{\tau} = \left(\tau^T, \mu\right)^T$, Le précédant problème min-max se reformule en :

$$S_p^2 = \min_{\tau}\max_{\mu} \left\{ \kappa(x_p,x_p) - 2v^T\bar{\tau} + \bar{\tau}^T V \bar{\tau} \right\}$$

avec V la sous-matrice de $\bar{K}$ obtenue en enlevant la ligne et la colonne d'indice p et v la p-ème colonne de $\bar{K}$ privée de sa p-ème composante. L'existence de V^{-1} dans le cas général n'est pas discutée : seul le cas où la matrice de Gram est de rang plein nous intéresse[1]. Puisque la valeur optimale de $\bar{\tau}$ est $V^{-1}v$, l'équation 12 de [27] donne la valeur de S_p :

$$\begin{aligned} S_p^2 &= \kappa(x_p, x_p) - v^T V^{-1} v \\ &= 1/\left(\bar{K}^{-1}\right)_{p,p} \quad . \end{aligned}$$

Ainsi, la plus coûteuse étape du calcul de l'approximation de l'erreur de validation croisée leave-one-out est l'inversion de $\bar{K}$. Ce résultat est seulement valable pour les machines à marge dure : l'extension aux machines à marge douce nécessite une reformulation à marge dure (uniquement possible pour la ℓ_2-SVM), ce qui implique la construction de K à partir de κ_λ.

1.4.3 Méthodes dédiées aux SVM multi-classes

Cette section présente les principales méthodes existantes de sélection de modèle pour les SVM multi-classes en mettant l'accent sur la LLW-M-SVM. Nous commençons par une méthode se basant sur une méthode de décomposition. Nous présentons ensuite la borne Rayon-Marge pour la machine Lee et coauteurs. Nous finissons par une estimation de l'erreur de test pour cette machine.

1.4.3.1 Extension multi-classe de Wang et coauteurs

Dans [109], Wang et ses coauteurs proposent deux extensions multi-classes de la borne Rayon-Marge dédiées à la méthode de décomposition un contre un. Dans la première extension, qui est une vraie borne, le terme de droite de l'équation 1.22, est justifiée par la proposition suivante, conséquence directe du théorème 2.

Proposition 3. *Considérons la méthode de décomposition un contre un utilisant des SVM à marge dure comme classifieurs de base. Soit $h_{kl} = \bar{h}_{kl} + b_{kl}$ la fonction calculée par le classifieur binaire entraîné à distinguer les catégories d'indices k et l, $\mathcal{D}_m(k,l)$ le diamètre de la plus petite boule de l'espace de représentation contenant ses vecteurs support et $\hat{N}_{(k,l)}^{(cv,1)}$ le nombre de ses erreurs résultant de la mise en œuvre sur l'ensemble d'apprentissage d'une procédure de validation croisée* leave-one-out. *Alors,*

$$\sum_{1 \leqslant k < l \leqslant Q} \hat{N}_{(k,l)}^{(cv,1)} \leqslant \sum_{1 \leqslant k < l \leqslant Q} \mathcal{D}_m(k,l)^2 \left\|\bar{h}_{kl}\right\|_\kappa^2 . \tag{1.22}$$

La seconde extension, qui n'est plus une borne, s'appuie sur des considérations différentes, fondées sur la notion de matrices de dispersion [49, 40]. Elle est égale à $\frac{\mathcal{D}_m^2}{\bar{\gamma}^2}$, où l'expression du

1. La matrice de Gram d'un noyau $\kappa_\lambda = \kappa + \delta$ est de rang plein. La diagonalisation de la matrice associée à κ permet de l'établir.

carré de la marge $\bar{\gamma}$ est la suivante :

$$\bar{\gamma}^2 = \frac{1}{m^2} \sum_{k<l} \frac{m_k m_l}{\left\|\bar{h}_{kl}\right\|_\kappa^2}. \tag{1.23}$$

1.4.3.2 Borne Rayon-Marge pour la LLW-M-SVM

La démonstration de la borne Rayon-Marge multi-classe, présentée en détail dans [63], se base sur le même principe que celle du lemme sur lequel s'appuie la borne bi-classe. Puisque la borne Span est, dans sa démonstration, proche de la borne Rayon-Marge, nous utilisons la borne Rayon-Marge multi-classe comme inspiration pour la borne Span multi-classe (section 3.3.1).

Lemme 1. *Soit une LLW-M-SVM à marge dure entraînée sur d_m. Considérons à présent la même machine entraînée sur $d_m \setminus \{(x_p, y_p)\}$. Si elle fait une erreur sur (x_p, y_p), alors*

$$\max_{1 \leqslant l \leqslant Q} \alpha_{pl}^* \geqslant \frac{Q}{(Q-1)^3 \mathcal{D}_m^2},$$

avec $\mathcal{D}_m$ le diamètre de la plus petite boule de $\mathbf{H}_\kappa$ contenant l'ensemble $\{\kappa_{x_i} : 1 \leqslant i \leqslant m\}$.

Nous conseillons au lecteur curieux d'aller lire la démonstration du lemme [63].

La formulation de la borne Rayon-Marge multi-classe de [63] fait intervenir la notion de marge géométrique entre catégories. Ce concept ne nous est pas utile dans le manuscrit, nous ne présenterons donc la borne Rayon-Marge que par son expression la plus compacte.

Théorème 5. *Soit une LLW-M-SVM à marge dure à Q catégories entraînée sur d_m. Soit $\mathcal{D}_m$ le diamètre de la plus petite boule de $\mathbf{H}_\kappa$ contenant l'ensemble $\{\kappa_{x_i} : 1 \leqslant i \leqslant m\}$. Alors*

$$\hat{Err}^{(cv,1)} \leqslant \frac{(Q-1)^4}{Q} \|\bar{h}^*\|_{\kappa,Q}^2 \mathcal{D}_m^2. \tag{1.24}$$

1.5 Conclusion

Dans ce chapitre, le cadre théorique de la discrimination a été introduit. Nous avons particulièrement mis l'accent sur l'importance de la complexité de la classe de fonctions sur laquelle se base le classifieur. Nous avons ensuite présenté différentes machines à vecteurs support aussi bien bi-classes que multi-classes. L'attention du lecteur a été attirée sur l'importance de la norme utilisée pour le terme d'adéquation aux données. Ce chapitre s'est conclu sur un bref état de l'art des méthodes de sélection de modèle pour les SVM. Cette sélection de modèle consiste à choisir une classe de fonctions adaptée au problème. La sélection de l'hyperparamètre λ (le coefficient de régularisation) est l'un de ceux permettant ce choix. Le chapitre 2 lui est dédié dans le cas bi-classe alors que le chapitre 3 se concentre sur une machine multi-classe à coût quadratique.

Chapitre 2

Sélection de modèle par parcours du chemin de régularisation pour la SVM bi-classe

« Pour réussir, il ne suffit pas de prévoir. Il faut aussi savoir improviser. »
—Fondation, Isaac Asimov

Les méthodes de parcours de chemin s'intègrent dans le cadre théorique des méthodes de continuation [2]. L'origine de ces méthodes est attribuée, en partie, aux travaux de Poincaré, que Leray et Schauder ont raffinés. Afin de résoudre le problème d'intérêt (systèmes d'équations, équations aux dérivées partielles, etc.), ces méthodes, présentées en détail dans [3], cherchent la solution d'un problème plus simple. La solution de ce problème trouvée, il est progressivement (continuement) transformé afin de revenir au problème d'intérêt et ainsi obtenir la solution voulue. L'apprentissage d'une machine à vecteurs support peut être vu comme une forme de méthode de continuation : lorsque le coefficient de régularisation tend vers l'infini, la solution est triviale [1] puis, au fur et à mesure de la diminution du coefficient, le classifieur gagne en complexité (c'est-à-dire en adéquation aux données). L'utilisation habituelle consiste à parcourir l'ensemble du chemin. En effet, le problème d'origine correspond à l'extrémité non triviale de ce chemin. Dans le cadre d'un problème d'apprentissage, cette extrémité équivaut à un apprentissage non régularisé. Dans une optique de sélection de modèle, nous cherchons alors une quantité de *déformation optimale* entre le problème complètement régularisé et celui non régularisé. Définissons à présent le chemin de régularisation dans le cadre d'un algorithme d'apprentissage.

Définition 12 (Chemin de régularisation)**.** *Le chemin de régularisation d'un algorithme d'apprentissage est l'ensemble des solutions optimales pour toutes les valeurs du coefficient de régularisation.*

1. Elle ne dépend plus des descriptions.

En pratique, nous ne parcourons le chemin de régularisation que pour un intervalle du coefficient de régularisation. Par abus de langage, cette portion de chemin sera appelée chemin de régularisation. Dans ce chapitre, nous présentons en premier lieu l'algorithme proposé par Hastie et ses coauteurs [66] pour la ℓ_1-SVM. La seconde section traite du parcours du chemin de régularisation de la ℓ_2-SVM tandis que la troisième est dédiée à la sélection de modèle pour cette machine. Le chapitre se conclut sur les résultats expérimentaux de la sélection de modèle par parcours du chemin de régularisation pour la ℓ_2-SVM.

2.1 Parcours du chemin de régularisation pour la ℓ_1-SVM

Cette section détaille la méthode proposée par Hastie et ses coauteurs [66] pour parcourir l'ensemble des solutions de l'apprentissage d'une ℓ_1-SVM pour une plage de valeurs de λ. La première section détaille la manière d'obtenir l'équation de mise à jour des multiplicateurs de Lagrange. La suivante reprend les calculs de recherche de point de départ de Hastie et coauteurs en détaillant et précisant les lemmes correspondants. La dernière section présente la manière d'identifier les modifications des ensembles d'indices d'exemples sur lesquels se base la méthode de parcours du chemin de régularisation.

2.1.1 Calcul des multiplicateurs de Lagrange de la ℓ_1-SVM en suivant le chemin de régularisation

L'idée d'un parcours efficace du chemin de régularisation consiste à chercher à identifier des intervalles pour le coefficient λ dans lesquels le vecteur de multiplicateurs de Lagrange $\beta(\lambda)$ du problème 2 s'obtient facilement. En combinant l'équation (1.3) et les conditions complémentaires de Kuhn-Tucker du problème 1, on obtient le système de propositions suivant :

$$\begin{cases} \text{si } y_i h_\lambda(x_i) > 1, \text{ alors } \xi_i(\lambda) = 0 \text{ et } \beta_i(\lambda) = 0 \\ \text{si } y_i h_\lambda(x_i) = 1, \text{ alors } \xi_i(\lambda) = 0 \text{ et } \beta_i(\lambda) \in [0,1] \\ \text{si } y_i h_\lambda(x_i) < 1, \text{ alors } \xi_i(\lambda) > 0 \text{ et } \beta_i(\lambda) = 1 \end{cases} .$$

Rappelons, sur un exemple, le raisonnement permettant d'obtenir ces implications :

$$y_i h_\lambda(x_i) > 1 \Rightarrow y_i h_\lambda(x_i) - 1 + \xi_i(\lambda) > 0$$

$$\Rightarrow \beta_i(\lambda) = 0 \text{ (condition de Kuhn-Tucker complémentaire)}$$

$$\Rightarrow \theta_i(\lambda) = 1$$

$$\Rightarrow \xi_i(\lambda) = 0.$$

Pour déterminer les intervalles de λ précités, nous nous appuyons sur la partition de l'ensemble d'apprentissage induite par ces propositions. Ceci nous conduit à utiliser les ensembles définis

dans [66] :

$$\begin{aligned}\mathcal{E}(\lambda) &= \{i \ : y_i h_\lambda(x_i) = 1\} \\ \mathcal{L}(\lambda) &= \{i \ : y_i h_\lambda(x_i) < 1\} \\ \mathcal{R}(\lambda) &= \{i \ : y_i h_\lambda(x_i) > 1\}.\end{aligned}$$

Dans un soucis de simplification des notations, nous utilisons $\mathcal{E}$ au lieu de $\mathcal{E}(\lambda)$ lorsqu'aucune ambiguïté n'est à craindre. Soient $m_\mathcal{E}$, $m_\mathcal{L}$ et $m_\mathcal{R}$ les cardinaux respectifs de $\mathcal{E}$, $\mathcal{L}$ et $\mathcal{R}$ ($m = m_\mathcal{E} + m_\mathcal{L} + m_\mathcal{R}$). Les indices des exemples de l'ensemble d'apprentissage sont modifiés de manière que les plus petits indices correspondent aux exemples de $\mathcal{E}$ et les plus grands à ceux de $\mathcal{R}$. Ainsi, y peut se réordonner en $y = \left(y_\mathcal{E}^T y_\mathcal{L}^T y_\mathcal{R}^T\right)^T$ et il en va de même pour les vecteurs $\alpha(\lambda)$ et $\beta(\lambda)$. De manière analogue, la matrice hessienne se réordonne en

$$H = \begin{pmatrix} H_{\mathcal{E},\mathcal{E}} & H_{\mathcal{E},\mathcal{L}} & H_{\mathcal{E},\mathcal{R}} \\ H_{\mathcal{L},\mathcal{E}} & H_{\mathcal{L},\mathcal{L}} & H_{\mathcal{L},\mathcal{R}} \\ H_{\mathcal{R},\mathcal{E}} & H_{\mathcal{R},\mathcal{L}} & H_{\mathcal{R},\mathcal{R}} \end{pmatrix}$$

où la sous-matrice $H_{\mathcal{I},\mathcal{J}}$ pour $(\mathcal{I},\mathcal{J}) \in \{\mathcal{E},\mathcal{L},\mathcal{R}\}^2$ est la matrice $(h_{ij})_{i\in\mathcal{I},j\in\mathcal{J}}$.

Les β_i varient continuement en fonction de λ (continuité de la fonction objectif par rapport à λ), en conséquence, un point ne peut pas passer de $\mathcal{L}$ à $\mathcal{R}$ sans passer par $\mathcal{E}$. Grâce à cette propriété, il est possible de montrer que le vecteur des multiplicateurs de Lagrange pour les exemples d'indice dans $\mathcal{E}$ s'obtient par résolution d'un système linéaire :

Proposition 4. *Pour toute valeur strictement positive de λ, le vecteur $\beta_\mathcal{E}^a(\lambda)$ vérifie la relation linéaire suivante :*

$$A_\mathcal{E} \beta_\mathcal{E}^a(\lambda) = C_{\mathcal{E},\mathcal{L}}(\lambda)$$

avec :

- $\beta_\mathcal{E}^a(\lambda) = \left(\beta_0(\lambda) \ \ \beta_\mathcal{E}(\lambda)^T\right)^T$,
- $A_\mathcal{E} = \begin{pmatrix} 0 & y_\mathcal{E}^T \\ y_\mathcal{E} & H_{\mathcal{E},\mathcal{E}} \end{pmatrix} \in \mathcal{M}_{m_\mathcal{E}+1, m_\mathcal{E}+1}(\mathbb{R})$,
- $C_{\mathcal{E},\mathcal{L}}(\lambda) = \begin{pmatrix} -y_\mathcal{L}^T 1_{m_\mathcal{L}} \\ \lambda 1_{m_\mathcal{E}} - H_{\mathcal{E},\mathcal{L}} 1_{m_\mathcal{L}} \end{pmatrix} \in \mathbb{R}^{m_\mathcal{E}+1}$.

Démonstration. L'équation (1.4) peut se reformuler matriciellement :

$$\begin{aligned}\forall i \in [\![1,m]\!], \ \ y_i h_\lambda(x_i) &= \frac{1}{\lambda}\left(\sum_{j=1}^m \beta_j(\lambda) y_j y_i \kappa(x_j,x_i) + y_i \beta_0(\lambda)\right) \\ &= \frac{1}{\lambda}\left(\left(H\beta(\lambda)\right)_i + y_i\beta_0(\lambda)\right).\end{aligned}$$

Pour $i \in \mathcal{E}$, nous avons donc :

$$\lambda = \left(\left(H\beta\left(\lambda\right) \right)_i + y_i \beta_0\left(\lambda\right) \right). \tag{2.1}$$

La combinaison de (1.2) et de (2.1) donne :

$$\begin{cases} y_{\mathcal{E}}^T \beta_{\mathcal{E}}\left(\lambda\right) + y_{\mathcal{L}}^T \beta_{\mathcal{L}}\left(\lambda\right) + y_{\mathcal{R}}^T \beta_{\mathcal{R}}\left(\lambda\right) = 0 \\ H_{\mathcal{E},\mathcal{E}} \beta_{\mathcal{E}}\left(\lambda\right) + H_{\mathcal{E},\mathcal{L}} \beta_{\mathcal{L}}\left(\lambda\right) + H_{\mathcal{E},\mathcal{R}} \beta_{\mathcal{R}}\left(\lambda\right) + \beta_0\left(\lambda\right) y_{\mathcal{E}} = \lambda 1_{m_{\mathcal{E}}} \end{cases}$$

soit encore, compte tenu des définitions des ensembles $\mathcal{E}$, $\mathcal{L}$ et $\mathcal{R}$:

$$\begin{cases} y_{\mathcal{E}}^T \beta_{\mathcal{E}}\left(\lambda\right) + y_{\mathcal{L}}^T 1_{m_{\mathcal{L}}} = 0 \\ H_{\mathcal{E},\mathcal{E}} \beta_{\mathcal{E}}\left(\lambda\right) + H_{\mathcal{E},\mathcal{L}} 1_{m_{\mathcal{L}}} + \beta_0\left(\lambda\right) y_{\mathcal{E}} = \lambda 1_{m_{\mathcal{E}}} \end{cases}.$$

La réécriture matricielle de ce système fournit le résultat annoncé. □

Soient $\lambda > 0$ et $\delta < 0$ tels que la partition de l'ensemble d'apprentissage demeure inchangée entre λ et $\lambda + \delta$. Soient $\beta_{\mathcal{E}}^a\left(\lambda\right)$ et $\beta_{\mathcal{E}}^a\left(\lambda + \delta\right)$ des racines du système linéaire de la proposition 4 respectivement associées à λ et $\lambda + \delta$. On a

$$A_{\mathcal{E}} \left[\beta_{\mathcal{E}}^a\left(\lambda + \delta\right) - \beta_{\mathcal{E}}^a\left(\lambda\right) \right] - \left(C_{\mathcal{E},\mathcal{L}}\left(\lambda + \delta\right) - C_{\mathcal{E},\mathcal{L}}\left(\lambda\right) \right) = 0_{m_{\mathcal{E}}+1}$$

avec

$$C_{\mathcal{E},\mathcal{L}}\left(\lambda + \delta\right) - C_{\mathcal{E},\mathcal{L}}\left(\lambda\right) = \begin{pmatrix} 0 \\ \delta 1_{m_{\mathcal{E}}} \end{pmatrix},$$

qui se simplifie en :

$$A_{\mathcal{E}} \left[\beta_{\mathcal{E}}^a\left(\lambda + \delta\right) - \beta_{\mathcal{E}}^a\left(\lambda\right) \right] = \begin{pmatrix} 0 \\ \delta 1_{m_{\mathcal{E}}} \end{pmatrix}.$$

Si $A_{\mathcal{E}}$ n'est pas de rang plein, alors l'unicité du chemin n'est pas assurée. Ce cas est discuté dans la section 4 de [66]. Dans la suite, nous faisons l'hypothèse que cette matrice est régulière. L'équation précédente nous fournit alors une formule unique de mise à jour de $\beta_{\mathcal{E}}^a$ tant que $\mathcal{E}$ demeure inchangé :

$$\beta_{\mathcal{E}}^a\left(\lambda + \delta\right) = \beta_{\mathcal{E}}^a\left(\lambda\right) + A_{\mathcal{E}}^{-1} \begin{pmatrix} 0 \\ \delta 1_{m_{\mathcal{E}}} \end{pmatrix}. \tag{2.2}$$

Cette formule donnant $\beta_{\mathcal{E}}^a\left(\lambda + \delta\right)$ en fonction de $\beta_{\mathcal{E}}^a\left(\lambda\right)$ et de δ correspond à l'équation (37) de [66]. La démonstration présentée ici n'est pas celle d'Hastie mais est proche de celle de [108]. Afin d'appliquer l'équation (2.2), il est nécessaire de disposer d'une valeur initiale pour le vecteur $\beta_{\mathcal{E}}^a$. Il est naturellement possible de résoudre le problème de programmation quadratique pour une valeur de λ donnée, puis de calculer les mises à jour à partir de la valeur de $\beta_{\mathcal{E}}^a\left(\lambda\right)$ ainsi obtenue. Cependant, nous allons voir que lorsque λ est suffisamment grand, il est possible de calculer la valeur des multiplicateurs de Lagrange pour une complexité de calcul plus faible qu'un

apprentissage normal.

2.1.2 Calcul du point de départ du chemin de régularisation

L'intérêt d'une initialisation spécifique pour une grande valeur de λ, par rapport à un apprentissage par résolution directe du problème dual, est double. D'une part, elle permet de trouver la solution du problème 1 pour un temps de calcul moindre, d'autre part elle permet de connaître la plus grande valeur de λ pour laquelle un événement intervient. Notons respectivement $I_+ = \{i \in [\![1, m]\!] / \ y_i = 1\}$ l'ensemble des indices des exemples de la catégorie "positive" et $I_- = \{i \in [\![1, m]\!] / \ y_i = -1\}$ celui des exemples des indices de la catégorie "négative". Notons respectivement $m_+ = |I_+|$ et $m_- = |I_-|$ leurs cardinalités. Le calcul de la valeur des multiplicateurs pour λ suffisamment grand dépend des valeurs relatives de m_+ et m_-. Le premier cas que nous traitons est l'égalité des cardinaux, alors que le second est celui où la classe "positive" est majoritaire. Ce choix est fait pour simplifier la démonstration et ne nuit en rien à la généralité du résultat. En effet, si la classe "négative" est majoritaire, il suffit alors d'inverser les étiquettes pour retrouver le cas traité.

2.1.2.1 Cas $m_+ = m_-$

Ce cas est traité par le lemme 1 de [66]. Nous allons démontrer un lemme plus précis.

Lemme 2 (d'après le lemme 1 de [66])**.** *Si $m_+ = m_-$, alors il existe un λ^* tel que pour tout $\lambda > \lambda^*$ nous avons : $\beta(\lambda) = 1_m$ et $b_\lambda \in [-1, 1]$. De plus, la limite lorsque λ tend vers l'infini de $\sum_{i=1}^m \xi_i(\lambda)$ est m.*

Démonstration. Les composantes du vecteur de multiplicateurs de Lagrange appartiennent à $[0,\ 1]$ donc, l'équation (1.1) implique que la norme de $\bar{h}_\lambda$ tend vers zéro lorsque λ tend vers l'infini. L'équation (1.4) nous permet donc d'affirmer que h_λ tend vers une fonction constante.

Pour λ grand, les contraintes de bon classement peuvent être réécrites sous la forme

$$\forall i \in [\![1, m]\!], \quad \xi_i(\lambda) \geqslant 1 - y_i b_\lambda + o(1).$$

Étudions quelles valeurs peut prendre b_λ.

- Cette réécriture permet de déduire qu'il existe λ' tel que si $\lambda > \lambda'$ et si b_λ est dans $]-1, 1[$, alors $\xi(\lambda)$ est dans $\left(\mathbb{R}_+^*\right)^m$. La stricte positivité de toutes les variables d'écart impose, par les conditions complémentaires de Kuhn-Tucker, que $\theta(\lambda)$ est égal au vecteur nul, ce qui induit que le vecteur $\beta(\lambda)$ est égal à 1_m (d'après l'équation (1.3)).
- Intéressons-nous au cas où b_λ est supérieur ou égal à 1. Il existe λ'' tel que si $\lambda > \lambda''$ et si $b_\lambda \geqslant 1$ alors pour tout $i \in I_+$, $\xi_i(\lambda) = 0$ et pour tout $i \in I_-$, $\xi_i(\lambda) = 1 + b_\lambda + o(1)$. Pour minimiser $\sum_{i=1}^m \xi_i(\lambda)$, il faut donc choisir $b_\lambda = 1$. De plus, comme tous les exemples de la catégorie "négative" sont mal classés, les conditions complémentaires de Kuhn-Tucker et l'équation (1.3) imposent $\forall i \in I_-$, $\beta_i(\lambda) = 1$. En injectant cette expression dans la

contrainte égalité du problème 2, nous obtenons $\sum_{i\in I_-}\beta_i(\lambda) = m_- = \sum_{i\in I_+}\beta_i(\lambda)$, ce qui impose à nouveau que $\beta(\lambda) = 1_m$.

- Pour le cas où b_λ est inférieur ou égal à -1, alors, par symétrie, il existe λ''' tel que si $\lambda > \lambda'''$ et si $b_\lambda \leqslant -1$ alors tous les exemples de la catégorie "positive" sont mal classés. On trouve donc $\beta(\lambda) = 1_m$ et $b_\lambda = -1$.

En posant $\lambda^* = \max\{\lambda', \lambda'', \lambda'''\}$, on a donc établi que pour $\lambda > \lambda^*$, on a à la fois $\beta(\lambda) = 1_m$ et $b_\lambda \in [-1, 1]$.

Établissons maintenant que la limite de la somme des variables d'écart est égale à la cardinalité de l'ensemble d'apprentissage. Pour $\lambda > \lambda^*$, d'après (1.1), nous avons

$$\frac{\lambda}{2}\|\bar{h}_\lambda\|_\kappa^2 = \frac{1}{2\lambda}1_m^T H 1_m \xrightarrow[\lambda\to\infty]{} 0.$$

Nous pouvons alors définir un problème équivalent au problème 1 lorsque $\lambda \to \infty$:

$$\min_{b',\xi'} \sum_{i=1}^m \xi'_i$$

$$\text{s.c.} \begin{cases} \forall i \in [\![1, m]\!],\ y_i b' \geqslant 1 - \xi'_i \\ \forall i \in [\![1, m]\!],\ \xi'_i \geqslant 0 \end{cases}.$$

La solution optimale est :

$$\forall i \in [\![1, m]\!],\ \xi_i'^* = \left(1 - y_i b'^*\right)_+.$$

Puisque $b' \in [\![-1, 1]\!]$, ceci se simplifie en

$$\forall i \in [\![1, m]\!],\ \xi_i'^* = 1 - y_i b'^*$$

et donc $\sum_{i=1}^m \xi_i'^* = m$ puisque $m_+ = m_-$. Nous avons donc $\lim_{\lambda\to\infty}\sum_{i=1}^m \xi_i(\lambda) = \sum_{i=1}^m \xi_i'^* = m$. Nous concluons ainsi la démonstration en montrant que la limite de la somme des variables d'écart est connue : $\sum_{i=1}^m \xi_i(\lambda) \xrightarrow[\lambda\to+\infty]{} m$. □

Étudions à présent la partition de l'ensemble d'apprentissage pour $\lambda > \lambda^*$.

Lemme 3 (Existence de points dans $\mathcal{E}$ pour $\lambda > \lambda^*$)**.** *Supposons $m_+ = m_-$. Définissons les ensembles :*

- $\mathcal{J} = \left\{j \in [\![1,\ m]\!] \ :\ y_j h_\lambda(x_j) = \max_{k\in[\![1,\ m]\!]} y_k h_\lambda(x_k)\right\}$,
- $\mathcal{K}_+ = \left\{j \in I_+ \ :\ y_j h_\lambda(x_j) = \max_{k\in I_+} y_k h_\lambda(x_k)\right\}$,
- $\mathcal{K}_- = \left\{j \in I_- \ :\ y_j h_\lambda(x_j) = \max_{k\in I_-} y_k h_\lambda(x_k)\right\}$

et les constantes $c_+ = \max_{k\in I_+}\sum_{i=1}^m y_i\kappa(x_i, x_k)$ et $c_- = \max_{k\in I_-} -\sum_{i=1}^m y_i\kappa(x_i, x_k)$. Pour $\lambda > \lambda^*$ le classifieur n'est pas défini de manière unique par l'algorithme d'apprentissage : il existe plusieurs valeurs de b_λ fournissant des solutions optimales. Les valeurs optimales possibles de b_λ dépendent du choix d'avoir un ensemble $\mathcal{E}$ vide ou non :

- *Si on choisit $\mathcal{E}$ non vide, alors $\mathcal{E} = \mathcal{J}$ et les indices de $\mathcal{E}$ appartiennent tous à la même catégorie, notée $y_{\mathcal{J}}$. L'ensemble $\mathcal{L}$ contient tous les autres indices. Pour tout $j \in \mathcal{J}$, la somme $\sum_{i=1}^{m} y_i \kappa(x_i, x_j)$ est une constante que nous notons $c_{\mathcal{E}}$. Nous avons $b_\lambda = y_{\mathcal{J}} - \frac{1}{\lambda} c_{\mathcal{E}}$ dont la limite à l'infini est $y_{\mathcal{J}}$.*
- *Si on choisit $\mathcal{E}$ vide, alors $\mathcal{L}$ est égal à $[\![1, m]\!]$. La valeur de b_λ peut être choisie arbitrairement dans l'intervalle $\left]-1 + \frac{1}{\lambda} c_-, 1 - \frac{1}{\lambda} c_+\right[$.*

Démonstration. Cette démonstration se déroule en deux étapes. La première consiste à envisager deux cas possibles pour la cardinalité de $\mathcal{E}$. Sous chacune de ces hypothèses, nous cherchons la solution optimale. La seconde étape est une synthèse de ces deux cas et montre que ces solutions fournissent la même valeur de fonction objectif. Il n'y a pas une solution unique.

Premier cas : $\mathcal{E}$ est non vide

Montrons par l'absurde que si l appartient à $\mathcal{E}$ alors l appartient nécessairement à $\mathcal{J}$. Supposons $l \in \mathcal{E}$ et $l \notin \mathcal{J}$. Par définition,

$$\forall j \in \mathcal{J}, \quad y_j h_\lambda(x_j) > y_l h_\lambda(x_l) = 1.$$

Cette inégalité implique que les éléments de $\mathcal{J}$ appartiennent à $\mathcal{R}$, et donc que leurs multiplicateurs de Lagrange sont nuls, ce qui est contradictoire avec le lemme 2 qui montre qu'ils sont égaux à 1. Nous avons ainsi montré que $\mathcal{E} \subset \mathcal{J}$. Une conséquence directe de la définition de $\mathcal{E}$ est que $\forall j \in \mathcal{J}, \quad y_j h_\lambda(x_j) = 1$ et donc $\mathcal{J} \subset \mathcal{E}$. Cette inclusion mutuelle implique naturellement que $\mathcal{J} = \mathcal{E}$.

Montrons à présent par l'absurde que tous les points dont les indices sont dans $\mathcal{E}$ appartiennent à la même catégorie. Supposons qu'il existe $j^+ \in \mathcal{E}$ avec $y_{j^+} = 1$ et $j^- \in \mathcal{E}$ avec $y_{j^-} = -1$. Nous avons alors

$$\forall \lambda > \lambda^* \qquad 1 = h_\lambda\left(x_{j^+}\right) = \frac{1}{\lambda} \sum_{i=1}^{m} y_i \kappa\left(x_i, x_{j^+}\right) + b_\lambda$$

et

$$\forall \lambda > \lambda^* \qquad -1 = h_\lambda\left(x_{j^+}\right) = \frac{1}{\lambda} \sum_{i=1}^{m} y_i \kappa\left(x_i, x_{j^-}\right) + b_\lambda.$$

On obtient ainsi deux valeurs asymptotiques distinctes pour b_λ, -1 et 1, ce qui établit la contradiction. Tous les indices de $\mathcal{E}$ appartiennent donc à la même catégorie. Il en découle que $c_{\mathcal{E}}$ est égal soit à c_+ soit à c_-, selon la catégorie des indices de $\mathcal{E}$. Nous avons prouvé que si $\mathcal{E}$ est non vide alors $\forall j \in \mathcal{J}, \; b_\lambda = y_j - \frac{1}{\lambda} c_{\mathcal{E}}$ dont la limite lorsque λ tend vers ∞ est $y_j = y_{\mathcal{J}}$.

Second cas : $\mathcal{E}$ est vide

Puisque $\mathcal{E}$ est vide et que les multiplicateurs de Lagrange sont égaux à 1, tous les exemples ont leur indice dans $\mathcal{L}$. Recherchons les valeurs possibles de b_λ satisfaisant $y_i h_\lambda(x_i) < 1$ pour tout i

dans $[\![1, m]\!]$. Pour tout $k_+ \in \mathcal{K}_+$, nous avons alors

$$\begin{aligned} & y_{k_+} h_\lambda \left(x_{k_+}\right) < 1 \\ \Leftrightarrow & \frac{1}{\lambda} \sum_{i=1}^{m} y_i \kappa \left(x_i, x_{k_+}\right) + b_\lambda < 1 \\ \Leftrightarrow & b_\lambda < 1 - \frac{1}{\lambda} c_+. \end{aligned}$$

Pour tout $k_- \in \mathcal{K}_-$ nous obtenons de manière analogue

$$b_\lambda > -1 + \frac{1}{\lambda} c_-.$$

Nous avons donc prouvé que si $\mathcal{E}$ est vide, alors b_λ appartient à $\left]-1 + \frac{1}{\lambda} c_-, 1 - \frac{1}{\lambda} c_+\right[$.

Synthèse des deux cas :
Rappelons quelques relations utiles à la démonstration

$$\|\bar{h}_\lambda\|_\kappa^2 = \frac{1}{\lambda^2} \sum_{i,j} y_i y_j \kappa \left(x_i, x_j\right),$$

$$\begin{aligned} \forall \lambda > \lambda^*,\ \forall i \in \mathcal{L},\ \xi_i &= 1 - y_i h_\lambda \left(x_i\right) \\ &= 1 - y_i \frac{1}{\lambda} \sum_j y_j \kappa \left(x_j, x_i\right) - y_i b_\lambda > 0, \end{aligned}$$

$$\forall i \in \mathcal{E},\ \xi_i = 0$$

et si $\mathcal{E} \neq \varnothing$ alors pour tout élément j de $\mathcal{E}$, $b_\lambda = y_j - \frac{1}{\lambda} \sum_{i=1}^{m} y_i \kappa \left(x_i, x_j\right)$.

Notons $J_{\mathcal{L}}$ la valeur de la fonction objectif du problème 1 en choisissant un ensemble $\mathcal{E}$ vide et notons $J_{\mathcal{E}}$ la valeur de la même fonction objectif en ayant choisi un ensemble $\mathcal{E}$ non vide.

Commençons par nous intéresser au cas où $\mathcal{E}$ est non vide. Le vecteur des multiplicateurs de Lagrange est donné par le lemme 2 et, par les conditions de Kuhn-Tucker, nous obtenons la valeur optimale de b_λ, qui est $y_{\mathcal{J}} - \frac{1}{\lambda} c_{\mathcal{E}}$. Comme $J_{1,p}(h, \xi) = \sum_{i=1}^{m} \xi_i + \frac{\lambda}{2} \|\bar{h}\|_\kappa^2$, nous pouvons

alors évaluer la fonction objectif $J_{\mathcal{E}}$,

$$\begin{aligned}
\forall \lambda > \lambda^*, \quad J_{\mathcal{E}} &= \sum_i \xi_i + \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) \\
&= \sum_{i \in \mathcal{L}} \left(1 - y_i \frac{1}{\lambda} \sum_j y_j \kappa(x_i, x_j) - y_i b_\lambda \right) + \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) \\
&= (m - m_{\mathcal{E}}) - \frac{1}{\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) + y_{\mathcal{J}} \frac{1}{\lambda} \underbrace{\sum_{i \in \mathcal{E}} \sum_j y_j \kappa(x_i, x_j)}_{= m_{\mathcal{E}} c_{\mathcal{E}}} - \left(\underbrace{\sum_{i \in I^+} y_i b_\lambda}_{= m_+ b_\lambda} - \underbrace{\sum_{i \in \mathcal{E}} y_i b_\lambda}_{= y_{\mathcal{J}} m_{\mathcal{E}} b_\lambda} + \underbrace{\sum_{i \in I^-} y_i b_\lambda}_{= -m_- b_\lambda} \right) \\
&\quad + \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) \\
&= m - y_{\mathcal{J}} m_{\mathcal{E}} \underbrace{\left(y_{\mathcal{J}} - \frac{1}{\lambda} c_{\mathcal{E}} \right)}_{= b_\lambda} + y_{\mathcal{J}} m_{\mathcal{E}} b_\lambda - \frac{1}{\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) + \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) \\
&= m - \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) .
\end{aligned}$$

Montrons à présent que, pour tout b_λ, la fonction objectif $J_{\mathcal{L}}$ est égale à $J_{\mathcal{E}}$

$$\begin{aligned}
\forall \lambda > \lambda^*, \quad J_{\mathcal{L}} &= \sum_i \xi_i + \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) \\
&= \sum_i \left(1 - y_i \frac{1}{\lambda} \sum_j y_j \kappa(x_i, x_j) - y_i b_\lambda \right) + \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) \\
&= m - \frac{1}{\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) - m_+ b_\lambda + m_- b_\lambda + \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) \\
&= m - \frac{1}{2\lambda} \sum_{i,j} y_i y_j \kappa(x_i, x_j) \\
&= J_{\mathcal{E}}.
\end{aligned}$$

Nous avons montré que les valeurs de la fonction objectif du problème primal prises aux solutions optimales correspondant à $\mathcal{E} = \emptyset$ et $\mathcal{E} \neq \emptyset$ sont identiques. Nous concluons donc cette démonstration en ayant montré que, lorsque $\lambda > \lambda^*$, on peut choisir indifféremment de prendre $\mathcal{E}$ vide ou non pour obtenir une solution optimale de l'apprentissage. □

Nous avons montré que pour $\lambda > \lambda^*$, il existe des solutions optimales pour $\mathcal{E}$ non vide et pour $\mathcal{E}$ vide. Comme dans [66], nous choisissons de poursuivre l'exposé en retenant la seconde option.

D'après le lemme 2, nous savons que les multiplicateurs de Lagrange restent constants au-

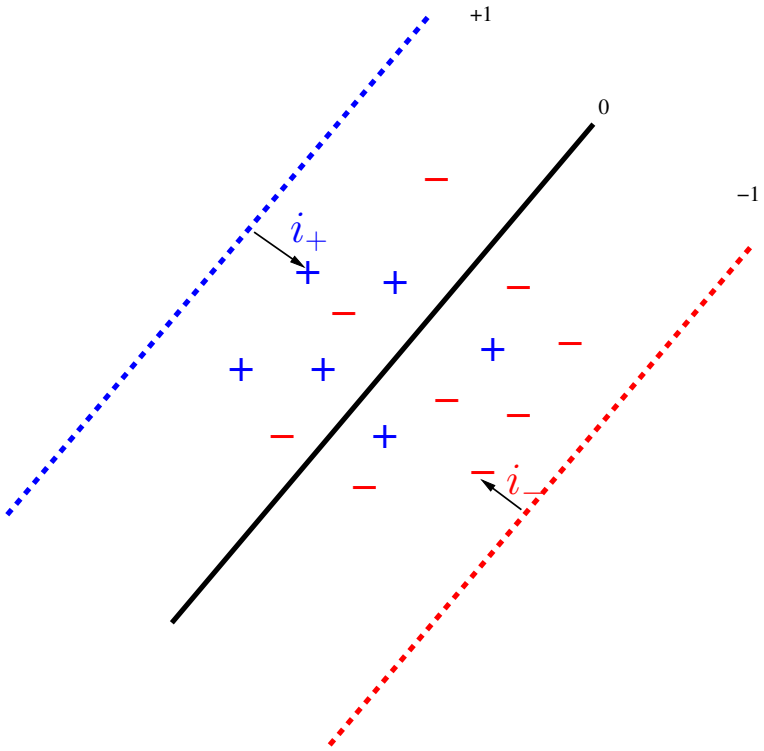


FIGURE 2.1 – Répartition des exemples par rapport à la marge pour $\lambda > \lambda_0$: tous les exemples sont à l'intérieur de la marge. Les indices i_- et i_+ sont ceux des exemples les plus proches de la marge.

delà d'une certaine valeur de λ qu'on appellera dans la suite λ_0 et qui est la plus petite valeur possible de λ^*. Cette valeur λ_0 correspond au moment où β cesse d'être constant, c'est-à-dire qu'un événement permettant l'évolution de $\beta(\lambda)$ se produit. Le seul événement possible, puisque tous les indices des points sont dans $\mathcal{L}$, est l'arrivée de deux indices (ou plus) de $\mathcal{L}$ dans $\mathcal{E}$. En effet, la contrainte égalité du problème 2 ne peut pas être satisfaite si un seul $\beta_i(\lambda)$ varie alors que les autres sont constants. Il faut nécessairement qu'un point de chaque catégorie soit sur la marge pour que cette contrainte soit vérifiée. Pour savoir quels points vont se positionner sur la marge, nous allons chercher, dans $\mathcal{L}$, les indices des points les plus proches de la marge. La valeur de h_λ est connue :
$\forall i \in \mathcal{L},\ \forall \lambda > \lambda_0$,

$$h_\lambda(x_i) = \frac{1}{\lambda}\sum_{j=1}^{m} \beta_j(\lambda)\, y_j \kappa(x_j, x_i) + b_\lambda$$

$$= \frac{1}{\lambda}\sum_{j=1}^{m} y_j \kappa(x_j, x_i) + b_\lambda.$$

Nous allons maintenant identifier, pour chacune des catégories, les points les plus proches de la marge (cf. Fig 2.1). Sans perte de généralité, nous traitons seulement le cas où un seul exemple de chaque catégorie est sur la marge. En effet, dans le cas où plusieurs points de la

même catégorie arrivent en même temps sur la marge, il suffit d'en choisir un pour calculer λ_0 et b_{λ_0}.
Puisque tous les indices appartiennent à $\mathcal{L}$, nous avons pour $i \in I_+$:

$$\frac{1}{\lambda}\sum_{j=1}^{m} y_j \kappa\left(x_j, x_i\right) + b_\lambda < 1$$

et donc le point le plus proche de la marge est celui d'indice $i_+ = \operatorname{argmax}_{i\in I_+} \sum_{j=1}^{m} y_j \kappa\left(x_j, x_i\right)$ puisqu'il maximise le côté gauche de l'inéquation.
De manière analogue, le point d'indice $i_- = \operatorname{argmin}_{i\in I_-} \sum_{j=1}^{m} y_j \kappa\left(x_j, x_i\right)$ est le point de la catégorie "négative" le plus proche de la marge.

Maintenant que les points les plus proches de la marge ont été identifiés, déterminons la valeur de λ_0 pour laquelle leurs indices entrent dans l'ensemble $\mathcal{E}$, c'est-à-dire où $y_{i_+} h_{\lambda_0}\left(x_{i_+}\right) = 1$ et $y_{i_-} h_{\lambda_0}\left(x_{i_-}\right) = 1$:

$$\begin{cases} h_{\lambda_0}\left(x_{i_+}\right) = \frac{1}{\lambda_0}\sum_{j=1}^{m} y_j \kappa\left(x_j, x_{i_+}\right) + b_{\lambda_0} = 1 \\ h_{\lambda_0}\left(x_{i_-}\right) = \frac{1}{\lambda_0}\sum_{j=1}^{m} y_j \kappa\left(x_j, x_{i_-}\right) + b_{\lambda_0} = -1 \end{cases}.$$

La résolution de ce système permet d'obtenir les équations (19) et (20) de [66] :

$$\begin{cases} \lambda_0 = \frac{1}{2}\left(\sum_{j=1}^{m} y_j \kappa\left(x_j, x_{i_+}\right) - \sum_{j=1}^{m} y_j \kappa\left(x_j, x_{i_-}\right)\right) \\ b_{\lambda_0} = -\frac{\sum_{j=1}^{m} y_j \kappa\left(x_j, x_{i_+}\right) + \sum_{j=1}^{m} y_j \kappa\left(x_j, x_{i_-}\right)}{\sum_{j=1}^{m} y_j \kappa\left(x_j, x_{i_+}\right) - \sum_{j=1}^{m} y_j \kappa\left(x_j, x_{i_-}\right)} \end{cases}. \tag{2.3}$$

Une fois le système résolu, nous connaissons complètement le classifieur pour $\lambda = \lambda_0$:
- les multiplicateurs de Lagrange valent tous 1,
- l'ensemble $\mathcal{E}$ ne comporte que i_- et i_+,
- l'ensemble $\mathcal{L}$ comporte les indices de tous les autres exemples,
- l'ensemble $\mathcal{R}$ est vide.

2.1.2.2 Cas $m_+ \neq m_-$

Sans perte de généralité, nous allons traiter le cas où les exemples "positifs" sont plus nombreux que les exemples "négatifs" ($m_+ > m_-$). Hastie et ses coauteurs ont montré que pour une très grande valeur de λ, on peut calculer les multiplicateurs de Lagrange en résolvant un problème de programmation quadratique à m_+ variables au lieu du problème 2 à m variables.

Lemme 4 (Initialisation à coût réduit (lemme 2 de [66])). *Soit β^* défini de la manière suivante :*

$$\beta^* = \operatorname*{argmin}_{\beta} \beta^T H \beta$$

$$s.c. \begin{cases} \forall i \in I_+,\ \beta_i \in [0,1] \\ \forall i \in I_-,\ \beta_i = 1 \\ \sum_{i \in I_+} \beta_i = m_- \end{cases} .$$

Alors il existe λ^* *tel que pour tout* $\lambda > \lambda^*$, $\beta(\lambda) = \beta^*$.

Nous allons redémontrer ce lemme afin de préparer l'étude de l'initialisation du chemin de régularisation de la ℓ_2-SVM.

Démonstration. De la même manière que dans le cas précédent, la norme de $\bar{h}_\lambda = \frac{1}{\lambda} \sum_{i=1}^m \beta_i(\lambda) y_i \kappa_{x_i}$ est dominée par $\frac{1}{\lambda}$ en $+\infty$. Les conditions complémentaires de Kuhn-Tucker nous donnent l'expression des $\xi_i(\lambda)$:

$$\forall i \in I_-,\ \xi_i(\lambda) = (1 + b_\lambda + o(1))_+$$

et

$$\forall i \in I_+,\ \xi_i(\lambda) = (1 - b_\lambda + o(1))_+ .$$

On cherche la valeur de $b_\infty = \lim_{\lambda \to \infty} b_\lambda$. Elle minimise $1_m^T \xi_\infty$ avec $\xi_\infty = \lim_{\lambda \to \infty} \xi(\lambda) = (\xi_{\infty,i})_{1 \leqslant i \leqslant m}$. Afin de trouver le minimum, nous allons évaluer ξ_∞ pour plusieurs valeurs de b_∞.

- Pour $b_\infty \in [-1, 1]$, nous avons

$$\forall i \in I_-,\ \xi_{\infty,i} = 1 + b_\infty$$

et

$$\forall i \in I_+,\ \xi_{\infty,i} = 1 - b_\infty.$$

La connaissance du vecteur ξ_∞ nous permet de calculer la somme de ses composantes

$$\sum_{i=1}^m \xi_{\infty,i} = (1 + b_\infty) m_- + (1 - b_\infty) m_+ = m + b_\infty (m_- - m_+) .$$

La valeur de b_∞ est donc 1, puisque $(m_- - m_+)$ est négatif.

- Pour $b_\infty \geqslant 1$, les exemples positifs sont tous bien classés. Puisque les $\xi_{\infty,i}$ de ces exemples sont nuls, nous avons $\sum_{i=1}^m \xi_{\infty,i} = (1 + b_\infty) m_-$, et cette somme est minimale pour $b_\infty = 1$.
- Pour $b_\infty \leqslant -1$, nous avons $\sum_{i=1}^m \xi_{\infty,i} = (1 - b_\infty) m_+ \geqslant 2m_+ > 2m_-$. La somme de la limite des variables d'écart pour $b_\infty \leqslant -1$ est plus grande que la somme pour $b_\infty = 1$, il n'y a donc pas de minimum global vérifiant $b_\infty \leqslant -1$.

En résumé, la valeur de b_∞ est 1. Démontrons que chaque multiplicateur de Lagrange associé à un exemple de la catégorie "négative" est égal à 1. Pour cela, nous montrons qu'au-delà d'une certaine valeur de λ, tous les exemples de la catégorie "négative" sont mal classés. La convergence

de b_λ vers 1 nous permet d'écrire que

$$\forall \varepsilon > 0,\ \exists \lambda'(\varepsilon) /\ \ \forall \lambda > \lambda'(\varepsilon),\ 1 - \varepsilon < b_\lambda < 1 + \varepsilon.$$

En utilisant cette première inégalité on obtient :

$$\forall i \in [\![1, m]\!],\ \forall \lambda > \lambda'(\varepsilon),\ \bar{h}_\lambda(x_i) + 1 - \varepsilon < h_\lambda(x_i).$$

De plus, puisque $\|\bar{h}_\lambda\|_\kappa$ tend vers zéro, nous avons :

$$\exists \lambda''(\varepsilon) /\ \ \forall \lambda > \lambda''(\varepsilon),\ \forall i \in I_-,\ \|\bar{h}_\lambda\|_\kappa < \varepsilon.$$

En posant $\lambda'''(\varepsilon) = \max(\lambda'(\varepsilon), \lambda''(\varepsilon))$, la relation suivante est vérifiée :

$$\forall \lambda > \lambda'''(\varepsilon),\ \forall i \in I_-,\ 1 - 2\varepsilon < h_\lambda(x_i).$$

En choisissant $\varepsilon = 1/2$, nous avons montré qu'il existe $\lambda^* = \lambda'''(\varepsilon)$ tel que tous les exemples sont classés dans la catégorie "positive", et donc que tous les exemples dont les indices appartiennent à I_- sont mal classés. Puisque ces exemples sont mal classés, les multiplicateurs de Lagrange qui leur sont associés sont égaux à 1. En reportant ceci dans la contrainte égalité du problème 2, on obtient $\sum_{i \in I_+} \beta_i(\lambda) = \sum_{i \in I_-} \beta_i(\lambda) = m_-$.

Rappelons que la fonction objectif du problème d'apprentissage de la machine de norme 1 dans sa formulation duale est : $J_{1,d}(\beta) = -\frac{1}{2}\beta^T H \beta + \lambda 1_m^T \beta$. Puisque, pour $\lambda > \lambda^*$, la somme des composantes de $\beta(\lambda)$ est constante, il suffit de minimiser $\beta^T H \beta$ sur les m_+ variables dont on ne connaît pas la valeur, ce qui achève la démonstration du lemme. □

Ce vecteur β^* possède une propriété intéressante pour caractériser le premier événement : il ne peut pas avoir une et une seule composante comprise strictement entre 0 et 1.[1] Seuls deux cas de figure se présentent :

- soit aucun β_i^* n'appartient à $]0, 1[$, et donc tous valent soit 0 soit 1,
- soit il existe au moins deux β_i^* appartenant à $]0, 1[$.

Puisque tous les exemples de la catégorie "négative" ont des multiplicateurs de Lagrange égaux à 1, la contrainte égalité impose que la somme des valeurs des multiplicateurs de Lagrange des exemples de la catégorie "postive" diminue, il en découle que les multiplicateurs de Lagrange nuls doivent rester nuls et que l'on ne s'intéresse qu'aux multiplicateurs strictement positifs. Dans le premier cas, nous nous retrouvons donc dans le même cas que celui où les cardinaux des catégories sont égaux. Nous identifions alors les exemples les plus proches de la marge :

$$i_- = \underset{i \in I_-}{\operatorname{argmin}} \sum_{j=1}^{m} y_j \kappa(x_j, x_i)$$

1. La somme des β_i pour $i \in I_+$ doit être un nombre entier.

et

$$i_+ = \underset{i \in I_+^1}{\operatorname{argmax}} \sum_{j=1}^{m} y_j \kappa\left(x_j, x_i\right)$$

avec $I_+^1 = \{i \in I_+ / \beta_i^* = 1\}$. Les formules d'obtention de λ_0 et de b_{λ_0} sont identiques à celles du cas $m_+ = m_-$ qui sont données par le système d'équations (2.3).

Dans le second cas, nous choisissons i_+ tel que $\beta_{i_+}^* \in]0,1[$ (sur la marge) et $i_- = \operatorname{argmin}_{i \in I_-} \sum_{j=1}^{m} y_j \kappa\left(x_j, x_i\right)$. On retrouve de nouveau les mêmes formules pour λ_0 et b_{λ_0}.
Puisque la méthode pour calculer les multiplicateurs de Lagrange et la valeur de λ au début du chemin de régularisation est définie, nous allons à présent nous intéresser au troisième point nécessaire afin de suivre ce chemin : l'identification des changements d'ensemble.

2.1.3 Identification des points de transition

La détection des changements d'ensemble s'effectue en recherchant les valeurs de λ pour lesquelles des points transitent entre $\mathcal{E}$ et $\mathcal{L} \cup \mathcal{R}$; ces événements sont appelés les points de transition. Soit λ_l une valeur du paramètre de régularisation pour laquelle un nouveau jeu d'ensembles est défini. Notons $\mathcal{E}_l = \mathcal{E}\left(\lambda_l\right)$. L'équation (2.2) de mise à jour de $\beta\left(\lambda\right)$ est valide si et seulement si l'ensemble $\mathcal{E}$ reste inchangé. Il y a donc deux cas possible :

- une des composantes $\beta_i\left(\lambda\right)$, pour $i \in \mathcal{E}$, devient égale à 0 ou à 1. Cette condition est équivalente au fait de quitter $\mathcal{E}$. En effet, si l'exemple restait dans $\mathcal{E}$, alors l'application de (2.2) entraînerait que $\beta_i\left(\lambda\right)$ devienne négatif (ou supérieur à 1),
- un nouveau point satisfait la condition d'appartenance à $\mathcal{E}$.

Dans le premier cas, pour chaque indice i de $\mathcal{E}$, il faut rechercher pour quelle valeur de λ, la quantité $\beta_i\left(\lambda\right)$ est égale soit à 1 soit à 0. Pour le second, il faut évaluer le classifieur pour chaque exemple dont l'indice appartient à $\mathcal{L} \cup \mathcal{R}$ afin de connaître la valeur de λ pour laquelle cet exemple atteint la marge. Pour ce faire, afin de faciliter l'écriture de h_λ, nous posons

$$\gamma_{\mathcal{E}} = \left(\gamma_{\mathcal{E},i}\right)_{0 \leqslant i \leqslant m_{\mathcal{E}}} = A_{\mathcal{E}}^{-1} \begin{pmatrix} 0 \\ 1_{m_{\mathcal{E}}} \end{pmatrix},$$

l'équation (2.2) se réécrit donc

$$\beta_{\mathcal{E}}^a\left(\lambda + \delta\right) = \beta_{\mathcal{E}}^a\left(\lambda\right) + \delta \gamma_{\mathcal{E}}.$$

Par substitution dans l'expression de $h_{\lambda+\delta}(x)$ donnée par (1.4), nous obtenons :

$$\begin{aligned} h_{\lambda+\delta}(x) &= \frac{1}{\lambda+\delta}\left(\sum_{i=1}^{m}\beta_i(\lambda+\delta)\,y_i\kappa(x_i,x)+\beta_0(\lambda+\delta)\right) \\ &= \frac{1}{\lambda+\delta}\left(\lambda h_\lambda(x)+\delta g(x)\right) \text{ en posant } g(x)=\sum_{i=1}^{m}\gamma_{\mathcal{E},i}y_i\kappa(x_i,x)+\gamma_{\mathcal{E},0} \\ &= \frac{\lambda}{\lambda+\delta}\left(h_\lambda(x)-g(x)\right)+g(x)\,. \end{aligned}$$

Nous retrouvons ainsi l'équation (39) de [66] qui décrit le comportement de la fonction calculée par la SVM. Cette formulation est particulièrement intéressante pour déterminer la valeur de δ telle que l'exemple atteigne la marge. Le plus grand de ces λ satisfaisant $\lambda < \lambda_l$ est le prochain point de transition λ_{l+1}. L'intérêt principal de cette méthode de parcours du chemin de régularisation consiste à ne résoudre que des systèmes linéaires de taille $m_{\mathcal{E}}+1$, dont la complexité en temps est $O\left(m_{\mathcal{E}}^3\right)$. En pratique, $m_{\mathcal{E}}$ peut être très petit devant m, ce qui permet de n'avoir à résoudre que des *petits* systèmes. Les applications de la méthode montrent qu'on obtient effectivement un algorithme rapide. Maintenant que la procédure standard de parcours du chemin de régularisation de la ℓ_1-SVM a été revue en détail, nous nous intéressons au parcours du chemin pour sa variante de norme 2.

2.2 Parcours du chemin de régularisation de la ℓ_2-SVM

Afin de suivre le chemin de régularisation, en prenant inspiration sur le travail de Hastie et coauteurs [66] décrit dans la section 2.1, nous nous concentrons sur la manière dont les multiplicateurs de Lagrange varient en fonction de λ lorsque l'ensemble des multiplicateurs de Lagrange nuls reste inchangé. L'étude du parcours du chemin de régularisation de la ℓ_2-SVM commence donc par la définition d'une partition des indices des exemples. Nous nous intéressons ensuite à la manière de calculer les multiplicateurs de Lagrange pour des ensembles d'indices fixés. Dans une premier temps, nous en fournissons une formule analytique puis présentons des méthodes utilisables en pratiques. Ces calculs étant réalisés à ensemble fixé, les conditions de modification de cette partition sont ensuite explorées. Enfin, le calcul d'un point de départ du chemin de régularisation est proposé en utilisant le comportement asymptotique de la ℓ_2-SVM. Cette section se conclut par une étude de la complexité de l'algorithme proposé.

2.2.1 Partition de l'ensemble d'apprentissage

Les conditions de Kuhn-Tucker permettent de définir, comme pour la ℓ_1-SVM, la partition de l'ensemble d'apprentissage. En effet, les conditions complémentaires de Kuhn-Tucker pour la machine à marge douce et pour sa reformulation en machine à marge dure sont :

$$\forall i \in [\![1,m]\!],\quad \alpha_i(\lambda)\left[y_i h_\lambda(x_i)-1+\xi_i(\lambda)\right]=\alpha_i(\lambda)\left[y_i\tilde{h}_\lambda(x_i)-1\right]=0 \tag{2.4}$$

avec

$$\tilde{h}_\lambda(\cdot) = \sum_{i=1}^{m} y_i \alpha_i(\lambda) \kappa_\lambda(x_i, \cdot) + \alpha_0(\lambda) \tag{2.5}$$

le classifieur calculé par la machine à marge dure, c'est-à-dire en utilisant κ_λ au lieu de κ.

Il résulte de (1.7) et de (2.4) que pour chaque exemple, il y a trois possibilités :

- $y_i h_\lambda(x_i) < 1$: l'exemple est dans la marge ou alors mal classé, ce qui donne $\xi_i(\lambda) > 0$ et $\alpha_i(\lambda) > 0$,
- $y_i h_\lambda(x_i) = 1$: l'exemple est sur la marge, et donc $\alpha_i(\lambda) = 0$,
- $y_i h_\lambda(x_i) > 1$: l'exemple est bien classé et il est en dehors de la marge, et donc $\alpha_i(\lambda) = 0$.

Pour résumer, $\alpha_i(\lambda) > 0$ si et seulement si $y_i h_\lambda(x_i) < 1$. Cette constatation nous amène à proposer de partitionner $[\![1, m]\!]$ en :

$$\begin{aligned} \mathcal{E}(\lambda) &= \{i \in [\![1, m]\!] : y_i h_\lambda(x_i) < 1\} \\ \mathcal{I}(\lambda) &= \{i \in [\![1, m]\!] : y_i h_\lambda(x_i) \geqslant 1\} \quad . \end{aligned} \tag{2.6}$$

Cette partition n'est pas sans rappeler les méthodes d'ensemble actif (voir par exemple la section 10.3 de [45]), en effet $i \in \mathcal{I}(\lambda)$ si et seulement si la contrainte $\alpha_i \geqslant 0$ est active à l'optimum du problème 6. Plus exactement, résoudre le problème 6 sous l'hypothèse que la partition associée à l'optimum est connue se réduit à effectuer la dernière étape d'une méthode d'ensemble actif. Chacune de ces étapes correspond à la résolution d'un système linéaire sur les variables associées aux contraintes actives. Avant de présenter le système linéaire d'intérêt, nous introduisons les notations nécessaires à son écriture concise. De manière analogue à ce que nous avons fait pour la ℓ_1-SVM, nous utilisons $\mathcal{E}$ (respectivement $\mathcal{I}$) plutôt que $\mathcal{E}(\lambda)$ (respectivement $\mathcal{I}(\lambda)$) lorsqu'aucune confusion n'est à craindre. Les cardinalités de ces ensembles sont respectivement notées $m_\mathcal{E}$ et $m_\mathcal{I}$. Par souci de simplicité de langage, nous notons $d_\mathcal{E}$ l'ensemble des exemples dont les indices appartiennent à $\mathcal{E}$. Les exemples sont réordonnés de telle manière que les premiers appartiennent à $\mathcal{E}$. Ainsi nous pouvons introduire des notations compactes et facilement interprétables :

$$y = (y_i)_{1 \leqslant i \leqslant m} = \begin{pmatrix} y_\mathcal{E}^T & y_\mathcal{I}^T \end{pmatrix}^T,$$

$$\alpha(\lambda) = (\alpha_i(\lambda))_{1 \leqslant i \leqslant m} = \begin{pmatrix} \alpha_\mathcal{E}(\lambda)^T & \alpha_\mathcal{I}(\lambda)^T \end{pmatrix}^T,$$

$$d_{\mathcal{E}(\lambda)} = \{(x_i, y_i) \ : \ i \in \mathcal{E}(\lambda)\},$$

et

$$H = \begin{pmatrix} H_{\mathcal{E},\mathcal{E}} & H_{\mathcal{E},\mathcal{I}} \\ H_{\mathcal{I},\mathcal{E}} & H_{\mathcal{I},\mathcal{I}} \end{pmatrix},$$

où la sous-matrice $H_{\mathcal{K},\mathcal{L}}$ avec $(\mathcal{K}, \mathcal{L}) \in \{\mathcal{E}, \mathcal{I}\}^2$ est la matrice $(h_{i,j})_{i \in \mathcal{K}, j \in \mathcal{L}} = (y_i y_j \kappa(x_i, x_j))_{i \in \mathcal{K}, j \in \mathcal{L}}$. La matrice $H(\lambda)$ se partitionne de la même manière que H.

2.2.2 Expression analytique des multiplicateurs de Lagrange

Afin d'obtenir cette expression simplement, nous utilisons l'implication suivante, qui résulte de (2.4) et (1.7) :

$$i \in \mathcal{E} \Longrightarrow y_i \tilde{h}_\lambda (x_i) = 1 \ .$$

En d'autres mots, nous travaillons avec la machine à marge dure. Avec ces notations, en utilisant l'expression de $\tilde{h}_\lambda$, la contrainte égalité du problème 6 et les conditions de Kuhn-Tucker associées aux exemples dans $d_\mathcal{E}$ deviennent :

$$\begin{cases} y_\mathcal{E}^T \alpha_\mathcal{E} (\lambda) + y_\mathcal{I}^T \alpha_\mathcal{I} (\lambda) = 0 \\ H_{\mathcal{E},\mathcal{E}} (\lambda) \alpha_\mathcal{E} (\lambda) + H_{\mathcal{E},\mathcal{I}} (\lambda) \alpha_\mathcal{I} (\lambda) + \alpha_0 (\lambda) y_\mathcal{E} = 1_{m_\mathcal{E}} \end{cases} .$$

Puisque $\alpha_\mathcal{I} (\lambda) = 0$, ces équations se simplifient en

$$\begin{cases} y_\mathcal{E}^T \alpha_\mathcal{E} (\lambda) = 0 \\ H_{\mathcal{E},\mathcal{E}} (\lambda) \alpha_\mathcal{E} (\lambda) + \alpha_0 (\lambda) y_\mathcal{E} = 1_{m_\mathcal{E}} \end{cases} . \tag{2.7}$$

Et donc, en posant

$$A_\mathcal{E} (\lambda) = \begin{pmatrix} 0 & y_\mathcal{E}^T \\ y_\mathcal{E} & H_{\mathcal{E},\mathcal{E}} (\lambda) \end{pmatrix} = \begin{pmatrix} 0 & y_\mathcal{E}^T \\ y_\mathcal{E} & H_{\mathcal{E},\mathcal{E}} + \frac{\lambda}{2} I_{m_\mathcal{E}} \end{pmatrix},$$

$$C_\mathcal{E} = \begin{pmatrix} 0 & 1_{m_\mathcal{E}}^T \end{pmatrix}^T$$

et

$$\alpha_\mathcal{E}^a (\lambda) = \begin{pmatrix} \alpha_0 (\lambda) & \alpha_\mathcal{E} (\lambda)^T \end{pmatrix}^T ,$$

nous obtenons la proposition suivante.

Proposition 5. *Pour tout $\lambda \in \mathbb{R}_+^*$, le vecteur $\alpha_\mathcal{E}^a (\lambda)$ est une solution du système linéaire :*

$$A_\mathcal{E} (\lambda) \alpha_\mathcal{E}^a (\lambda) = C_\mathcal{E} \ . \tag{2.8}$$

Cette écriture est équivalente à celle obtenue dans l'annexe 2.B. de [26]. Une fois que $\mathcal{E}$ est connu, l'apprentissage de la ℓ_2-SVM se réduit à résoudre (2.8). La principale différence avec la formule obtenue pour la ℓ_1-SVM (voir Proposition 4) réside dans la dépendance en λ de la matrice du système linéaire. En effet, il en résulte que les multiplicateurs de Lagrange ne sont plus une fonction linéaire de λ. A la lumière de la proposition 5, le lien entre ℓ_2-SVM et LS-SVM (voir le problème 7) est immédiat.

En effet, résoudre (2.8) peut se voir comme l'apprentissage d'une LS-SVM sur $d_{\mathcal{E}(\lambda)}$, ce qui permet d'énoncer la proposition suivante.

Proposition 6. *L'apprentissage d'une ℓ_2-SVM sur d_m se réduit à l'apprentissage d'une LS-SVM sur $d_{\mathcal{E}(\lambda)}$ dès que l'ensemble $\mathcal{E}(\lambda)$ est connu.*

L'intérêt de cette proposition est double. En effet, elle permet de faire le lien entre l'erreur exacte de validation croisée de la LS-SVM et l'approximation de l'erreur de validation croisée leave-one-out par les spans (voir Equation (1.21)). Mais elle permet aussi de s'inspirer de la LS-SVM pour proposer une méthode pour obtenir les multiplicateurs de Lagrange.

2.2.3 Calcul pratique des multiplicateurs de Lagrange

Notre stratégie de suivi du chemin de régularisation requiert le calcul de plusieurs solutions du système linéaire (2.8), pour différentes valeurs du coefficient λ. Il est donc utile de disposer d'un algorithme dédié. Dans cette section, nous commençons par reformuler l'équation (2.8) à l'aide d'une technique standard puis nous l'utilisons pour obtenir un algorithme permettant de résoudre ce système dans le cadre du parcours du chemin de régularisation pour un coût réduit.

2.2.3.1 Calcul de la solution pour une valeur de λ fixée

Nous prenons notre inspiration de [98] qui utilise une technique habituelle des méthodes d'ensemble actif. Dans un premier temps, remarquons que, puisque $H_{\mathcal{E},\mathcal{E}}$ est une matrice symétrique et semi-définie positive, $H_{\mathcal{E},\mathcal{E}}(\lambda)$ est symétrique définie positive et donc inversible.

L'objectif, ici, est de reformuler (2.8) afin de faire apparaître $H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1}$. A partir de la seconde équation de (2.7), nous obtenons :

$$\alpha_{\mathcal{E}}(\lambda) = H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1}\left(1_{m_{\mathcal{E}}} - \alpha_0(\lambda)\, y_{\mathcal{E}}\right) \quad . \tag{2.9}$$

Par substitution dans la première équation de (2.7), nous avons :

$$y_{\mathcal{E}}^T H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} 1_{m_{\mathcal{E}}} = \alpha_0(\lambda)\, y_{\mathcal{E}}^T H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} y_{\mathcal{E}} \quad . \tag{2.10}$$

Toujours en suivant la démarche de [98], afin d'obtenir une expression compacte de $\alpha_0(\lambda)$ et $\alpha_{\mathcal{E}}(\lambda)$, nous introduisons deux vecteurs de $\mathbb{R}^{m_{\mathcal{E}}}$, ν et ρ, tels que

$$\begin{cases} H_{\mathcal{E},\mathcal{E}}(\lambda)\,\nu = y_{\mathcal{E}} \\ H_{\mathcal{E},\mathcal{E}}(\lambda)\,\rho = 1_{m_{\mathcal{E}}} \end{cases} . \tag{2.11}$$

Par substitution dans (2.9) et (2.10) nous avons

$$\begin{cases} \alpha_{\mathcal{E}}(\lambda) = \rho - \alpha_0(\lambda)\,\nu \\ y_{\mathcal{E}}^T \rho = \alpha_0(\lambda)\, y_{\mathcal{E}}^T \nu \end{cases} .$$

Ainsi, l'expression de $\alpha_0(\lambda)$ et $\alpha_{\mathcal{E}}(\lambda)$ en fonction de ν et ρ est :

$$\begin{cases} \alpha_0(\lambda) = \frac{y_{\mathcal{E}}^T \rho}{y_{\mathcal{E}}^T \nu} \\ \alpha_{\mathcal{E}}(\lambda) = \rho - \alpha_0(\lambda)\nu \end{cases} . \tag{2.12}$$

A la lumière de (2.12), la résolution de (2.8) pour une valeur donnée de λ se réduit à calculer les valeurs correspondantes de ν et de ρ. Puisque la matrice $H_{\mathcal{E},\mathcal{E}}(\lambda)$ est symétrique définie positive, cette résolution peut se faire par l'utilisation d'une décomposition de Cholesky. La décomposition de Gaxpy Cholesky (voir par exemple [54]) dont la complexité est $\frac{1}{3}m_{\mathcal{E}}^3$ est une possibilité. Toutefois, des améliorations significatives peuvent être obtenues en traitant globalement la résolution de l'ensemble de ces systèmes (paramétrés par λ). Nous décrivons à présent cet algorithme qui présente un coût réduit.

2.2.3.2 Calcul d'une série de solutions

Notre méthode fait un usage central du fait que si la matrice H peut être approchée par une matrice de rang faible (indépendant de m) alors la complexité de la résolution du système (2.11) est linéaire en $m_{\mathcal{E}}$. Ce résultat correspond à la proposition 7. Ainsi, nous commençons par introduire une approximation de rang faible de H et nous l'utilisons pour calculer les multiplicateurs de Lagrange. Pour éviter les instabilités numériques, nous définissons une petite constante positive ε qui est supposée suffisament grande pour que $H(\varepsilon)$ soit "numériquement définie positive" (c'est-à-dire que ses valeurs propres ne soient pas trop petites). Pour obtenir une solution approchée de (2.11) pour $\lambda_0 > \varepsilon$ tel que $\mathcal{E}(\lambda_0)$ est connu, nous utilisons une approximation de rang faible de $H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0)}(\varepsilon)$, approximation qui peut être obtenue directement de celle de $H(\varepsilon)$. L'approximation de $H(\varepsilon)$ (et donc H) que nous considérons est une matrice $H_r(\varepsilon)$ satisfaisant

$$H_r(\varepsilon) = R_r R_r^T$$

avec $R_r \in \mathcal{M}_{m,r}(\mathbb{R})$. Cette matrice $H_r(\varepsilon)$ est de rang r et la matrice de Gram associée est de même rang [1]. Remarquons que cette factorisation peut reposer sur le fait que $H(\varepsilon)$ est symétrique définie positive. Comme dans section 2.2.1, nous réordonnons les exemples d'apprentissage et décomposons la matrice R_r en fonction des ensembles $\mathcal{E}$ et $\mathcal{I}$. Nous avons alors $R_r = \begin{pmatrix} R_{\mathcal{E}(\lambda_0)} \\ R_{\mathcal{I}(\lambda_0)} \end{pmatrix}$, avec $R_{\mathcal{E}(\lambda_0)} \in \mathcal{M}_{m_{\mathcal{E}(\lambda_0)},r}(\mathbb{R})$ et $R_{\mathcal{I}(\lambda_0)} \in \mathcal{M}_{m_{\mathcal{I}(\lambda_0)},r}(\mathbb{R})$, et obtenons

$$H_r(\varepsilon) = \begin{pmatrix} H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0),r}(\varepsilon) & H_{\mathcal{E}(\lambda_0),\mathcal{I}(\lambda_0),r}(\varepsilon) \\ H_{\mathcal{I}(\lambda_0),\mathcal{E}(\lambda_0),r}(\varepsilon) & H_{\mathcal{I}(\lambda_0),\mathcal{I}(\lambda_0),r}(\varepsilon) \end{pmatrix} = \begin{pmatrix} R_{\mathcal{E}(\lambda_0)} R_{\mathcal{E}(\lambda_0)}^T & R_{\mathcal{E}(\lambda_0)} R_{\mathcal{I}(\lambda_0)}^T \\ R_{\mathcal{I}(\lambda_0)} R_{\mathcal{E}(\lambda_0)}^T & R_{\mathcal{I}(\lambda_0)} R_{\mathcal{I}(\lambda_0)}^T \end{pmatrix} .$$

1. Pour une idée de la preuve, penser au fait que r représente la dimension de l'espace de représentation. Dans ce cas là, la matrice de Gram et le hessien doivent être de même rang. Pour une preuve analytique, on écrit H_r en fonction de la factorisation de la matrice de Gram puis avec l'inégalité de Sylvester on montre l'égalité de leurs rangs.

Par construction,

$$H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0)}(\lambda_0) = H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0)}(\varepsilon) + \frac{\lambda_0 - \varepsilon}{2} I_{m_{\mathcal{E}(\lambda_0)}} \ ,$$

et donc, $H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0)}(\lambda_0)$ peut être approchée par

$$H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0),r}(\lambda_0,\varepsilon) = H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0),r}(\varepsilon) + \frac{\lambda_0 - \varepsilon}{2} I_{m_{\mathcal{E}(\lambda_0)}} \ .$$

L'intérêt de cette approximation est que finalement, on ne travaille plus qu'avec $R_{\mathcal{E}(\lambda_0)}$, qui peut être directement obtenue à partir de la matrice R_r, c'est-à-dire pour une seule approximation de rang faible de $H(\varepsilon)$. Le fait que $H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0),r}(\varepsilon)$ ne soit pas la matrice que l'on aurait obtenue en calculant l'approximation de rang faible de $H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0)}(\varepsilon)$ ne soulève aucune difficulté puisque nous n'utilisons pas les éventuelles propriétés de cette factorisation. De plus, cette matrice est régulière.

Proposition 7. *Soit le système d'équations linéaires suivant*

$$H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0),r}(\lambda_0,\varepsilon)\, v = z,$$

avec $H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0),r}(\lambda_0,\varepsilon) \in \mathcal{M}_{m_{\mathcal{E}(\lambda_0)},m_{\mathcal{E}(\lambda_0)}}(\mathbb{R})$ *la somme d'une matrice de rang* r *et de la matrice identité multipliée par un scalaire. La complexité de résolution de ce système est linéaire en* $m_{\mathcal{E}(\lambda_0)}$*, plus précisément en* $O\left(m_{\mathcal{E}(\lambda_0)} r^2 + r^3\right)$.

Démonstration. L'application de l'identité de Sherman-Morrison-Woodbury (voir par exemple [54]), permet d'obtenir :

$$\begin{aligned}
\left(H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0),r}(\varepsilon) + \frac{\lambda_0 - \varepsilon}{2} I_{m_{\mathcal{E}(\lambda_0)}}\right)^{-1} &= \left(R_{\mathcal{E}(\lambda_0)} R_{\mathcal{E}(\lambda_0)}^T + \frac{\lambda_0 - \varepsilon}{2} I_{m_{\mathcal{E}(\lambda_0)}}\right)^{-1} \\
&= \frac{2}{\lambda_0 - \varepsilon} I_{m_{\mathcal{E}(\lambda_0)}} - \frac{4}{(\lambda_0 - \varepsilon)^2} R_{\mathcal{E}(\lambda_0)} \left(I_r + \frac{2}{\lambda_0 - \varepsilon} R_{\mathcal{E}(\lambda_0)}^T R_{\mathcal{E}(\lambda_0)}\right)^{-1} R_{\mathcal{E}(\lambda_0)}^T \\
&= \frac{2}{\lambda_0 - \varepsilon} I_{m_{\mathcal{E}(\lambda_0)}} - \frac{4}{(\lambda_0 - \varepsilon)^2} R_{\mathcal{E}(\lambda_0)} T(\lambda_0)^{-1} R_{\mathcal{E}(\lambda_0)}^T \qquad (2.13)
\end{aligned}$$

avec

$$T(\lambda_0) = I_r + \frac{2}{\lambda_0 - \varepsilon} R_{\mathcal{E}(\lambda_0)}^T R_{\mathcal{E}(\lambda_0)} \ .$$

Résoudre le système d'intérêt se réduit à résoudre un système de taille r (celui correspondant à l'inverse de $T(\lambda_0)$) et à effectuer une série de multiplications de matrices et de vecteurs. Une fois que $T(\lambda_0)$ est obtenu en $O\left(m_{\mathcal{E}(\lambda_0)} r^2\right)$ opérations, ce système est résolu par une méthode de Cholesky en seulement $\frac{1}{3} r^3$ opérations (voir la section précédente). Puisque les multiplications de matrices de (2.13) impliquent des matrices de tailles inférieures à $\left(m_{\mathcal{E}(\lambda_0)}, r\right)$, leurs complexités n'excèdent pas $O\left(m_{\mathcal{E}(\lambda_0)} r^2\right)$. La somme de ces deux complexités conclut la preuve. □

Pour obtenir une solution approchée de (2.11) pour $\lambda = \lambda_0$, la matrice $H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0)}(\lambda_0)$ est remplacée par $H_{\mathcal{E}(\lambda_0),\mathcal{E}(\lambda_0),r}(\varepsilon) + \frac{\lambda_0 - \varepsilon}{2} I_{m_{\mathcal{E}(\lambda_0)}}$. Les valeurs correspondantes de ν et ρ sont

calculées par résolution d'un système produisant $T(\lambda_0)$. Pour obtenir un système triangulaire pour chacune des équations, une décomposition de Cholesky de cette matrice est effectuée afin que $T(\lambda_0) = L^T_{\mathcal{E}(\lambda_0)} L_{\mathcal{E}(\lambda_0)}$. Grâce à la proposition 7, la complexité du calcul de ν and ρ se réduit de $O\left(m_{\mathcal{E}}^3\right)$ à $O\left(m_{\mathcal{E}} r^2 + r^3\right)$, ce qui confirme la complexité linéaire en $m_{\mathcal{E}}$ annoncée au début de la section. Cette constatation établit l'avantage de notre algorithme abordant de manière globale la résolution approchée des systèmes (2.11) par rapport aux méthodes traitant indépendamment chacun de ces systèmes. Par abus de notation, nous continuons d'utiliser $\alpha^a_{\mathcal{E}}(\lambda_0) = \begin{pmatrix} \alpha_0(\lambda_0) & \alpha_{\mathcal{E}}(\lambda_0)^T \end{pmatrix}^T$ pour désigner la solution approchée de (2.8) (pour $\lambda = \lambda_0$). Les détails concernant la complexité et la conséquence de cette approximation sont donnés dans la section 2.2.6. Puisque cette approche considère que $\mathcal{E}(\lambda_0)$ est connu, nous nous intéressons à présent à la méthode proposée pour suivre l'évolution de cet ensemble.

2.2.4 Détection des changements dans la partition de l'ensemble d'apprentissage

Soit $\left(\lambda^l\right)_{l \in \mathbb{N}^*}$ la séquence strictement décroissante de valeurs du coefficient de régularisation associée aux modifications des ensembles $\mathcal{E}$ et $\mathcal{I}$. Ces événements peuvent être directement inférés des définitions de ces ensembles. En effet, si la séquence est connue jusqu'à son terme d'indice l, alors trouver λ^{l+1} revient à identifier la plus grande valeur de λ dans l'intervalle $\left(0, \lambda^l\right)$ pour laquelle un nouvel indice $i \in [\![1, m]\!]$ vérifie $y_i h_\lambda(x_i) = 1$, c'est-à-dire résoudre en λ l'ensemble des équations indexées par i :

$$y_i h_\lambda(x_i) = H_{i,\mathcal{E}(\lambda^l)} \alpha_{\mathcal{E}(\lambda^l)}(\lambda) + y_i \alpha_0(\lambda) = 1 \ . \tag{2.14}$$

L'implantation naïve de cette méthode la rend inapplicable pour les grandes valeurs de $m_{\mathcal{E}}$, et nous verrons dans la section 2.2.5 que le cas extrême $\mathcal{E}(\lambda) = [\![1, m]\!]$ est atteint en pratique. Heureusement, le système (2.12) nous fournit une expression analytique des variables duales qui permet d'écrire le développement de Taylor du premier ordre de h_λ. Ce développement est alors utilisée comme approximation de cette fonction. Dans ce but, nous définissons

$$\forall i \in [\![1, m]\!], \ \ \lambda_i^{l+1} = \lambda^l - \frac{1 - y_i h_{\lambda^l}(x_i)}{\left.\frac{\partial(1 - y_i h_\lambda(x_i))}{\partial \lambda}\right|_{\lambda = \lambda^l}} = \lambda^l + \frac{1 - y_i h_{\lambda^l}(x_i)}{\left. y_i \frac{\partial h_\lambda(x_i)}{\partial \lambda}\right|_{\lambda = \lambda^l}} \ .$$

L'expression de $\frac{\partial h_\lambda(x_i)}{\partial \lambda}$ s'obtient facilement une fois que $\frac{\partial \alpha_0(\lambda)}{\partial \lambda}$ et $\frac{\partial \alpha_{\mathcal{E}}(\lambda)}{\partial \lambda}$ sont connues. Ces quantités sont obtenues à partir de (2.11) :

$$\begin{cases} \frac{\partial \alpha_0(\lambda)}{\partial \lambda} = -\frac{1}{2} \frac{\nu^T \rho y_{\mathcal{E}}^T \nu - \nu^T \nu y_{\mathcal{E}}^T \rho}{\left(y_{\mathcal{E}}^T \nu\right)^2} \\ \frac{\partial \alpha_{\mathcal{E}}(\lambda)}{\partial \lambda} = -\frac{1}{2} H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} \alpha_{\mathcal{E}}(\lambda) - \frac{\partial \alpha_0(\lambda)}{\partial \lambda} \nu \end{cases} . \tag{2.15}$$

Si $\left(\lambda_i^{l+1}\right)_{i \in [\![1,m]\!]}$ étaient le vecteur des racines des équations (2.14), alors l'expression de λ^{l+1}

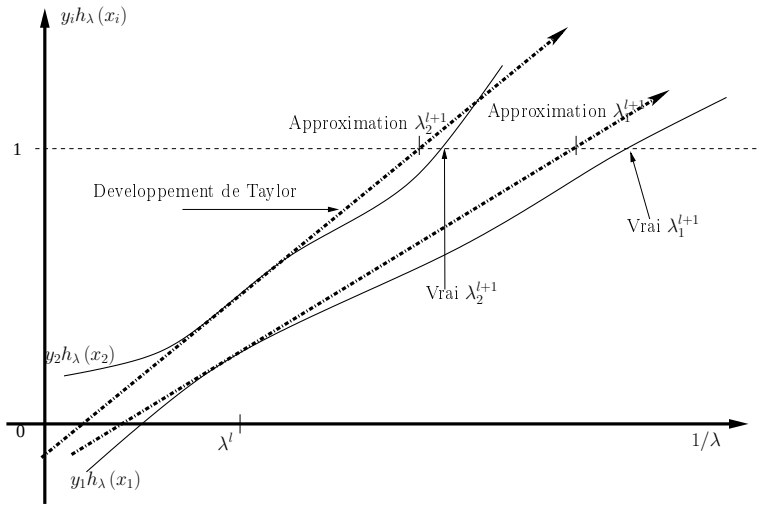


FIGURE 2.2 – Illustration du développement de Taylor utilisé pour la détection de changement d'ensembles. Il s'agit du développement au premier ordre de $y_i h_\lambda(x_i)$ pour deux exemples x_1 et x_2.

serait simplement

$$\lambda^{l+1} = \max_{\left\{i:\lambda_i^{l+1}<\lambda^l\right\}} \lambda_i^{l+1} \ .$$

Toutefois, l'utilisation d'une approximation nécessite une autre formule pour éviter à l'algorithme d'effectuer des pas *vides* (c'est-à-dire sans changement d'ensemble) ou des pas *trop grands*. Nous avons trouvé que la règle empirique suivante était particulièrement efficace : λ^{l+1} est choisi tel que 2% des indices de $\mathcal{E}\left(\lambda^l\right)$ satisfont $\lambda^{l+1} < \lambda_i^{l+1} < \lambda^l$. Cette approximation peut néanmoins fournir des solutions inappropriées lors de l'existence de plateaux. Pour éviter ces situations, lorsque λ^{l+1} devrait être choisi plus petit que $\frac{1}{2}\lambda^l$, il est alors pris égal à $\frac{1}{2}\lambda^l$. Nos expériences ont montré que ce problème n'a lieu que lorsque λ est très grand. La prédiction de la partition des exemples d'apprentissage pour λ^{l+1} se base simplement sur le signe de l'approximation du premier ordre de $1 - y_i h_\lambda(x_i)$ (penser à la définition des ensembles $\mathcal{E}$ et $\mathcal{I}$). De part le comportement non linéaire des multiplicateurs de Lagrange, cette prédiction doit être corrigée. La partition exacte est alors obtenue par une méthode d'ensemble actifs. Pour une partition donnée, la valeur des multiplicateurs de Lagrange est calculée puis la consistance du résultat est vérifiée. Si ce n'est pas le cas, une nouvelle partition est calculée et ainsi de suite. Puisque la partition prédite est proche de l'optimale, cette étape de correction est peu coûteuse. Dans les simulations, hormis pour les très petites valeurs de λ, la convergence est atteinte en une seule mise à jour.

2.2.5 Calcul du point de départ du chemin de régularisation

De manière analogue à la section 2.1.2, afin de définir un point de départ pour le chemin de régularisation, nous tirons profit de la particularité du problème 6 lorsque λ tend vers l'infini. Asymptotiquement, $H(\lambda)$ est équivalent à $\frac{\lambda}{2} I_m$, c'est-à-dire que les exemples d'apprentissage

n'interviennent plus que par leurs catégories par le biais de la contrainte $y^T\alpha = 0$. Le problème d'apprentissage est alors "équivalent" à

Problème 14 (Problème équivalent au problème 6 quand λ tend vers l'infini)**.**

$$\max_{\alpha} \left\{ -\frac{\lambda}{4}\alpha^T\alpha + 1_m^T\alpha \right\}$$

$$s.c. \quad \begin{cases} \forall i \in [\![1, m]\!], \ \ \alpha_i \geqslant 0 \\ y^T\alpha = 0 \end{cases} .$$

Dans [28] (Annexe C), les auteurs annoncent que la solution du problème 14 est :

$$\forall i \in [\![1, m]\!], \quad \begin{cases} \alpha_i(\lambda) = \frac{4m_-}{m\lambda} \text{ si } y_i = 1 \\ \alpha_i(\lambda) = \frac{4m_+}{m\lambda} \text{ sinon} \end{cases} \tag{2.16}$$

Ces valeurs s'obtiennent facilement en substituant la contrainte égalité dans la fonction objectif puis en cherchant le minimum de la parabole (en une seule dimension). Une fois les valeurs des multiplicateurs connues, pour connaître le comportement asymptotique du classifieur, il suffit de calculer la limite b_∞ de b_λ lorsque λ tend vers l'infini. Dans ce but, nous faisons usage du fait que puisque tous les multiplicateurs de Lagrange sont asymptotiquement positifs, alors les conditions complémentaires de Kuhn-Tucker (2.4) imposent que pour tout i dans $[\![1, m]\!]$, $y_i\tilde{h}_\lambda(x_i)$ tend vers 1. Ainsi, les équations (2.5) et (2.16) donnent

$$b_\infty = \frac{m_+ - m_-}{m} .$$

La conséquence directe de cette équation est que si $m_+ \neq m_-$, alors le classifieur limite retournera toujours la catégorie de plus grande cardinalité. De plus, contrairement à la ℓ_1-SVM, il n'existe pas de valeur λ_∞ telle que pour toute valeur de λ supérieure les multiplicateurs de Lagrange soient constants. Tout ce que l'on sait est le comportement asymptotique des multiplicateurs de Lagrange, ce qui n'est pas suffisant pour définir un point de départ de manière analogue à celle de la ℓ_1-SVM (voir 2.1.2). Toutefois, cette connaissance est utile pour obtenir deux critères. Le premier permet de savoir si une valeur de λ est suffisamment grande pour être la valeur de λ associée au point de départ du chemin (noté par la suite λ^1). Le second critère permet de savoir quand commencer à calculer le critère de sélection de modèle [1].

2.2.6 Vue d'ensemble de l'algorithme et analyse de la complexité

La figure 2.3 présente la synthèse des différents éléments constitutifs de l'algorithme de parcours du chemin de régularisation.

Il y a cinq éléments principaux :

1. Un classifieur ne renvoyant toujours qu'une catégorie ne peut pas être optimal pour un problème non trivial.

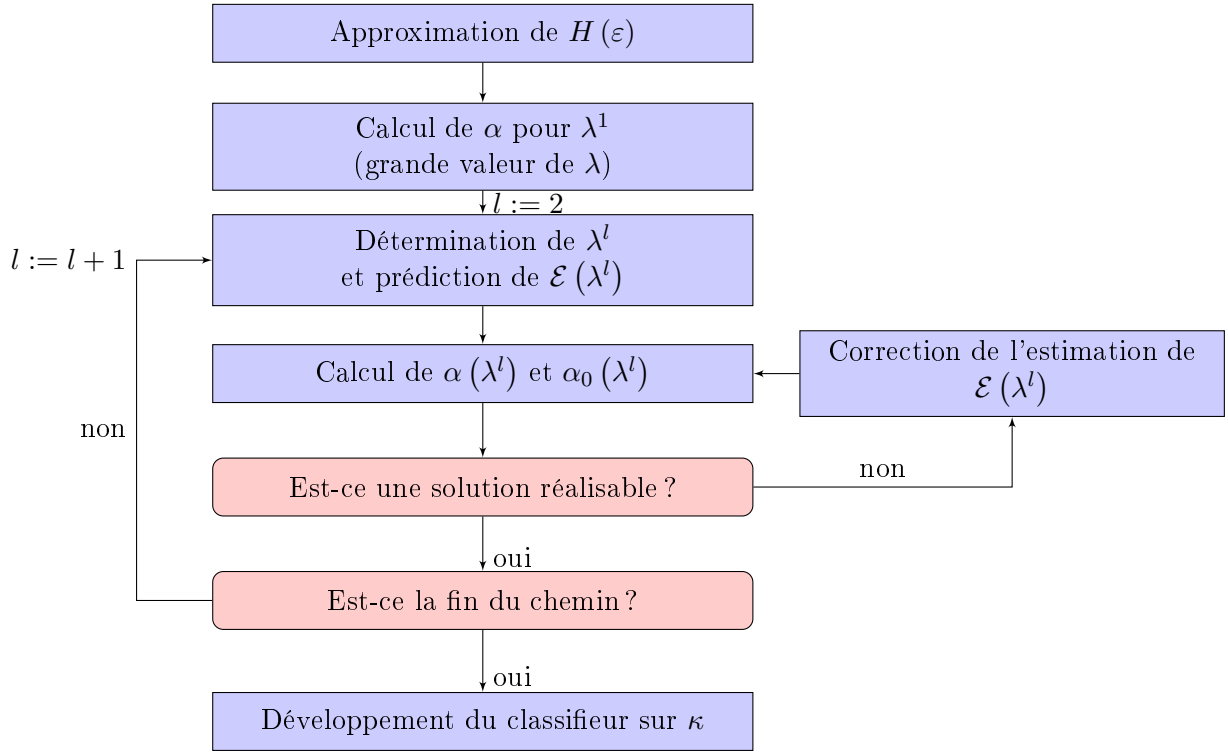


FIGURE 2.3 – Diagramme de l'algorithme de parcours du chemin de régularisation pour la ℓ_2-SVM

- L'approximation de rang faible de $H(\varepsilon)$,
- Le calcul des multiplicateurs de Lagrange,
- La boucle extérieure qui engendre la suite de (λ^l),
- la boucle intérieure qui met à jour $\mathcal{E}$ (pour la valeur courante de λ),
- le recalcul du classifieur obtenu.

Nous présentons à présent les complexités de chacun de ces éléments et les choix concernant leur mise en œuvre.

2.2.6.1 Approximation de rang faible du hessien

Nous nous intéressons à présent à l'obtention de $H_r(\varepsilon)$ et aux conséquences de cette approximation. Parmi les différents algorithmes possibles pour cette tâche, nous choisissons de retenir la factorisation incomplète de Cholesky (Incomplete Cholesky Factorization, ICF) proposée dans [44], la décomposition en valeurs propres tronquées pour ne retenir que les plus grandes valeurs propres et la méthode de Nyström pondérée par les densités [112]. Leurs complexités respectives sont en $O(mr^2)$, $O(m^3)$ et $O(l^3 + lm)$ avec l le nombre de *landmarks*. Le nombre de *landmarks* est un paramètre interne de l'algorithme de [112] qui correspond au nombre de centroïdes utilisés pour estimer la densité des exemples. On peut aussi voir l comme un majorant de r. D'après les résultats présentés dans l'étude comparative de [113], nous choisissons de retenir les

méthodes de Nyström [1]. Il est intéressant de remarquer que l'utilisation d'une décomposition en valeurs propres dans notre méthode correspond à effectuer une analyse en composante principale *kernelisée* [93] avant d'entraîner une SVM linéaire.

Si le RKHS engendré par κ est de dimension finie, le choix de l est évident puisque r est le minimum entre le nombre d'exemples et la dimension de ceux-ci. Dans le cas contraire, les matrices $H(\varepsilon)$ et $H_r(\varepsilon)$ ne seront en général pas égales, ce qui rend le choix de r plus difficile. Ce choix est d'une importance cruciale pour l'apprentissage puisqu'il impacte la capacité de la classe de fonctions (voir la *Kernel Projection Machine* [14] qui travaille sur un espace construit à partir des fonctions propres du noyau). En effet, moins on choisit de fonctions propres plus on réduit la taille de la classe de fonctions considérée. Le choix d'un rang r faible a donc un effet régularisant. Les deux principales options consistent soit à pré-spécifier un rang soit à sélectionner toutes les valeurs propres au-dessus d'un seuil donné. La méthode du rang pré-spécifié peut être reliée à l'apprentissage d'un classifieur de complexité réduite (voir par exemple [72]). L'utilisation d'un seuil sur les valeurs propres permet de contrôler la précision de la solution [2] et donc de limiter la possibilité d'imposer une régularisation trop forte au problème d'apprentissage. Pour la régression ridge *kernelisée*, la proposition 1 de [31] donne une borne supérieure sur la différence des fonctions calculées avec et sans l'approximation. Lorsque le modèle est linéaire (et non affine) dans l'espace de représentation, cette proposition est valable pour la LS-SVM et donc pour la ℓ_2-SVM. Un tel résultat permet de comprendre l'impact de l'approximation de la matrice $H(\varepsilon)$ et du coefficient de régularisation sur le classifieur. L'avantage de cette technique de seuil est qu'elle permet de s'assurer que $H_r(\varepsilon)$ est symétrique définie positif. Ainsi, cette sélection du rang par seuillage des valeurs propres sera utilisée dans les expériences. Il est important de garder à l'esprit que l'approximation de $H(\varepsilon)$ ne laisse pas le problème d'apprentissage inchangé. En effet, elle introduit un changement de noyau (qui est d'autant plus petit que l'approximation est exacte). Dans le cas bi-classe, nous n'utiliserons pas l'expression analytique de κ_r (voir l'expression analytique en fonction des fonctions propres dans [53]) en lui préférant une autre approche qui est décrite dans la partie dédiée au développement du classifieur sur κ (paragraphe 2.2.6.3)

2.2.6.2 Analyse de la complexité de la boucle extérieure

Comme vu en section 2.2.4, le calcul de λ^{l+1} et la prédiction de l'ensemble $\mathcal{E}\left(\lambda^{l+1}\right)$ font usage du calcul de la dérivée de $\alpha^a_{\mathcal{E}}$ par rapport à λ. Cette dérivée (2.15) est obtenue en résolvant un système linéaire construit autour de la matrice $H_{\mathcal{E},\mathcal{E}}\left(\lambda^l\right)^{-1}$ déjà rencontrée lors du calcul de la valeur des variables duales (2.11). Le calcul de la dérivée tire profit de la disponibilité de la décomposition de Cholesky de $T\left(\lambda^l\right)$. La conséquence directe est que la résolution du système est simplement obtenue par deux résolutions de systèmes triangulaires et quelques produits de matrices pour une complexité en $2m_{\mathcal{E}\left(\lambda^l\right)}r + 2r^2 + 4m_{\mathcal{E}\left(\lambda^l\right)}$ opérations. Le calcul des valeurs λ_i^{l+1} se fait en $4mr + m$ opérations. La complexité globale du calcul de λ^{l+1} est donc en $O\left(mr + r^2\right)$.

1. Nous nous autorisons à choisir des méthodes de Nyström classique ou une méthode ponderée.
2. Le contrôle de la précision se fait par l'utilisation d'une norme sur la différence de H et H_r.

La complexité pour obtenir la solution approchée de (2.11) en λ^{l+1} est en $O\left(m_{\mathcal{E}\left(\lambda^{l+1}\right)}r + r^3\right)$ une fois que la matrice $R^T_{\mathcal{E}\left(\lambda^{l+1}\right)}R_{\mathcal{E}\left(\lambda^{l+1}\right)}$ est connue. Le calcul naïf de ce produit $R^T_{\mathcal{E}\left(\lambda^{l+1}\right)}R_{\mathcal{E}\left(\lambda^{l+1}\right)}$ nécessite $m_{\mathcal{E}\left(\lambda^{l+1}\right)}r^2$ opérations faisant de lui la tâche la plus coûteuse de l'algorithme. En pratique, nous utilisons à notre avantage le fait que l'ensemble $\mathcal{E}$ change lentement [1] pour calculer une mise à jour de la matrice $R^T_{\mathcal{E}\left(\lambda^l\right)}R_{\mathcal{E}\left(\lambda^l\right)}$ qui coûte seulement $2r^2\Delta\mathcal{E}\left(\lambda^{l+1},\lambda^l\right)$ opérations avec $\Delta\mathcal{E}\left(\lambda^{l+1},\lambda^l\right) = \left|m_{\mathcal{E}\left(\lambda^{l+1}\right)} - m_{\mathcal{E}\left(\lambda^l\right)}\right|$ la valeur absolue de la différence des cardinaux de deux ensembles consécutifs. La manière de choisir λ^{l+1} (voir 2.2.4) permet d'espérer que $\Delta\mathcal{E}\left(\lambda^{l+1},\lambda^l\right)$ soit proche de 2% de $m_{\mathcal{E}\left(\lambda^l\right)}$.

2.2.6.3 Développement de h_λ en κ

Comme annoncé en section 2.2.6.1, le classifieur retourné par l'algorithme est construit sur le noyau κ_r (dont le calcul par sa formule analytique demande $O(mr)$ évaluations du noyau κ). Afin d'obtenir un classifieur $\hat{h}_\lambda$ utilisable facilement en test, nous proposons de revenir de κ_r vers κ en nous inspirant de [14]. Soit un sous-ensemble $\mathcal{J}$ de $[\![1,m]\!]$, nous cherchons le vecteur de coefficients $\gamma(\lambda) \in \mathbb{R}^{m_\mathcal{J}}$ tel que

$$\hat{h}_\lambda = \sum_{i\in\mathcal{J}} \gamma_i(\lambda)\, y_i \kappa_{x_i} + \alpha_0(\lambda)$$

et que $\hat{h}_\lambda$ soit identique au classifieur construit avec κ_r sur l'ensemble d'apprentissage, soit,

$$\left(y_i\left[\hat{h}_\lambda(x_i) - \alpha_0(\lambda)\right]\right)_{1\leqslant i\leqslant m} = \left(y_i \sum_{j=1}^{m_{\mathcal{E}(\lambda)}} \alpha_j(\lambda)\, y_j \kappa_r(x_j,x_i)\right)_{1\leqslant i\leqslant m} .$$

Cette dernière équation peut être reformulée algébriquement en :

$$H_{\cdot,\mathcal{J}}(\varepsilon)\,\gamma(\lambda) = R_r R^T_{\mathcal{E}(\lambda)}\alpha_\mathcal{E}(\lambda) \quad .$$

Dans le cas d'un espace de représentation de dimension infinie, nous suggérons de choisir un ensemble $\mathcal{J}$ égal à $\mathcal{E}(\lambda)$ afin de conserver la parcimonie de la SVM. Ainsi l'obtention de $\gamma(\lambda)$ nécessite $O\left(m^3_{\mathcal{E}(\lambda)}\right)$ opérations. Lorsque la dimension de l'espace de représentation est r, on peut choisir $m_\mathcal{J} = r$ et ainsi calculer $\gamma(\lambda)$ en seulement $O\left(r^3\right)$ opérations. [2] Ce résultat prouve l'intérêt d'utiliser des polynômes de degré faible puisque cela induit que le temps pour évaluer $\hat{h}_\lambda$ sur un exemple de test dépend de r plutôt que $m_{\mathcal{E}(\lambda)}$. Bien évidemment, cela suppose que le nombre d'exemples est très supérieur à r.

1. La lenteur de variation de $\mathcal{E}$ est obtenue par la manière de choisir λ^{l+1}.
2. En effet, puisque H est de rang r, seuls r colonnes sont linéairement indépendantes, et donc il suffit d'en choisir r.

2.3 Sélection de modèle pour la ℓ_2-SVM

Dans la section 1.4.2, différentes méthodes permettant d'évaluer la qualité d'un modèle ont été présentées, nous ne les discuterons pas en détail ici. Nous montrons dans un premier temps que le calcul de l'erreur de validation croisée leave-one-out appliqué à la LS-SVM associée à une ℓ_2-SVM correspond à l'approximation de l'erreur de validation croisée leave-one-out présentée dans [27]. A partir du constat que l'hypothèse sur les vecteurs support correspond à manipuler la LS-SVM associée (proposition 6), nous présentons une approximation de l'estimation bootstrap leave-one-out qui permet naturellement d'obtenir une approximation du bootstrap .632+. A l'aide de cette formulation, nous proposons aussi une approximation de l'erreur de validation croisée à V pas se basant sur le même concept. La motivation pour développer des approximations d'autres estimateurs de l'erreur en généralisation que celui de la validation croisée leave-one-out vient de [22]. En effet, les auteurs y montrent que la variance est aussi un élément à prendre en compte dans la sélection de modèle puisqu'elle peut induire du sur-apprentissage. Les caractéristiques du bootstrap .632+ en font donc naturellement un bon candidat.

2.3.1 Intégration de l'approximation de l'erreur de validation croisée leave-one-out

Cette section propose une implantation efficace de l'approximation de l'erreur de validation croisée leave-one-out (ou simplement approximation de l'erreur de test) dans le cadre du parcours du chemin de régularisation lorsque le rang de la matrice de Gram est de rang faible. Tout d'abord, nous démontrons que la méthode de Cawley et coauteurs [21] pour calculer l'erreur de test de la procédure de validation croisée leave-one-out pour la LS-SVM entraînée sur $d_{\mathcal{E}}$ est égale à l'approximation de l'erreur de validation croisée leave-one-out de la ℓ_2-SVM. Nous présentons ensuite un moyen pour calculer cette approximation lorsque $H_r(\epsilon)$ est de rang faible, auquel cas, la complexité est linéaire en $m_{\mathcal{E}}$.

Proposition 8. *Le nombre d'erreur associé à l'approximation de l'erreur de validation croisée leave-one-out de la ℓ_2-SVM est égal à celui de la procédure de validation croisée leave-one-out de la LS-SVM entrainée sur $d_{\mathcal{E}}$ qui est*

$$\sum_{i\in\mathcal{E}} \mathbb{1}_{\left\{y_i h^i_\lambda(x_i)\leqslant 0\right\}} = \sum_{i\in\mathcal{E}} \mathbb{1}_{\left\{\frac{\alpha_i(\lambda)}{\left(A_{\mathcal{E}}(\lambda)^{-1}\right)_{i,i}}-1\geqslant 0\right\}} \ . \tag{2.17}$$

Cette proposition est contenue dans l'énoncé de cette approximation fait par Chapelle et coauteurs dans [105]. Toutefois, nous le redémontrons ici afin de mettre en évidence des relations utilisées pour obtenir la complexité linéaire en nombre d'exemple. De plus, cette proposition est particulièrement intéressante puisqu'elle montre que l'approche géométrique de la notion d'espace engendré coïncide avec l'approche purement analytique de [20].

Démonstration. Soit $\alpha^i_0(\lambda)$, $\alpha^i_{\mathcal{E}}(\lambda)$ et $\tilde{h}^i_\lambda$ respectivement le biais, le vecteur des multiplicateurs

de Lagrange et la fonction calculée par la LS-SVM entraînée sur $d_{\mathcal{E}} \setminus \{(x_i, y_i)\}$ avec le noyau κ_λ. A la fin de la preuve, nous montrerons que $\tilde{h}^i_\lambda(x_i)$ est égal à $h^i_\lambda(x_i)$ et que c'est donc la quantité d'intérêt de la procédure de validation croisée leave-one-out.

Cawley et coauteurs ont montré dans [21] que $\tilde{h}^i_\lambda(x_i)$ peut être déduit des quantités impliquées dans l'expression de la LS-SVM entrainée sur l'ensemble d'apprentissage complet d_m. Leur démonstration fait usage de la matrice de Gram, mais nous la présenterons en utilisant la matrice $H(\lambda)$ afin de l'intégrer dans le cadre de l'étude.

Puisque la démonstration fait une utilisation intensive de (2.8), rappelons la :

$$\begin{pmatrix} 0 & y_{\mathcal{E}}^T \\ y_{\mathcal{E}} & H_{\mathcal{E},\mathcal{E}} + \frac{\lambda}{2} I_{m_{\mathcal{E}}} \end{pmatrix} \begin{pmatrix} \alpha_0(\lambda) \\ \alpha_{\mathcal{E}}(\lambda) \end{pmatrix} = \begin{pmatrix} 0 \\ 1_{m_{\mathcal{E}}} \end{pmatrix} .$$

Afin de faciliter les notations, les coefficients de $A_{\mathcal{E}}(\lambda)$ varient de 0 à $m_{\mathcal{E}}$ de manière à ce que les indices des colonnes correspondent à ceux des indices de $\alpha^a_{\mathcal{E}}(\lambda)$:

$$A_{\mathcal{E}}(\lambda) = (a_{i,j})_{0 \leqslant i,j \leqslant m_{\mathcal{E}}} .$$

A partir de cette matrice, nous définissons :

- $A^i \in \mathcal{M}_{m_{\mathcal{E}}, m_{\mathcal{E}}}(\mathbb{R})$ la matrice $A_{\mathcal{E}}(\lambda)$ dont les lignes et les colonnes d'indices i ont été enlevés,
- $a_i \in \mathbb{R}^{m_{\mathcal{E}}}$ la colonne d'indice i dont le coefficient d'indice i est enlevé,
- $a_{i,i} \in \mathbb{R}$ le coefficient d'indice (i,i).

Avec ces notations, la ligne d'indice i ($i \in [\![1, m_{\mathcal{E}}]\!]$) du système (2.8), correspondant à la condition de Kuhn-Tucker pour l'exemple i, peut être réécrite en

$$a_i^T \left(\alpha_0(\lambda)\, \alpha_{\mathcal{E}\setminus\{i\}}(\lambda)^T \right)^T = 1 - a_{i,i} \alpha_i(\lambda) \tag{2.18}$$

alors que les autres lignes en

$$\left(A^i \; a_i\right) \left(\alpha_0(\lambda)\, \alpha_{\mathcal{E}\setminus\{i\}}(\lambda)^T \, \alpha_i(\lambda) \right)^T = \left(0 \; 1^T_{m_{\mathcal{E}}-1}\right)^T \quad . \tag{2.19}$$

Cette décomposition de $A_{\mathcal{E}}(\lambda)$ permet d'exprimer le problème d'apprentissage de la LS-SVM entrainée sur $\mathcal{E} \setminus \{i\}$ en terme de A^i :

$$A^i \begin{pmatrix} \alpha^i_0(\lambda) \\ \alpha^i_{\mathcal{E}}(\lambda) \end{pmatrix} = \begin{pmatrix} 0 \\ 1_{m_{\mathcal{E}}-1} \end{pmatrix} . \tag{2.20}$$

Par la définition de a_i, nous avons $y_i \tilde{h}^i_\lambda(x_i) = H_{i,\mathcal{E}}(\lambda)\, \alpha^i_{\mathcal{E}}(\lambda) + y_i \alpha^i_0(\lambda) = a_i^T \left(\alpha^i_0(\lambda) \;\; \alpha^i_{\mathcal{E}}(\lambda)^T \right)^T$. Par substitution de (2.20) dans cette expression :

$$y_i \tilde{h}^i_\lambda(x_i) = a_i^T \left(A^i\right)^{-1} \left(0 \; 1^T_{m_{\mathcal{E}}-1}\right)^T \quad .$$

Les équations (2.18) et (2.19) permettent de simplifier la formule afin d'exprimer $y_i \tilde{h}^i_\lambda(x_i)$ en fonction de $\alpha_i(\lambda)$:

$$\begin{aligned} y_i \tilde{h}^i_\lambda(x_i) &= a_i^T \left(A^i\right)^{-1} \left(A^i \; a_i\right) \left(\alpha_0(\lambda) \;\; \alpha_{\mathcal{E}\setminus\{i\}}(\lambda)^T \alpha_i(\lambda)\right)^T \\ &= a_i^T \left(A^i\right)^{-1} A^i \left(\alpha_0(\lambda)\, \alpha_{\mathcal{E}\setminus\{i\}}(\lambda)^T\right)^T + \alpha_i(\lambda)\, a_i^T \left(A^i\right)^{-1} a_i \\ &= 1 - \alpha_i(\lambda)\, a_{i,i} + \alpha_i(\lambda)\, a_i^T \left(A^i\right)^{-1} a_i \\ &= 1 - \alpha_i(\lambda) \left(a_{i,i} - a_i^T \left(A^i\right)^{-1} a_i\right) \\ &= 1 - \frac{\alpha_i(\lambda)}{\left(A_\mathcal{E}(\lambda)^{-1}\right)_{i,i}} \quad . \end{aligned} \tag{2.21}$$

La dernière ligne vient de l'application de la formule d'inversion par bloc. De plus les sorties, pour l'exemple i, du classifieur à marge dure équivalent et celui à marge douce (c'est-à-dire construit sur κ_λ ou κ) sont égales :

$$\begin{aligned} \tilde{h}^i_\lambda(x_i) &= \sum_{j \neq i} \left(\alpha_j(\lambda)\, y_j \kappa(x_i, x_j) + \delta_{i,j} \frac{\lambda}{2}\right) + \alpha_0(\lambda) \\ &= \sum_{j \neq i} \alpha_j(\lambda)\, y_j \kappa(x_i, x_j) + \alpha_0(\lambda) \\ &= h^i_\lambda(x_i) \quad . \end{aligned} \tag{2.22}$$

L'utilisation conjointe de (2.21) et (2.22) permet d'obtenir le nombre d'exemples de $d_\mathcal{E}$ qui sont mal classés par la LS-SVM entraînée sur $d_\mathcal{E}$. Comme seuls les exemples de $d_\mathcal{E}$ contribuent à l'erreur de la ℓ_2-SVM [1], le nombre d'erreurs de (1.21) et de (2.17) sont les mêmes, ce qui conclut la démonstration. □

L'intérêt de cette proposition est double. En effet, cette approximation de l'erreur de validation croisée leave-one-out est connue pour avoir de bonnes propriétés en sélection de modèle mais de plus offre la possibilité d'être calculée efficacement. Comme vu précédemment, la matrice $A_\mathcal{E}(\lambda)$ joue un rôle central dans la sélection de modèle. La formule d'inversion par bloc nous permet d'obtenir les coefficients de la diagonale de l'inverse de $A_\mathcal{E}(\lambda)$:

$$\begin{aligned} A_\mathcal{E}(\lambda)^{-1} &= \begin{pmatrix} s^{-1} & -s^{-1} y_\mathcal{E}^T H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} \\ -H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} y_\mathcal{E} s^{-1} & H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} + H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} y_\mathcal{E} s^{-1} y_\mathcal{E}^T H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{s} & -\frac{1}{s}\nu^T \\ -\frac{1}{s}\nu & H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} + \frac{1}{s}\nu\nu^T \end{pmatrix} \end{aligned} \tag{2.23}$$

avec $s = -y_\mathcal{E}^T H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} y_\mathcal{E} = -y_\mathcal{E}^T \nu$.

1. Les exemples qui ne sont pas vecteurs support n'interviennent pas dans la procédure de validation croisée. En effet, si on les enlève de l'ensemble d'apprentissage le classifieur reste inchangé.

Pour $i \in [\![1, m_{\mathcal{E}}]\!]$, les coefficients de la diagonale sont :

$$A_{\mathcal{E}}(\lambda)^{-1}_{i,i} = \left(H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1}\right)_{i,i} + \frac{1}{s}\nu_i^2 \ . \tag{2.24}$$

Grâce à l'approximation de rang faible de $H_{\mathcal{E},\mathcal{E}}(\varepsilon)$, la complexité du calcul des coefficients de la diagonale de l'inverse de cette matrice est linéaire en $m_{\mathcal{E}}$. En effet, en conservant les notations de la section 2.2.3, nous obtenons

$$\begin{aligned} H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} \simeq \left(H_{\mathcal{E},\mathcal{E},r}(\varepsilon) + \frac{\lambda-\varepsilon}{2} I_{m\varepsilon}\right)^{-1} &= \frac{2}{\lambda-\varepsilon} I_{m\varepsilon} - \frac{4}{(\lambda-\varepsilon)^2} R_{\mathcal{E}} T(\lambda)^{-1} R_{\mathcal{E}}^T \\ &= \frac{2}{\lambda-\varepsilon} I_{m\varepsilon} - \frac{4}{(\lambda-\varepsilon)^2} \left(L_{\mathcal{E}}^{-1} R_{\mathcal{E}}^T\right)^T \left(L_{\mathcal{E}}^{-1} R_{\mathcal{E}}^T\right). \end{aligned}$$

L'obtention de la matrice $L_{\mathcal{E}}^{-1} R_r^T$ présente une complexité en $O\left(m_{\mathcal{E}} r^2\right)$ opérations. Soit $u_{i,j}$ son terme général. Les termes de la diagonale $H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1}$ sont données par

$$\forall i \in [\![1, m_{\mathcal{E}}]\!], \quad \left(H_{\mathcal{E},\mathcal{E},r}(\lambda,\varepsilon)^{-1}\right)_{i,i} = \frac{2}{\lambda-\varepsilon} - \frac{4}{(\lambda-\varepsilon)^2} \sum_{j=1}^{r} u_{j,i}^2 \ . \tag{2.25}$$

L'utilisation combinée de (2.24) et (2.25), ainsi que le fait que ν et $L_{\mathcal{E}}$ sont déjà calculés, permet d'obtenir une complexité en seulement $O\left(m_{\mathcal{E}} r^2\right)$ opérations pour calculer l'approximation de l'erreur de validation croisée leave-one-out par les spans. Lorsque r est petit par rapport à m, le calcul de cette approximation est plus rapide que celui de la borne Rayon-Marge. En effet, la complexité est inférieure à celle de la résolution du problème quadratique nécessaire pour obtenir le rayon. L'intégration de la sélection de modèle dans l'algorithme de parcours du chemin de régularisation est présentée dans la figure 2.4. Les principales différences sont le calcul de l'approximation de l'erreur de validation croisée leave-one-out ainsi que le développement du classifieur optimal (selon le critère de sélection de modèle) sur κ.

2.3.2 Approximation de l'estimateur de bootstrap .632+

L'intérêt principal de l'estimateur de bootstrap .632+ est qu'il permet d'obtenir un bon compromis entre biais et variance (voir la section 1.4.1.3). Il apparaît alors comme un critère idéal pour la sélection de modèle. Hélas, son coût de calcul est souvent prohibitif. Nous proposons donc une approximation rapide de la procédure de bootstrap en utilisant l'hypothèse que les vecteurs support ne changent pas, ce qui nous permet de travailler avec la LS-SVM associée. Nous détaillons uniquement l'obtention du bootstrap leave-one-out, puisque le bootstrap .632+ se construit à partir de celui-ci.

Notons $\mathcal{E}$ l'ensemble des indices des exemples utilisés pour entraîner la LS-SVM et $\mathcal{E}^*$ un échantillon de bootstrap obtenu à partir de $\mathcal{E}$. Par abus de langage, nous appellerons $\mathcal{E}^*$ un ensemble et nous ferons de même pour les objets construits à partir de $\mathcal{E}^*$. Nous allons montrer qu'à partir d'un apprentissage sur $\mathcal{E}$, il est possible de connaître la prédiction de la LS-SVM

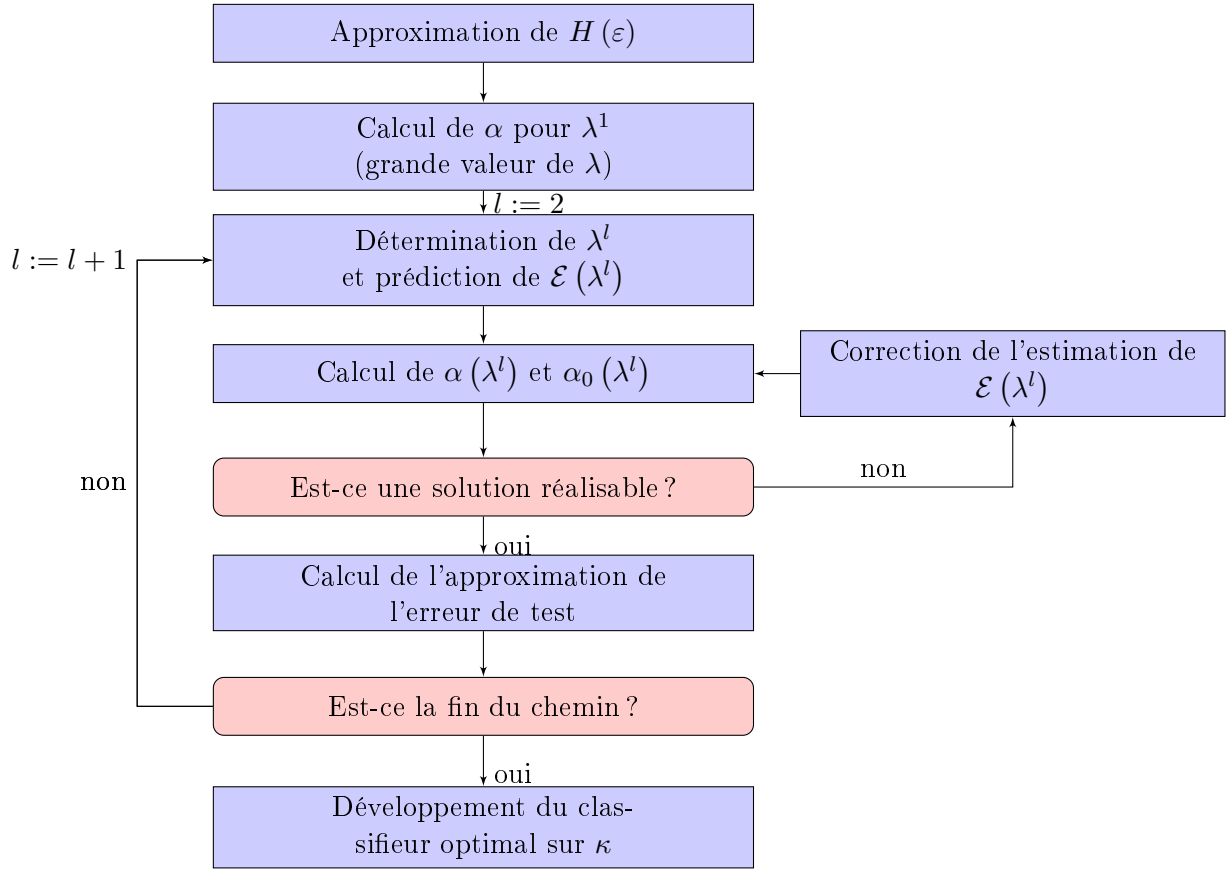


FIGURE 2.4 – Diagramme de l'algorithme de sélection de modèle par parcours du chemin de régularisation pour la ℓ_2-SVM

entraînée sur $\mathcal{E}^*$ pour les exemples d'indices $\mathcal{E} \setminus \mathcal{E}^*$. Un échantillon *bootstrap* étant obtenu par un tirage avec remise, des exemples d'indices dans $\mathcal{E}$ sont absents de $\mathcal{E}^*$, d'autres sont présents une seule fois et certains le sont plusieurs fois. Pour simplifier la démonstration, nous ne considérons pas plus de deux copies de chaque exemple. Nous notons alors :

- $\mathcal{E}_0$ l'ensemble des indices n'apparaissant pas dans l'échantillon de bootstrap,
- $\mathcal{E}_1$ l'ensemble des indices apparaissant au moins une fois dans l'échantillon de bootstrap,
- $\mathcal{E}_2$ l'ensemble des indices apparaissant deux fois dans l'échantillon de bootstrap.

L'ensemble $\mathcal{E}_1$ peut s'écrire comme la réunion des ensembles des indices apparaissant une seule fois et ceux apparaissant strictement plus d'une fois, soit $\mathcal{E}_1 = \mathcal{E}_1' \cup \mathcal{E}_1''$ avec $\mathcal{E}_1'' = \mathcal{E}_2$, et donc $\mathcal{E}^* = \mathcal{E}_1' \cup \mathcal{E}_1'' \cup \mathcal{E}_2$. Réordonnons alors les exemples de manières à avoir :

- $y^* = \left(y_{\mathcal{E}_1'}^T y_{\mathcal{E}_1''}^T y_{\mathcal{E}_2}^T\right)^T$,
- $\alpha^* = \left(\alpha_{\mathcal{E}_1'}^{*T} \alpha_{\mathcal{E}_1''}^{*T} \alpha_{\mathcal{E}_2}^{*T}\right)^T$

avec α^* le vecteur des multiplicateurs de Lagrange optimaux de la LS-SVM apprise sur $\mathcal{E}^*$. On réorganise les matrices $H_{\mathcal{E}^*,\mathcal{E}^*}(\lambda)$, $A_{\mathcal{E}^*}(\lambda)$ et $C_{\mathcal{E}^*}$ de manière similaire.

Lemme 5. *L'entraînement d'une LS-SVM sur $\mathcal{E}^*$ ou sur $\mathcal{E}_1$ fournit le même classifieur.*

L'interprétation géométrique de la LS-SVM rend ce lemme évident. Nous en présentons ici une approche analytique qui permet, de plus, de savoir comment repartir les poids entre les exemples.

Démonstration. L'apprentissage de la LS-SVM sur l'échantillon de *bootstrap* $\mathcal{E}^*$ correspond à résoudre :

$$A_{\mathcal{E}^*}(\lambda)\,\alpha^a_{\mathcal{E}^*}(\lambda) = C_{\mathcal{E}^*},$$

qui s'écrit

$$\begin{cases} \alpha^T_{\mathcal{E}'_1}\alpha_{\mathcal{E}'_1} + y^T_{\mathcal{E}''_1}\alpha_{\mathcal{E}''_1} + y^T_{\mathcal{E}_2}\alpha_{\mathcal{E}_2} = 0 \\ \alpha^*_0 y_{\mathcal{E}'_1} + \tilde{H}_{\mathcal{E}'_1,\mathcal{E}'_1}\alpha^*_{\mathcal{E}'_1} + \tilde{H}_{\mathcal{E}'_1,\mathcal{E}''_1}\alpha^*_{\mathcal{E}''_1} + \tilde{H}_{\mathcal{E}'_1,\mathcal{E}_2}\alpha^*_{\mathcal{E}_2} = 1_{\mathcal{E}'_1} \\ \alpha^*_0 y_{\mathcal{E}''_1} + \tilde{H}_{\mathcal{E}''_1,\mathcal{E}'_1}\alpha^*_{\mathcal{E}'_1} + \tilde{H}_{\mathcal{E}''_1,\mathcal{E}''_1}\alpha^*_{\mathcal{E}''_1} + \tilde{H}_{\mathcal{E}''_1,\mathcal{E}_2}\alpha^*_{\mathcal{E}_2} = 1_{\mathcal{E}''_1} \\ \alpha^*_0 y_{\mathcal{E}_2} + \tilde{H}_{\mathcal{E}_2,\mathcal{E}'_1}\alpha^*_{\mathcal{E}'_1} + \tilde{H}_{\mathcal{E}_2,\mathcal{E}''_1}\alpha^*_{\mathcal{E}''_1} + \tilde{H}_{\mathcal{E}_2,\mathcal{E}_2}\alpha^*_{\mathcal{E}_2} = 1_{\mathcal{E}_2} \end{cases}.$$

Puisque $\mathcal{E}''_1 = \mathcal{E}_2$, nous pouvons écrire ce système en fonction de $\mathcal{E}'_1$ et $\mathcal{E}''_1$ uniquement :

$$\begin{cases} \alpha^T_{\mathcal{E}'_1}\alpha_{\mathcal{E}'_1} + y^T_{\mathcal{E}''_1}\alpha_{\mathcal{E}''_1} + y^T_{\mathcal{E}_2}\alpha_{\mathcal{E}_2} = 0 \\ \alpha^*_0 y_{\mathcal{E}'_1} + \tilde{H}_{\mathcal{E}'_1,\mathcal{E}'_1}\alpha^*_{\mathcal{E}'_1} + \tilde{H}_{\mathcal{E}'_1,\mathcal{E}''_1}\left(\alpha^*_{\mathcal{E}''_1} + \alpha^*_{\mathcal{E}_2}\right) = 1_{\mathcal{E}'_1} \\ \alpha^*_0 y_{\mathcal{E}''_1} + \tilde{H}_{\mathcal{E}''_1,\mathcal{E}'_1}\alpha^*_{\mathcal{E}'_1} + \tilde{H}_{\mathcal{E}''_1,\mathcal{E}''_1}\left(\alpha^*_{\mathcal{E}''_1} + \alpha^*_{\mathcal{E}_2}\right) = 1_{\mathcal{E}''_1} \quad (2.26) \\ \alpha^*_0 y_{\mathcal{E}''_1} + \tilde{H}_{\mathcal{E}''_1,\mathcal{E}'_1}\alpha^*_{\mathcal{E}'_1} + \tilde{H}_{\mathcal{E}''_1,\mathcal{E}''_1}\left(\alpha^*_{\mathcal{E}''_1} + \alpha^*_{\mathcal{E}_2}\right) = 1_{\mathcal{E}''_1} \quad (2.27) \end{cases}$$

Les équations (2.26) et (2.27) sont identiques. De plus $\alpha^*_{\mathcal{E}''_1}$ et $\alpha^*_{\mathcal{E}_2}$ jouent un rôle symétrique dans l'apprentissage et dans l'évaluation de la fonction de décision. Lorsque le même exemple apparaît plusieurs fois dans l'échantillon de *bootstrap*, alors les multiplicateurs de Lagrange optimaux ne sont plus uniques. Le choix de prendre $\alpha^*_{\mathcal{E}_2} = 0_{m_{\mathcal{E}_2}}$ prouve que l'apprentissage sur $\mathcal{E}^*$ est équivalent à l'apprentissage sur $\mathcal{E}_1$. □

Rappelons que l'apprentissage de la LS-SVM sur $\mathcal{E}$ s'effectue en résolvant le système suivant

$$\begin{pmatrix} A_{\mathcal{E}_1,\mathcal{E}_1} & A_{\mathcal{E}_1,\mathcal{E}_0} \\ A_{\mathcal{E}_0,\mathcal{E}_1} & A_{\mathcal{E}_0,\mathcal{E}_0} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_{\mathcal{E}_1} \\ \alpha_{\mathcal{E}_0} \end{pmatrix} = \begin{pmatrix} 0 \\ 1_{m_{\mathcal{E}_1}} \\ 1_{m_{\mathcal{E}_0}} \end{pmatrix} \quad (2.28)$$

avec

$$A_{\mathcal{E}_1,\mathcal{E}_1} = \begin{pmatrix} 0 & y^T_{\mathcal{E}_1} \\ y_{\mathcal{E}_1} & \tilde{H}_{\mathcal{E}_1,\mathcal{E}_1}(\lambda) \end{pmatrix},$$

$$A_{\mathcal{E}_1,\mathcal{E}_0} = A^T_{\mathcal{E}_0,\mathcal{E}_1} = \tilde{H}_{\mathcal{E}_1,\mathcal{E}_0}(\lambda)$$

et

$$A_{\mathcal{E}_0,\mathcal{E}_0} = \tilde{H}_{\mathcal{E}_0,\mathcal{E}_0}(\lambda).$$

Comme nous l'avons vu avec le lemme précédent, l'apprentissage d'une LS-SVM sur l'échantillon *boostrap* s'effectue en résolvant le système linéaire :

$$A_{\mathcal{E}_1,\mathcal{E}_1}\begin{pmatrix}\alpha_0^*\\ \alpha_{\mathcal{E}_1}^*\end{pmatrix} = \begin{pmatrix}0\\ 1_{m_{\mathcal{E}_1}}\end{pmatrix}.$$

Lemme 6. *Soit $\tilde{h}_{\mathcal{E}_0}$ le vecteur de $\mathbb{R}^{m_{\mathcal{E}_0}}$ dont les composantes sont les valeurs de la fonction calculée par la LS-SVM (apprise sur l'échantillon de* bootstrap *$\mathcal{E}^*$) pour les points d'indices dans $\mathcal{E}_0$. Nous avons alors*

$$y_{\mathcal{E}_0} \odot \tilde{h}_{\mathcal{E}_0} = 1_{m_{\mathcal{E}_0}} - \left(\left(A_{\mathcal{E},\mathcal{E}}(\lambda)^{-1}\right)_{\mathcal{E}_0,\mathcal{E}_0}\right)^{-1} \alpha_{\mathcal{E}_0}(\lambda).$$

avec $\odot$ le produit d'Hadamard.

Démonstration. La démonstration est similaire à celle donnée pour la procédure de la validation croisée leave-one-out. Dans toute la démonstration λ est omis car le calcul se fait à λ fixé. Par définition de $\tilde{h}$, nous avons :

$$y_{\mathcal{E}_0} \odot \tilde{h}_{\mathcal{E}_0} = A_{\mathcal{E}_0,\mathcal{E}_1}\begin{pmatrix}\alpha_0^*\\ \alpha_{\mathcal{E}_1}^*\end{pmatrix}.$$

Or

$$\begin{pmatrix}\alpha_0^*\\ \alpha_{\mathcal{E}_1}^*\end{pmatrix} = {A_{\mathcal{E}_1,\mathcal{E}_1}}^{-1}\begin{pmatrix}0\\ 1_{m_{\mathcal{E}_1}}\end{pmatrix}$$

ce qui nous donne :

$$y_{\mathcal{E}_0} \odot \tilde{h}_{\mathcal{E}_0} = A_{\mathcal{E}_0,\mathcal{E}_1}\ {A_{\mathcal{E}_1,\mathcal{E}_1}}^{-1}\begin{pmatrix}0\\ 1_{m_{\mathcal{E}_1}}\end{pmatrix}. \tag{2.29}$$

Les $m_{\mathcal{E}_1}+1$ premières équations de (2.28) imposent

$$\left(A_{\mathcal{E}_1,\mathcal{E}_1}A_{\mathcal{E}_1,\mathcal{E}_0}\right)\begin{pmatrix}\alpha_0\\ \alpha_{\mathcal{E}_1}\\ \alpha_{\mathcal{E}_0}\end{pmatrix} = \begin{pmatrix}0\\ 1_{m_{\mathcal{E}_1}}\end{pmatrix}$$

alors que les $m_{\mathcal{E}_0}$ dernières donnent

$$1_{m_{\mathcal{E}_0}} = A_{\mathcal{E}_0,\mathcal{E}_1}\begin{pmatrix}\alpha_0\\ \alpha_{\mathcal{E}_1}\end{pmatrix} + A_{\mathcal{E}_0,\mathcal{E}_0}\alpha_{\mathcal{E}_0}.$$

En substituant ces 2 équations dans (2.29), on obtient successivement

$$y_{\mathcal{E}_0} \odot \tilde{h}_{\mathcal{E}_0} = A_{\mathcal{E}_0,\mathcal{E}_1} \; {A_{\mathcal{E}_1,\mathcal{E}_1}}^{-1} \left(A_{\mathcal{E}_1,\mathcal{E}_1} A_{\mathcal{E}_1,\mathcal{E}_0} \right) \begin{pmatrix} \alpha_0 \\ \alpha_{\mathcal{E}_1} \\ \alpha_{\mathcal{E}_0} \end{pmatrix} \tag{2.30}$$

$$= A_{\mathcal{E}_0,\mathcal{E}_1} \begin{pmatrix} \alpha_0 \\ \alpha_{\mathcal{E}_1} \end{pmatrix} + A_{\mathcal{E}_0,\mathcal{E}_1} \; {A_{\mathcal{E}_1,\mathcal{E}_1}}^{-1} A_{\mathcal{E}_1,\mathcal{E}_0} \alpha_{\mathcal{E}_0} \tag{2.31}$$

puis

$$y_{\mathcal{E}_0} \odot \tilde{h}_{\mathcal{E}_0} = 1_{m_{\mathcal{E}_0}} - \left(A_{\mathcal{E}_0,\mathcal{E}_0} - A_{\mathcal{E}_0,\mathcal{E}_1} \; {A_{\mathcal{E}_1,\mathcal{E}_1}}^{-1} A_{\mathcal{E}_1,\mathcal{E}_0} \right) \alpha_{\mathcal{E}_0} \tag{2.32}$$

$$= 1_{m_{\mathcal{E}_0}} - \left(\left(A^{-1}_{\mathcal{E},\mathcal{E}} \right)_{\mathcal{E}_0,\mathcal{E}_0} \right)^{-1} \alpha_{\mathcal{E}_0}. \tag{2.33}$$

La dernière ligne est obtenue par la formule d'inversion par bloc, et elle conclut la démonstration. □

D'après [41], $\hat{Err}^{(1)}$ est presque égal à celle donnée dans [42], soit :

$$\frac{\sum_{p=1}^{m} \sum_{b=1}^{B} \mathbb{1}_{\{Z_p \notin D^{*b} \wedge Y_p \neq f_{D^{*b}}(X_p)\}}}{\sum_{p=1}^{m} \sum_{b=1}^{B} \mathbb{1}_{\{Z_p \notin D^{*b}\}}}.$$

Nous utiliserons cette formule afin de garder un dénominateur similaire aux formules obtenues pour les procédures de validation croisée.

Théorème 6. *Si les vecteurs support ne changent pas pendant la procédure de bootstrap, alors l'erreur de bootstrap leave-one-out sur les échantillons est égale à*

$$\hat{Err}^{(1)}_{sv} = \frac{\sum_b 1^T_{m_{\mathcal{E}_0^b}} \Psi \left(-y_{\mathcal{E}_0^b} \odot \tilde{h}_{\mathcal{E}_0^b} \right)}{\sum_b m_{\mathcal{E}_0^b}}.$$

avec

- $\mathcal{E}_0^b$ l'ensemble des indices absents du b-ième échantillon de bootstrap,
- $m_{\mathcal{E}_0^b}$ le cardinal de $\mathcal{E}_0^b$,
- $\Psi \left(-y_{\mathcal{E}_0^b} \odot \tilde{h}_{\mathcal{E}_0^b} \right)$ le vecteur de $\mathbb{R}^{m_{\mathcal{E}_0^b}}$ contenant autant de 1 que d'exemples de $\mathcal{E}_0^b$ mal classés par la LS-SVM apprise sur le b-ieme échantillon de bootstrap.

Cette approximation de l'erreur de bootstrap nécessite une inversion d'une matrice de taille $m_{\mathcal{E}} + 1$ puis B résolutions de systèmes d'une taille approximativement égale à $0.368 m_{\mathcal{E}}$ au lieu de $m_{\mathcal{E}}$. Ceci correspond au gain d'un coefficient 20 par rapport à une approche naïve.

2.3.3 Approximation de l'erreur de validation croisée à V pas

A partir de l'approximation de l'erreur de boostrap leave-one-out, il est possible de dériver immédiatement une approximation de l'erreur de validation croisée à V pas. En appliquant la même notation que pour le bootstrap, nous définissons pour chaque pas :

- $\mathcal{E}_0$ l'ensemble des indices des exemples non retenus pour l'apprentissage du pas,
- $\mathcal{E}_1$ celui servant à entraîner la SVM pour ce pas.

De manière identique nous avons :

Lemme 7. *Soit $\tilde{h}_{\mathcal{E}_0}$ le vecteur de $\mathbb{R}^{m_{\mathcal{E}_0}}$ dont les composantes sont les valeurs de la fonction calculée par la LS-SVM (entraînée sur $\mathcal{E}_1$) pour les points d'indice dans $\mathcal{E}_0$. Nous avons alors*

$$y_{\mathcal{E}_0} \odot \tilde{h}_{\mathcal{E}_0} = 1_{m_{\mathcal{E}_0}} - \left(\left(A^{-1}_{\mathcal{E},\mathcal{E}} (\lambda) \right)_{\mathcal{E}_0,\mathcal{E}_0} \right)^{-1} \alpha_{\mathcal{E}_0} (\lambda) .$$

A partir de ce lemme, de la même manière que pour le bootstrap, nous pouvons énoncer le théorème suivant :

Théorème 7. *Si les vecteurs support ne changent pas pendant la procédure de validation croisée à V pas, alors l'erreur est égale à*

$$\hat{Err}^{(cv,V/m)}_{sv} = \frac{1}{m} \sum_{n=1}^{V} 1^T_{m_{\mathcal{E}^n_0}} \Psi \left(-y_{\mathcal{E}^n_0} \odot \tilde{h}_{\mathcal{E}^n_0} \right)$$

avec

- $\mathcal{E}^n_0$ l'ensemble des indices absents du n-ieme pas de la procédure de validation croisée à V pas,
- $m_{\mathcal{E}^n_0}$ le cardinal de $\mathcal{E}^n_0$,
- $\Psi \left(-y_{\mathcal{E}^n_0} \odot \tilde{h}_{\mathcal{E}^n_0} \right)$ le vecteur de $\mathbb{R}^{m_{\mathcal{E}^n_0}}$ contenant autant de 1 que d'exemples de $\mathcal{E}^n_0$ mal classés par la LS-SVM entraînée sur le n-ieme pas de validation croisée.

La complexité de ce calcul, sans tenir compte des optimisations possibles pour les matrices de rang faible, est en $O\left(m^3 + V\left(\frac{m}{V}\right)^3\right)$: le premier terme pour l'inversion de $A_{\mathcal{E},\mathcal{E}}$ et le second pour les V inversions de ses sous-matrices. La validation croisée leave-one-out est celle associée à l'approximation ayant la complexité la moins élevée.

2.4 Résultats expérimentaux

Nous présentons à présent les résultats de plusieurs simulations numériques. Tout d'abord, une comparaison de la solution obtenue par notre algorithme de suivi de chemin et un algorithme de l'état de l'art est proposées. Nous illustrons ensuite le bon comportement de l'approximation de l'erreur de validation croisée leave-one-out sur un exemple. L'expérience centrale est la comparaison de notre procédure de sélection de modèle avec une autre méthode de l'état de l'art sur

plusieurs jeux de données. Nous concluons l'étude expérimentale par une comparaison des temps de calcul pour la procédure de sélection de modèle par rapport à un algorithme d'apprentissage dédié aux grands jeux de données. Toutefois, nous ne présentons pas de résultat dédié aux alternatives à l'approximation de l'erreur de test que sont l'approximation de l'erreur de bootstrap et de validation croisée à V pas. En effet, nous n'avons pas remarqué de différence significative dans la qualité du modèle sélectionné lors de l'utilisation d'une estimation de grande variance ou non. Une explication possible de cette constatation est que le nombre d'hyperparamètres dont nous recherchons la valeur est très réduit, c'est-à-dire *un*, et que, dans ces conditions, le sur-apprentissage est peu présent.

2.4.1 Conditions expérimentales

Toutes les approximations de $H(\epsilon)$ sont produites à l'aide d'une méthode de Nyström. Lorsque l'ensemble d'apprentissage est plus petit que 4000 exemples, la méthode de Nyström à échantillonage pondéré par la densité est utilisée et dans le cas contraire, un échantillonage uniforme est appliqué. Toutes les données sont standardisées.

Le tableau 2.1 présente les caractéristiques des jeux de données utilisés. La majeure partie vient de la base de données de Rätsch (`http://www.raetschlab.org/Members/raetsch/benchmark`). Le jeu de donnée *Spam* vient de l'*UCI repository* [47] alors que *IJCNN1* et *A9A* sont tirés de la page de Lin (`http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`). Le jeu de données synthétiques sera présenté un peu plus loin.

Jeu de données	# descripteurs	# apprentissage	# test	# réalisations
banana	2	400	4900	100
breast cancer	9	200	77	100
diabetis	8	468	300	100
flaresolar	9	666	400	100
german	20	700	300	100
heart	13	170	100	100
image	18	1300	1010	20
ringnorm	20	400	7000	20
splice	60	1000	2175	20
tyroid	5	140	75	100
titanic	3	150	2051	100
twonorm	20	400	7000	100
waveform	21	400	4600	100
Spam	57	3601	1000	30
IJCNN1	22	49990	91701	1
A9A	123	32561	16281	1

TABLE 2.1 – Caractéristiques des jeux de données utilisés pour l'évaluation des performances du parcours du chemin de régularisation de la ℓ_2-SVM. Les quatre colonnes correspondent respectivement au nombre de descripteurs, au nombre d'exemples d'apprentissage, au nombre d'exemple de test et au nombre de réalisations.

Sauf spécifié autrement, un noyau "RBF" de largeur de bande σ est utilisé. Afin d'éviter les

instabilités numériques, ε est fixé égal à 10^{-8}. Le coefficient de régularisation balaie une plage allant de 10^{-6} à 10^{7}. L'approximation de l'erreur de validation croisée leave-one-out est calculée pour chaque valeur de λ^l. Tous les temps de calcul annoncés incluent l'approximation de $H(\varepsilon)$, le parcours du chemin, le calcul de l'approximation de l'erreur de test et le développement du classifieur optimal (lorsque nécessaire). Notre algorithme est entièrement programmé en MATLAB (R2010B).

2.4.2 Optimalité de la solution obtenue par le parcours du chemin de régularisation

Cette section illustre le fait qu'une approximation de rang faible peut permettre d'obtenir les mêmes résultats qu'une approche classique. Un algorithme de l'état de l'art, la version ℓ_2 de libsvm (que nous notons ℓ_2-libsvm), est utilisé comme référence sur trois jeux de données. Puisque nous souhaitons obtenir une bonne approximation, les paramètres suivants sont utilisés :

- le seuil sur les valeurs propres est fixé à 10^{-6},
- 80% des exemples de l'ensemble d'apprentissage sont utilisés comme *landmarks* pour construire l'approximation.

L'utilisation d'une telle fraction de l'ensemble d'apprentissage dans l'approximation nous assure une bonne qualité dans l'estimation des valeurs propres et des vecteurs propres pendant la mise en œuvre de la méthode de Nyström.

Jeu de données	Noyau (σ)	Rang de $H_r(\varepsilon)$	Temps moyen en secondes (# d'apprentissages) Chemin	ℓ_2-libsvm	Rapport des fonct. obj
banana	Linéaire (X)	2	0.5996 (44)	1.6229 (10)	0.99988
	Gaussien (0.2)	317	1.5592 (49)	0.8799 (10)	0.98236
	Gaussien (0.6)	147	0.95875 (45)	1.455 (10)	0.99993
	Gaussien (1)	81	0.84712 (46)	16.0839 (10)	0.99945
	Gaussien (1.4)	56	0.79836 (45)	18.8339 (10)	1.0001
	Gaussien (1.8)	44	0.8895 (44)	13.0951 (10)	0.9991
twonorm	Linéaire (X)	20	0.69215 (50)	0.99338 (10)	0.99963
	Gaussien (10)	319	1.0794 (49)	1.1886 (10)	0.99304
	Gaussien (15)	319	1.0554 (49)	1.1857 (10)	0.99361
	Gaussien (20)	319	1.0842 (49)	1.1932 (10)	0.99231
	Gaussien (25)	319	1.0724 (49)	1.1952 (10)	0.99305
	Gaussien (30)	289	0.99932 (49)	1.2005 (10)	0.99553
	Gaussien (35)	244	0.90694 (49)	1.2022 (10)	0.99558
	Gaussien (40)	231	0.90001 (49)	1.2066 (10)	0.99672
image	Linéaire (X)	14	2.267 (44)	33.7736 (10)	1.0001
	Gaussien (2)	905	9.3627 (47)	5.4173 (10)	0.98943
	Gaussien (3)	873	7.8396 (46)	5.2021 (10)	0.99386
	Gaussien (4)	778	7.7303 (45)	5.5274 (10)	0.99759
	Gaussien (5)	683	6.0358 (45)	6.3224 (10)	0.99899
	Gaussien (6)	598	5.3664 (41)	5.6593 (10)	0.99977
	Gaussien (7)	527	5.3114 (41)	5.7693 (10)	0.99987

TABLE 2.2 – Etude de la précision du parcours du chemin de régularisation de la ℓ_2-SVM par comparaison du rapport entre les valeurs prises par la fonction objectif pour les multiplicateurs de Lagrange obtenus avec la ℓ_2-libsvm et par notre algorithme de parcours du chemin. Pour notre algorithme, le nombre entre parenthèse est le nombre de changements effectués pour la partition de l'ensemble d'apprentissage. Pour ℓ_2-libsvm, le temps correspond à dix fois le temps moyen nécessaire pour un apprentissage (la moyenne est effectuée sur toutes les valeurs de λ considérées).

Le tableau 2.2 illustre le fait que les solutions obtenues par le parcours du chemin de régularisation et par ℓ_2-libsvm sont similaires, pour ne pas dire identiques. Il est important de mentionner que ℓ_2-libsvm a des difficultés à converger lorsque la largeur de bande du noyau "RBF" est grande, ce qui explique pourquoi les temps d'apprentissage sont si élevés. Pour ℓ_2-libsvm, nous faisons figurer dix fois le temps d'apprentissage moyen en estimant que lors de l'utilisation d'une grille, le praticien entraîne la machine au moins pour 10 valeurs de λ (Nous ne considérons pas de validation croisée).

2.4.3 Comparaison des critères de sélection de modèle

Le comportement relatif des bornes sur l'erreur de validation croisée leave-one-out (ou des approximations de cette erreur) a déjà été considérablement étudié dans la littérature (par exemple dans [27, 28]). Les deux principales conclusions qui en ont été tirées sont qu'aussi bien le minimum de la borne Rayon-Marge que celui de l'approximation de l'erreur de validation croisée leave-one-out sont adaptés à la sélection de modèle et que cette approximation est un bon estimateur de l'erreur en généralisation. La figure 2.5 illustre ces propriétés sur le jeu de données

banana.

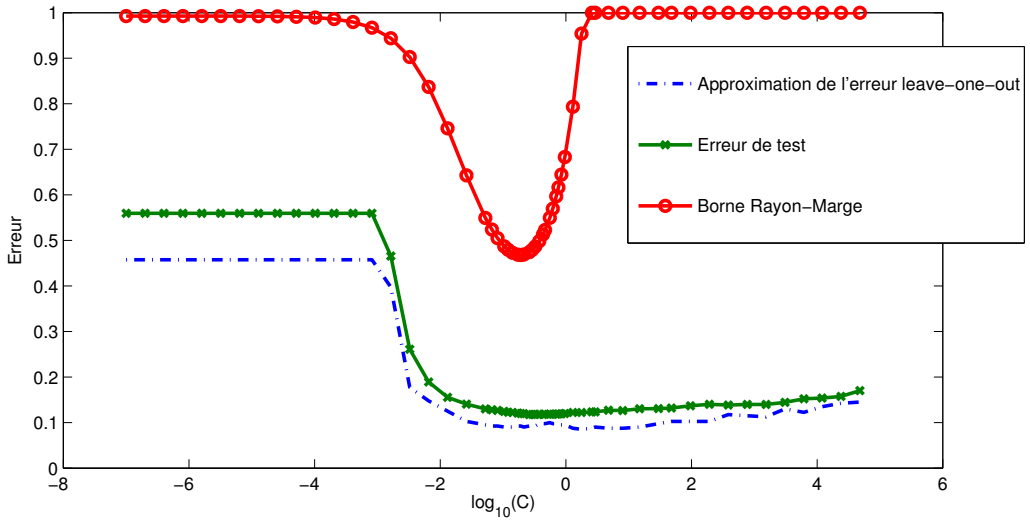


FIGURE 2.5 – Evolution de la borne Rayon-Marge (tronquée à 1), de l'approximation de l'erreur de validation croisée leave-one-out par les spans et de l'erreur de test le long du chemin de régularisation de la ℓ_2-SVM

2.4.4 Pertinence de la sélection de modèle par parcours du chemin de régularisation

Cette section illustre l'utilisation d'une approximation de faible rang de la matrice $H(\varepsilon)$ pour parcourir le chemin de régularisation dans une optique de sélection de modèle. L'algorithme de référence pour la sélection de modèle pour les ℓ_2-SVM, introduit dans [27], consiste en une descente en gradient sur une version dérivable de l'approximation de l'erreur de validation croisée leave-one-out. Voici les détails de la procédure de comparaison :

- Pour chaque réalisation du jeu de données, le classifieur optimal obtenu par l'algorithme sur l'ensemble d'apprentissage est testé sur le jeu de test correspondant. La valeur moyenne et l'écart type de cette erreur de test sont donnés dans les tableaux suivants.
- le t test de Student (décrit par exemple dans la section 3.3 of [38]) est effectué sur cette fréquence d'erreur. Si la différence est significative, le meilleur des résultats est écrit en gras.

La base de données de Rätsch La base de données de Rätsch a été largement utilisée comme benchmark pour la classification bi-classe (voir par exemple [87, 39, 72, 21]). Dans cette expérience, en raison du faible nombre d'exemples, nous utilisons une décomposition de Nyström pondérée par la densité en utilisant 60% de l'ensemble d'apprentissage comme *landmarks* et nous ne conservons que les valeurs propres supérieures à 10^{-4}.

Jeu de données	Erreur de test en % Moyenne (écart type)		Temps en secondes	
	Chemin	Gradient	Chemin	Gradient
banana	11.07 (0.88)	**10.85** (0.73)	0.6319	**0.5612**
breast cancer	27.10 (4.67)	26.64 (4.56)	0.4613	**0.2551**
diabetis	24.04 (1.94)	**23.81** (1.94)	**0.8224**	1.2781
flaresolar	**34.49** (1.87)	34.84 (1.82)	**0.4011**	2.4215
german	23.77 (2.13)	23.65 (2.03)	**1.9661**	2.8821
heart	16.83 (3.47)	16.71 (3.11)	0.4741	**0.1586**
image	**03.32** (0.72)	03.90 (0.69)	**6.5991**	8.1389
ringnorm	01.69 (0.40)	**01.61** (0.15)	0.9900	**0.4171**
splice	11.87 (0.72)	**11.16** (0.70)	**4.0894**	5.8144
tyroid	07.08 (3.69)	07.01 (3.66)	**0.4575**	0.0728
titanic	23.05 (1.11)	**22.59** (0.88)	**0.3804**	0.1292
twonorm	02.67 (0.34)	02.69 (0.18)	0.9998	**0.4748**
waveform	10.09 (0.56)	**09.90** (0.39)	1.0605	**0.7644**

TABLE 2.3 – Performances de l'algorithme de parcours du chemin sur la base de données de Rätsch avec pour valeur de σ celle obtenue par l'algorithme de Chapelle

Jeu de données	Erreur de test en % Moyenne (écart type)		Temps en secondes	
	Chemin	Gradient	Chemin	Gradient
banana	11.24 (0.95)	**10.85** (0.73)	0.7149	**0.7017**
breastcancer	27.35 (4.22)	**26.64** (4.56)	0.5152	**0.3171**
diabetis	24.05 (2.03)	**23.81** (1.94)	**1.1452**	1.6742
flaresolar	**34.40** (1.99)	34.84 (1.82)	**0.4521**	3.1518
german	23.76 (2.19)	23.65 (2.03)	**2.2267**	3.3662
heart	17.30 (3.51)	**16.71** (3.11)	0.5602	**0.2204**
image	**03.24** (0.74)	03.90 (0.69)	**2.8286**	9.1764
ringnorm	02.08 (0.37)	**01.61** (0.15)	0.9758	**0.4092**
splice	11.67 (0.74)	**11.16** (0.70)	**4.7143**	6.9180
thyroid	06.71 (3.28)	07.01 (3.66)	0.5089	**0.1000**
titanic	23.45 (4.28)	**22.59** (0.88)	0.3827	**0.1586**
twonorm	02.68 (0.34)	02.69 (0.18)	0.9846	**0.4683**
waveform	10.29 (0.75)	**09.90** (0.39)	0.9157	**0.6260**

TABLE 2.4 – Performances de l'algorithme de parcours du chemin sur la base de données de Rätsch avec la valeur de σ choisie par l'heuristique de [113]

Le tableau 2.3 présente les erreurs de test obtenues avec une largeur de bande choisie par l'algorithme de descente en gradient de Chapelle alors que le tableau 2.4 correspond à une largeur de bande obtenue par l'heuristique proposée dans [113]. Cette méthode fixe σ égal à la racine carrée de la distance moyenne de chaque exemple au centre de masse de l'ensemble d'apprentissage. Même avec une heuristique très simple, les performances de notre algorithme sont similaires à celles de l'algorithme de Chapelle. Les taux de reconnaissance sont du même

ordre que ceux de la littérature (voir par exemple [28]). Ces deux tableaux illustrent que les deux méthodes permettent d'obtenir de bonnes performances en sélection de modèle sans faire d'hypothèse sur le rang nécessaire pour approcher $H(\varepsilon)$. Les jeux de données de la base de Rätsch ont un nombre de descripteurs relativement élevé par rapport au nombre d'exemples. Il en résulte que le rang nécessaire pour obtenir de bonnes performances n'est pas très petit devant le nombre d'exemples, ce qui explique les temps de calcul similaires.

Jeu de données	Erreur de test en % Moyenne (écart type)	
	Chemin	Gradient
banana	11.22 (0.85)	11.07 (0.88)
breast cancer	26.53 (4.72)	27.10 (4.67)
diabetis	26.84 (2.25)	24.04 (1.94)
flare solar	35.51 (1.85)	34.49 (1.87)
german	23.8 (2.29)	23.77 (2.13)
heart	17.42 (3.43)	16.83 (3.47)
image	3.35 (0.65)	3.32 (0.72)
ringnorm	1.92 (0.37)	1.69 (0.40)
splice	12.18 (0.77)	11.87 (0.72)
thyroid	6.73 (3.08)	7.08 (3.69)
titanic	28.22 (4.56)	23.05 (1.11)
twonorm	2.66 (0.31)	2.67 (0.34)
waveform	10.26 (0.61)	0 (0.59)

TABLE 2.5 – Performances de l'algorithme de parcours du chemin sur la base de données de Rätsch pour un noyau gaussien dont la valeur de σ celle obtenue par l'algorithme de Chapelle et un noyau "thinplate"

Le tableau 2.5 présente le résultat de l'application de l'algorithme de parcours du chemin de régularisation au noyau "thinplate" [107]. Une propriété intéressante de l'algorithme présenté est que l'approximation de faible rang utilisée permet de travailler avec un noyau de type positif même si le noyau d'origine ne l'est pas. Ici le noyau "thinplate" n'est que conditionnellement défini positif. L'intérêt de ce noyau est de ne pas être avoir de paramètre et, grâce à son invariance par dilatation, de fournir des performances très intéressantes. L'application de cet algorithme à ce type de noyau soulève des questions importantes que abordons dans les perspectives.

Le jeu de données *Spam* Ce jeu de données est beaucoup plus grand que ceux de la base de données de Rätsch. Cette augmentation du nombre d'exemples permet de mettre en évidence, sur des données réelles, la manière dont l'algorithme de parcours du chemin se comporte en fonction du nombre d'exemples. Puisque ce jeu de données n'est pas divisé en un jeu d'apprentissage et un jeu de test, 1000 exemples, pour chaque réalisation, sont choisis pour constituer le jeu de test, le reste servant pour l'apprentissage.

Le tableau 2.6 présente les résultats de la sélection de modèle par chemin et par descente en gradient. Le choix de la valeur supérieur du rang de la matrice de Gram (600) est obtenu en

Critère	Erreur de test en % Moyenne (écart type)		Temps en secondes	
	Chemin	Gradient	Chemin	Gradient
défaut	**6.22** (0.87)	6.58 (0.84)	**27.8510**	97.5712
rang=600	6.64 (0.97)	6.56 (0.76)	**15.9084**	102.9281

TABLE 2.6 – Performance de l'algorithme de parcours du chemin sur le jeu de données *Spam* avec σ obtenu par l'heuristique de [113]. La ligne "défaut" correspond à l'utilisation de 60% de l'ensemble d'apprentissage comme proportion de *landmarks* alors que "rang=600" utilise 600 *landmarks*. Cette dernière condition impose que le rang de la matrice de Gram est au plus égal à 600. Le seuil des valeurs propres est fixé à 10^{-3} dans les deux cas.

utilisant la classique règle du *coude* (voir par exemple dans l'Analyse en Composante Principale). L'utilisation de cette règle simple permet quasiment de diviser le temps d'apprentissage par deux.

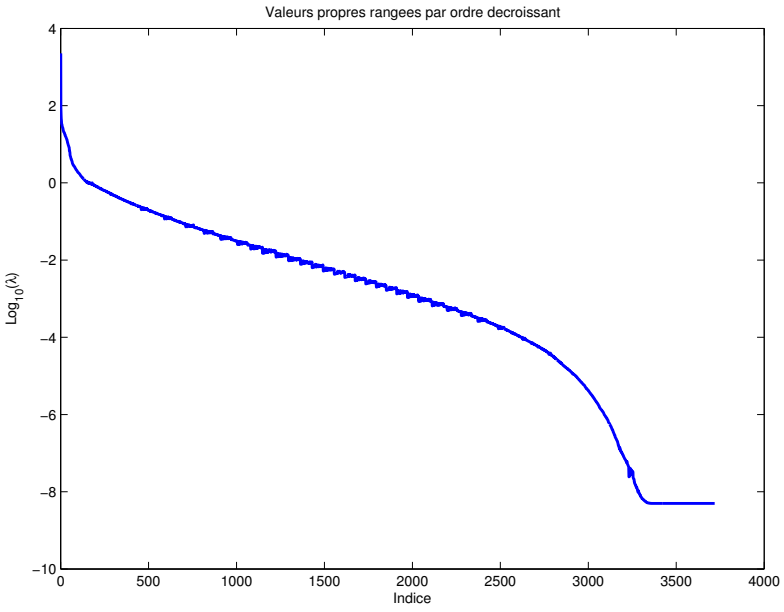


FIGURE 2.6 – Spectre de la matrice de Gram calculé sur le jeu de données *Spam*

Données synthétiques Afin d'évaluer l'efficacité de l'algorithme dans un cas favorable, nous avons créé un jeu de données synthétique. Les exemples de la catégorie positive sont tirés d'une distribution gaussienne alors que ceux de la catégorie négative le sont d'une mixture de gaussiennes, les catégories étant équiprobables :

$$p(x|y=+1) = p_{\mathcal{N}}\left(\mu^+, \Sigma^+, x\right)$$
$$p(x|y=-1) = \frac{1}{2}p_{\mathcal{N}}\left(\mu_1^-, \Sigma_1^-, x\right) + \frac{1}{2}p_{\mathcal{N}}\left(\mu_2^-, \Sigma_2^-, x\right)$$

avec

$$p_{\mathcal{N}}(\mu, \Sigma, x) = \frac{1}{(2\pi)\, det\,(\Sigma)^{1/2}} \exp^{-\frac{1}{2}(x-\mu)^T(\Sigma)^{-1}(x-\mu)},$$

$$\mu^+ = \begin{pmatrix}1\\1\end{pmatrix}, \Sigma^+ = \begin{pmatrix}8 & -6\\-6 & 8\end{pmatrix}, \mu_1^- = \begin{pmatrix}0\\0\end{pmatrix}, \Sigma_1^- = \begin{pmatrix}1/4 & 0\\0 & 1/4\end{pmatrix}, \mu_2^- = \begin{pmatrix}5\\-2\end{pmatrix}, \Sigma_2^- = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}.$$

Une estimation de l'erreur de Bayes par une méthode de Monte Carlo est de 12.11%. Pour chaque valeur de m, dix réalisations de l'ensemble d'apprentissage ont été produites (voir le tableau 2.7). L'évaluation de l'erreur en généralisation, pour chaque classifieur, se fait sur un grand jeu de test (30000 exemples).

m	Erreur de test en % Moyenne (écart type)		Temps en secondes		Rang de $H_r(\varepsilon)$
	Chemin	Gradient	Chemin	Gradient	
1000	14.22 (0.28)	14.05 (0.25)	5.68	9.21	127.5
2000	13.97 (0.16)	**13.93** (0.13)	12.84	57.55	165.8
3000	13.86 (0.17)	13.85 (0.19)	20.87	101.1	181.4
4000	13.85 (0.33)	13.83 (0.33)	28.41	262.56	155.2
5000	13.84 (0.19)	13.82 (0.2)	37.98	446.56	154.7

TABLE 2.7 – Performances de l'algorithme de parcours du chemin sur les données artificielles avec pour valeur de σ celle obtenue par l'algorithme de Chapelle

Le tableau 2.7 illustre le fait que les deux méthodes considérées obtiennent des performances similaires en termes de taux de reconnaissance. A l'inverse, les temps de calcul sont clairement en faveur de la méthode parcourant le chemin. Ce gain de temps est clairement dû à l'utilisation de l'approximation de rang faible (voir le rapport rang/nombre d'exemples). Il faut toutefois remarquer que l'algorithme de Chapelle n'est pas conçu pour les grands jeux de données. En conséquence, dans la section suivante, nous comparons notre méthode à un algorithme dédié à ce type de problème.

2.4.5 Comparaison des performances et des temps de calcul sur de grands jeux de données

Il est évident que l'approche proposée pour les approximations de rang faible est particulièrement efficace lorsque la matrice de Gram est effectivement de rang faible. Il se trouve que les polynômes de degré faible peuvent induire cette propriété [1]. A notre connaissance, il n'y a pas de procédure de sélection de modèle dédiée à cette situation. Ainsi, pour effectuer cette tâche nous utilisons une procédure de validation croisée à 5 pas mise en œuvre sur une grille. Les algorithmes qui seront utilisés sur cette grille sont des versions modifiées de ℓ_2-liblinear et de ℓ_2-libsvm qui sont introduites dans [25]. Ces versions sont le résultat d'une spécialisation afin

1. En effet, un noyau polynômial envoie dans un espace de dimension finie qui peut être de dimension plus faible que le nombre d'exemples.

d'être dédiées aux noyaux polynômiaux de degré faible. De plus liblinear est connu pour être un des algorithmes les plus rapides pour entraîner une SVM linéaire.

Comme dans [25] [1], nous choisissons la grille suivante : $C = \lambda^{-1} \in \left\{2^{-3}, 2^{-1}, \ldots, 2^{7}, 2^{9}\right\}$.

		IJCNN1	A9A
	Noyau	$(32\langle x_1, x_2\rangle + 1)^2$	$(0.032\langle x_1, x_2\rangle + 1)^2$
ℓ_2-liblinear/ℓ_2-libsvm	Critère d'arrêt	10^{-6}	10^{-6}
	C	0.125	8
Chemin de régularisation	Nombre de *landmarks*	500	1500
	Seuil sur les valeurs propres	10^{-6}	10^{-6}

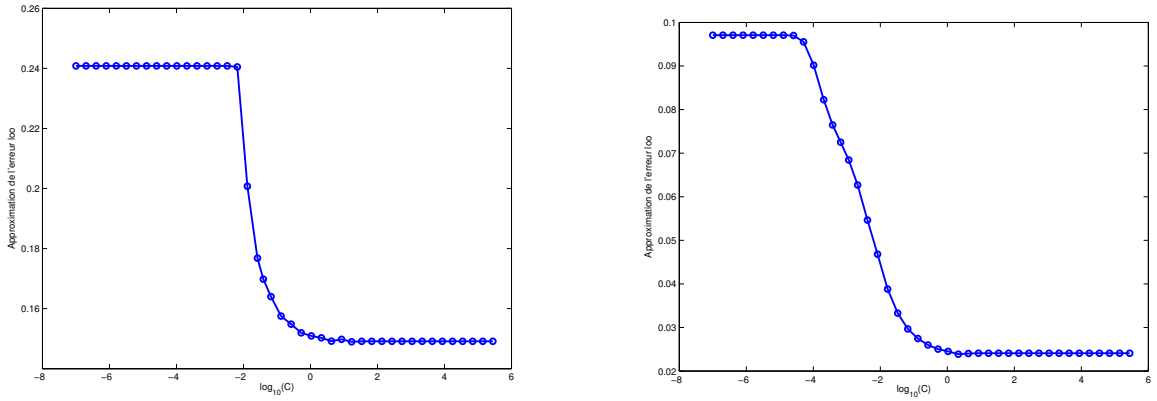
TABLE 2.8 – Paramétrages des algorithmes dédiés aux grands jeux de données. L'échantillonnage utilisé dans l'approximation de $H(\epsilon)$ est uniforme

Jeu de données	Algorithme	Temps en secondes	Temps de test (s) en secondes	Erreur de test (%) en %	Rang	C
IJCNN1	Chemin	**30.2s**	1.9s	2.44%	208	2
	ℓ_2-liblinear	57.8s	0.4s	2.46%	-	0.125
	ℓ_2-libsvm	$>$ 24 hours	27s	2.46%	-	0.125
A9A	Chemin	**179.9s**	1.9s	14.8%	1178	16
	ℓ_2-liblinear	459.7s	0.05s	14.7%	-	8
	ℓ_2-libsvm	46803s	31s	15.2%	-	8

TABLE 2.9 – Comparaison des temps d'apprentissage et de test pour *IJCNN1* et pour *A9A* avec un noyau polynômial de degré 2

Le paramétrage de l'algorithme est présenté dans le tableau 2.8 alors que les résultats de la comparaison sont dans le tableau 2.9. Ce dernier tableau met en évidence la diminution significative du temps d'apprentissage. De plus, le développement du classifieur sur r exemples permet de limiter le temps de test. On remarque que les taux de reconnaissance obtenus par les différents algorithmes sont les mêmes alors que les valeurs des coefficients de régularisation sont très différent. La figure 2.7 illustre le comportement de l'approximation de l'erreur de validation croisée leave-one-out par les spans en fonction du logarithme décimal de C. On remarque principalement un plateau à partir d'une certaine valeur, ce qui explique pourquoi les classifieurs présentent la même erreur en test pour des valeurs de C différentes.

1. Cette grille n'apparait pas dans la version finale de l'article.



(a) (b)

FIGURE 2.7 – Evolution de l'approximation de l'erreur de validation croisée leave-one-out pour *A9A* (a) et pour *IJCNN1* (b)

Ce comportement, que nous avons observé pour les espaces de représentation de dimension finie, soulevé une question importante : comment choisir le minimum sur un plateau ? D'autant plus que la présence de *bruit* peut rendre la notion de plateau plus difficile à déceler. Mis à part le minimum strict, il pourrait être intéressant de considérer d'autres méthodes de sélection telles que :

- la solution la plus régularisée de la partie *stable*,
- la solution centrale de la partie *stable* du chemin ,
- ou encore la solution la moins régularisée de la partie *stable*.

Le choix d'une de ces solutions devraient se faire en tenant compte d'a priori sur le problème d'apprentissage, par exemple en utilisant les liens entre la régularisation et l'apprentissage robuste [111].

Les résultats obtenus avec les noyaux polynomiaux de faible degré sont d'autant plus intéressants que notre procédure ne se restreint pas à ces noyaux. En effet, cette procédure peut notamment être utilisée avec des noyaux dédiés.

2.5 Conclusion

Dans ce chapitre, un algorithme de parcours du chemin de régularisation permettant de choisir le coefficient de régularisation de la ℓ_2-SVM a été introduit. Nous avons commencé par détailler et analyser celui de la ℓ_1-SVM, notamment en précisant certaines démonstrations. En nous inspirant de ce résultat, nous avons proposé trois contributions originales : l'algorithme de parcours du chemin de régularisation de la ℓ_2-SVM, l'intégration de l'approximation de l'erreur de validation croisée leave-one-out dans ce parcours et le développement d'autres approximations de l'erreur de généralisation du classifieur. La combinaison du parcours avec l'évaluation d'un critère de sélection de modèle permet d'éviter les minima locaux. De plus cette méthode est

particulièrement efficace lorsque le rang de la matrice de Gram est faible devant le nombre d'exemples. En effet, la complexité de la sélection de modèle est linéaire en le nombre d'exemples lorsque le rang de la matrice de Gram ne dépend pas du nombre d'exemples. Nous avons présenté des résultats expérimentaux mettant en évidence que le gain en temps est significatif.

Chapitre 3

Sélection de modèle par parcours du chemin de régularisation pour la M-SVM^2

« Reading furnishes the mind only with materials for knowledge ; it is thinking that makes what we read ours. »
—John Locke

Ce chapitre détaille nos contributions relatives à la sélection de modèle pour la M-SVM^2. Nous commençons par présenter une nouvelle machine, la M-SVM des moindres carrés ("Least Squares" M-SVM ou LS-M-SVM) qui est à la M-SVM^2 ce que la LS-SVM est à la ℓ_2-SVM. Ce développement effectué, nous détaillons, à la manière du cas bi-classe, l'obtention d'un algorithme de parcours du chemin de régularisation en nous aidant des notations et équations développées pour la LS-M-SVM. La partie suivante est dédiée à sélection de modèle pour la M-SVM^2, en présentant notamment une nouvelle borne exacte sur l'erreur de validation croisée leave-one-out ainsi que deux approximations de cette erreur. Un algorithme combinant le chemin de régularisation ainsi que l'utilisation d'un de ces critères est proposé puis comparé à un algorithme se basant sur l'optimisation, par une méthode de descente en gradient, de la borne Rayon-Marge (voir annexe A [1]).

3.1 M-SVM des moindres carrés

Dans le chapitre 2, nous avons démontré que l'apprentissage de la ℓ_2-SVM est similaire à celui de la LS-SVM, puisque se réduisant à la résolution d'un système linéaire. Nous avons ensuite exploité cette similarité afin d'obtenir notre algorithme de parcours de chemin ainsi que l'évaluation de l'erreur de validation croisée leave-one-out. Pour traiter le cas de la M-SVM^2,

1. Cette optimisation de la borne Rayon-Marge qui est aussi une contribution originale de la thèse mais est déportée en annexe pour ne pas perturber la lecture de ce chapitre dédie au parcours du chemin.

nous procédons de la même manière. Nous commençons donc par définir une machine de type "Least Squares", la M-SVM des moindres carrés (LS-M-SVM pour "Least squares" M-SVM), puis établissons les équations utiles pour le parcours du chemin de régularisation. Finalement, nous présentons une méthode analytique de calcul de l'erreur de validation croisée leave-one-out pour la LS-M-SVM.

3.1.1 Problème d'apprentissage

Puisque la LS-SVM permet d'étudier la ℓ_2-SVM, nous commençons par étudier une LS-SVM multi-classe permettant la même démarche. Or, l'étude de la machine définie par Suykens et Vandewalle [100] n'apporte pas d'aide pour celle de la M-SVM^2. En effet, nous souhaitons que les systèmes linéaires à résoudre pour la LS-M-SVM et la M-SVM^2 possèdent la même structure, ce qui impose de choisir les mêmes contraintes de bon classement, c'est-à-dire les mêmes que celles de la LLW. Nous proposons ainsi une machine en remplaçant les contraintes inégalités de la M-SVM^2 par des contraintes égalités. En utilisant les mêmes notations que celles introduites dans le modèle générique de M-SVM (voir section 1.3.2), nous définissons :

Définition 13. *Soit $\mathcal{X}$ un ensemble non vide et $Q \in \mathbb{N} \setminus [\![0,2]\!]$. Soit κ une fonction de type positif à valeurs réelles sur $\mathcal{X}^2$. Soit $\mathbf{H}_{\kappa,Q}$ et $\mathcal{H}_{\kappa,Q}$ les deux classes de fonctions induites par κ d'après les définitions 7 et 8. Soit $P_{\mathbf{H}_{\kappa,Q}}$ l'opérateur de projection orthogonale de $\mathcal{H}_{\kappa,Q}$ sur $\mathbf{H}_{\kappa,Q}$. Pour $m \in \mathbb{N}^*$, soit $d_m = ((x_i, y_i))_{1 \leqslant i \leqslant m} \in (\mathcal{X} \times [\![1,Q]\!])^m$ et $\xi \in \mathbb{R}^{Qm}(d_m)$. Une LS-M-SVM à Q catégories munie du noyau κ et dont l'ensemble d'apprentissage est d_m est un modèle discriminant à grande marge entraîné en résolvant un problème de programmation quadratique convexe de la forme*

Problème 15 (Problème d'apprentissage de la LS-M-SVM, formulation primale)**.**

$$\min_{h,\xi} J_{LS\text{-}M\text{-}SVM}(h,\xi)$$

$$s.c. \begin{cases} \forall i \in [\![1,m]\!],\ \forall k \in [\![1,Q]\!] \setminus \{y_i\},\ \ h_k(x_i) = -\frac{1}{Q-1} + \xi_{ik} \\ \sum_{k=1}^{Q} h_k = 0 \end{cases}$$

où

$$J_{LS\text{-}M\text{-}SVM}(h,\xi) = \|P_{H_{\kappa,Q}} h\|^2_{\kappa,Q} + C\|M\xi\|_2^2$$

et $M \in \mathcal{M}_{Qm,Qm}(d_m)$ est la matrice de terme général

$$m_{ik,jl} = (1 - \delta_{y_i,k})\left(1 - \delta_{y_j,l}\right)\left(\delta_{k,l} + \frac{\sqrt{Q}-1}{Q-1}\right)\delta_{i,j}.$$

L'obtention du dual est similaire à celle du dual de la M-SVM^2 et ne sera pas présentée ici. Nous exprimons ce problème à l'aide du coefficient $C = \frac{1}{\lambda}$ afin d'économiser un changement de

variable.[1] Toutefois, nous utilisons λ par la suite.

Problème 16 (Problème d'apprentissage de la LS-M-SVM, formulation duale)**.**

$$\max_{\alpha} J_{LS\text{-}M\text{-}SVM,d}(\alpha)$$

$$s.c.\ \forall k \in [\![1, Q-1]\!],\ \sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q} - \delta_{k,l}\right)\alpha_{il} = 0$$

avec

$$J_{LS\text{-}M\text{-}SVM,d}(\alpha) = -\frac{1}{2}\alpha^T H(\lambda)\,\alpha + \frac{1}{Q-1}1_{Qm}^T\alpha,$$

où $H(\lambda)$ *est la matrice de* $\mathcal{M}_{Qm,Qm}(\mathbb{R})$ *de terme général*

$$h_{ik,jl}(\lambda) = \left(\delta_{k,l} - \frac{1}{Q}\right)(\kappa(x_i, x_j) + \lambda\delta_{i,j}).$$

Ce problème d'apprentissage est exactement celui de l'apprentissage de la M-SVM^2 (Problème 13) privé de la contrainte de positivité des multiplicateurs de Lagrange.

A présent, en s'inspirant de la démarche de la section 2.2.2, nous exprimons la solution de ce problème de programmation quadratique comme la solution d'un système linéaire. Toujours dans une optique de préparation au parcours du chemin de régularisation de la M-SVM^2, nous introduisons l'ensemble E défini de la manière suivante :

$$E = \{(i,k) \in [\![1,m]\!] \times [\![1,Q]\!] \ /\ k \neq y_i\}.$$

Son cardinal est $(Q-1)\,m$. Pour garder les notations simples, nous associons l'indice $(i-1)\,Q+k$ à chaque couple (i,k) de E. Afin d'obtenir une écriture concise du système linéaire recherché, en nous inspirant de [77], nous introduisons les notations matricielles suivantes :

- $H_{E,E}(\lambda) \in \mathcal{M}_{(Q-1)m,(Q-1)m}(\mathbb{R})$ la sous-matrice de $H(\lambda)$ constituée seulement des lignes et des colonnes d'indice dans E,
- $\alpha_E \in \mathbb{R}^{(Q-1)m}$ le sous-vecteur de α constitué seulement des composantes d'indice de E,
- $\Delta \in \mathcal{M}_{Q-1,Q}(\mathbb{R})$ la matrice de terme général $\delta_{i,j} - \delta_{i+1,j}$ pour $i \in [\![1, Q-1]\!]$ et $j \in [\![1, Q]\!]$,
- $M_\Delta \in \mathcal{M}_{Q-1,(Q-1)m}(\mathbb{R})$ la sous-matrice de $1_m^T \otimes \Delta$ constituée des colonnes d'indice dans E,
- $M_I \in \mathcal{M}_{(Q-1)m,Q}(\mathbb{R})$ la sous-matrice de $1_m \otimes I_Q$ constituée seulement des lignes d'indice dans E.

Avec ces notations, l'obtention du système linéaire se fait par application de la méthode du lagrangien (section 10.2 de [45]) sur la formulation primale du problème d'apprentissage de la LS-M-SVM écrite comme une machine à marge dure. En effet le gradient du lagrangien par rapport aux variables duales et primales correspond aux contraintes des problèmes primal et dual. Les

1. On souhaite éviter d'obtenir le λ au dénominateur comme dans le problème 11.

contraintes de bon classement de la LS-M-SVM à marge dure s'expriment sous la forme :

$$-H_{E,E}(\lambda)\,\alpha_E + M_I b = -\frac{1}{Q-1}1_{(Q-1)m}. \tag{3.1}$$

La contrainte sur les multiplicateurs de Lagrange et la contrainte de sommation à zéro des composantes du vecteur b se reformulent respectivement :

$$\forall k \in [\![1, Q-1]\!],\ \sum_{i=1}^{m} \alpha_{ik} = \sum_{i=1}^{m} \alpha_{i(k+1)}$$

$$\iff \begin{pmatrix} 1 & -1 & & 0 & \dots & 1 & -1 & & 0 \\ & \ddots & \ddots & & \dots & & \ddots & \ddots & \\ 0 & \ddots & 1 & -1 & \dots & 0 & \ddots & 1 & -1 \end{pmatrix} \alpha = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\iff M_\Delta \alpha_E = 0_{Q-1} \tag{3.2}$$

et

$$1_Q^T b = 0. \tag{3.3}$$

Combiner les systèmes (3.1) à (3.3) produit le système suivant, de dimension $(Q-1)\,m+Q$:

$$\begin{pmatrix} -H_{E,E}(\lambda) & M_I \\ 0_{(Q-1)m}^T & 1_Q^T \\ M_\Delta & 0_{Q-1,Q} \end{pmatrix} \begin{pmatrix} \alpha_E \\ b \end{pmatrix} = \begin{pmatrix} -\frac{1}{Q-1}1_{(Q-1)m} \\ 0 \\ 0_{Q-1} \end{pmatrix} \tag{3.4}$$

dont la solution est l'optimum des problèmes 15 et 16.

Théorème 8. *Entraîner la LS-M-SVM se réduit à résoudre le système linéaire (3.4).*

Ce résultat est l'extension multi-classe des propositions 5 et 6. Le système 3.4 est proche dans sa structure de celui du parcours du chemin de régularisation proposé par Lee et Cui [77]. Nous retrouvons la dépendance à λ dans le terme de gauche pour la M-SVM^2 (comme pour la ℓ_2-SVM) et non dans le terme de droite (comme pour les machines de norme 1). L'évolution des multiplicateurs de Lagrange ne sera donc pas linéaire par morceaux.

3.1.2 Calcul pratique des multiplicateurs de Lagrange

Dans cette section, nous appliquons la même approche que celle de la section 2.2.3 afin d'obtenir la solution du système (3.4). Nous proposons une méthode pour le cas général puis une méthode dédiée au cas où la matrice de Gram est de rang faible.

3.1.2.1 Calcul général

En admettant que $H_{E,E}(\lambda)$ est inversible, l'équation (3.1) se réécrit

$$\alpha_E = H_{E,E}(\lambda)^{-1} M_I b + \frac{1}{Q-1} H_{E,E}(\lambda)^{-1} 1_{(Q-1)m} \tag{3.5}$$

et donc par substitution dans (3.2), on obtient :

$$M_\Delta H_{E,E}(\lambda)^{-1} M_I b = -\frac{1}{Q-1} M_\Delta H_{E,E}(\lambda)^{-1} 1_{(Q-1)m}. \tag{3.6}$$

Cette équation étant l'analogue de l'équation (2.10) du cas bi-classe, nous prolongeons l'analogie en notant $\nu \in \mathcal{M}_{(Q-1)m,Q}(\mathbb{R})$ et $\rho \in \mathbb{R}^{(Q-1)m}$ tels que :

$$\begin{cases} H_{E,E}(\lambda)\,\nu = -M_I \\ H_{E,E}(\lambda)\,\rho = \frac{1}{Q-1} 1_{(Q-1)m} \end{cases}. \tag{3.7}$$

L'utilisation des équations (3.3) à (3.7) donne l'équivalence :

$$(3.4) \Leftrightarrow \begin{cases} b = \begin{pmatrix} 1_Q^T \\ M_\Delta \nu \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ M_\Delta \rho \end{pmatrix} \\ \alpha_E = \rho - \nu b \end{cases}$$

en supposant que la matrice $\begin{pmatrix} 1_Q^T \\ M_\Delta \nu \end{pmatrix}$ de taille Q est inversible. Comme dans le cas bi-classe, la principale difficulté consiste à résoudre les systèmes de (3.7). Puisque la résolution du système est la même que dans la section 2.2.3, la complexité est similaire. En effet, le nombre d'opérations nécessaires est composé d'un coût cubique, pour la décomposition de Cholesky (Gaxpy), et d'un coût quadratique, pour la substitution. La dimension de $H_{E,E}(\lambda)$ est $(Q-1)m$ et il y a Q équations ce qui donne une complexité en $\frac{1}{3}(Q-1)^3 m^3 + 2(Q+1)(Q-1)^2 m^2$ pour résoudre (3.7) à condition que $H_{E,E}(\lambda)$ soit de rang plein et tienne en mémoire. Ce coût en temps et en espace mémoire est beaucoup trop élevé pour être acceptable sur des jeux de données de taille importante. Toutefois, cette formulation nous permet d'adapter l'apprentissage de la LS-M-SVM au cas où la matrice de Gram est de rang faible.

3.1.2.2 Cas d'une matrice de Gram de rang faible

Lorsque la matrice de Gram est de rang faible, le calcul de ν et ρ peut se faire à coût réduit (voir la section 2.2.3). Supposons que K est de rang r, alors il existe une matrice $R \in \mathcal{M}_{m,r}(\mathbb{R})$ telle que $K = RR^T$. De manière analogue, puisque $I_Q - \frac{1}{Q} 1_Q 1_Q^T$ est de rang $Q-1$, il existe

$T \in \mathcal{M}_{Q,Q-1}(\mathbb{R})$ telle que $I_Q - \frac{1}{Q}1_Q 1_Q^T = TT^T$. [1]

Nous définissons alors

- la matrice $R_E \in \mathcal{M}_{(Q-1)m,(Q-1)r}(\mathbb{R})$ la sous-matrice de $R \otimes T$ constituée seulement des lignes d'indices dans E,
- la matrice $D_E \in \mathcal{M}_{(Q-1)m,(Q-1)m}(\mathbb{R})$ la sous-matrice de $I_m \otimes \left(I_Q - \frac{1}{Q}1_Q 1_Q^T\right)$ constituée seulement des lignes et des colonnes d'indice dans E. [2]

Ces notations permettent de formuler $H_{E,E}(\lambda)$ comme la somme d'une matrice de rang plein et d'une matrice de rang $(Q-1)r$. Les propriétés du produit de Kronecker permettent d'écrire :

$$\begin{aligned} H(\lambda) &= (K + \lambda I_m) \otimes \left(I_Q - \frac{1}{Q}1_Q 1_Q^T\right) \\ &= \left(RR^T + \lambda I_m\right) \otimes \left(TT^T\right) \\ &= RR^T \otimes TT^T + \lambda I_m \otimes \left(I_Q - \frac{1}{Q}1_Q 1_Q^T\right) \\ &= (R \otimes T)(R \otimes T)^T + \lambda I_m \otimes \left(I_Q - \frac{1}{Q}1_Q 1_Q^T\right) \end{aligned}$$

et donc

$$H_{E,E}(\lambda) = R_E R_E^T + \lambda D_E.$$

La formule de Sherman-Morrison-Woodburry donne une méthode pour calculer son inverse :

$$H_{E,E}(\lambda)^{-1} = \frac{1}{\lambda}D_E^{-1} - \frac{1}{\lambda^2}D_E^{-1}R_E\left(I_{(Q-1)r} + \frac{1}{\lambda}R_E^T D_E^{-1}R_E\right)^{-1}R_E^T D_E^{-1}. \tag{3.8}$$

Cette équation est particulièrement utile puisque la matrice $D_E^{-1}R_E$ est facile à obtenir. En effet, en exploitant le fait que D_E est diagonale par blocs, il suffit de $mr(Q-1)^2$ opérations pour la calculer. La plus grosse contribution à la complexité vient du produit de matrices $R_E^T D_E^{-1}R_E$ qui demande $O\left(mr^2(Q-1)^3\right)$ opérations. Il en résulte que l'obtention de ρ se fait en $O\left(mr(Q-1)^2 + r^3(Q-1)^3 + mr^2(Q-1)^3\right)$ opérations [3] et celle de ν coûte un peu plus cher mais réutilise une partie des calculs. Pour cette méthode, les besoins en mémoire sont de l'ordre de $O\left(mr(Q-1)^2\right)$ ce qui est raisonnable lorsque le rang est faible.

3.1.3 Calcul de l'erreur de validation croisée leave-one-out

En suivant la démarche proposée pour la LS-SVM dans la section 2.3.1, l'équation (3.4) permet d'obtenir la valeur exacte de l'erreur de validation croisée leave-one-out.

La définition de la règle de décision (1.9) impose de connaître la totalité des sorties de la

1. La matrice $I_Q - \frac{1}{Q}1_Q 1_Q^T$ ayant pour spectre 0 de multiplicité 1 et 1 de multiplicité $Q-1$, cette factorisation s'obtient en effectuant une décomposition en valeurs propres.

2. Pour E, l'expression analytique de D_E est $I_m \otimes \left(I_{Q-1} - \frac{1}{Q}1_{Q-1}1_{Q-1}^T\right)$.

3. On retrouve la composante cubique de la décomposition de Cholesky de la matrice de taille $(Q-1)r$ et les deux composantes dues aux produits de matrices de tailles $(Q-1)m$ et $(Q-1)r$.

M-SVM afin de décider de la classification. Il est donc nécessaire de connaître les Q sorties pour déterminer si la M-SVM fait une erreur. En pratique, grâce à la contrainte de sommation à zéro des fonctions composantes, il suffit de $Q-1$ sorties. Le théorème 9 permet justement de calculer toutes les sorties sauf celle d'indice $k = y_i$ de la LS-M-SVM pour l'exemple exclu pendant la procédure de validation croisée leave-one-out.

Afin de faciliter l'écriture de ce théorème, nous introduisons les notations suivantes en réécrivant le système (3.4) :

$$A_{E,E}\alpha_E^a = \begin{pmatrix} -\frac{1}{Q-1}1_{(Q-1)m} \\ 0_Q \end{pmatrix} \tag{3.9}$$

avec

$$A_{E,E} = \begin{pmatrix} -H_{E,E}(\lambda) & M_I \\ 0_{(Q-1)m}^T & 1_Q^T \\ M_\Delta & 0_{Q-1,Q} \end{pmatrix}$$

et

$$\alpha_E^a = B = \begin{pmatrix} \alpha_E \\ b \end{pmatrix}.$$

Soit $\underline{i}$ l'ensemble des indices de α_E tel que les multiplicateurs de Lagrange correspondants soient associés à l'exemple x_i. Son cardinal est $Q-1$. Afin d'éviter la surcharge des notations, α_E^a sera noté B dans la preuve. De plus, les minuscules ne désignent pas des vecteurs (ce sont des matrices) mais rappellent que dans le cas bi-classe, ces quantités sont des vecteurs. Réordonnons à présent la matrice $A_{E,E}$ et le vecteur α_E^a de manière à avoir :

$$\begin{pmatrix} a_{\underline{i},\underline{i}} & a_{\underline{i}}^T \\ b_{\underline{i}} & A^{\underline{i}} \end{pmatrix} \begin{pmatrix} B_{\underline{i}} \\ B_{\underline{i}}^C \end{pmatrix} = \begin{pmatrix} -\frac{1}{Q-1}1_{(Q-1)m} \\ 0_Q \end{pmatrix}, \tag{3.10}$$

avec :

- $a_{\underline{i},\underline{i}} \in \mathcal{M}_{Q-1,Q-1}(\mathbb{R})$ la sous-matrice de $A_{E,E}$ constituée uniquement des lignes et des colonnes d'indices dans $\underline{i}$,
- $a_{\underline{i}}^T \in \mathcal{M}_{Q-1,(Q-1)(m-1)+Q}(\mathbb{R})$ la sous-matrice de $A_{E,E}$ obtenue en sélectionnant uniquement les lignes d'indices dans $\underline{i}$ et en enlevant les colonnes d'indices dans $\underline{i}$,
- $b_{\underline{i}} \in \mathcal{M}_{(Q-1)(m-1)+Q,Q-1}(\mathbb{R})$ est construite de la même manière en sélectionnant les colonnes d'indices dans $\underline{i}$ et en enlevant les lignes d'indices dans $\underline{i}$.
- $A^{\underline{i}} \in \mathcal{M}_{(Q-1)(m-1)+Q,(Q-1)(m-1)+Q}(\mathbb{R})$ la sous-matrice de $A_{E,E}$ constituée de $A_{E,E}$ privé des lignes et des colonnes d'indices dans $\underline{i}$.
- $B_{\underline{i}} \in \mathbb{R}^{(Q-1)}$ le vecteur construit à partir de B en ne sélectionnant que les coefficients d'indices dans $\underline{i}$.
- $B_{\underline{i}}^C \in \mathbb{R}^{(Q-1)(m-1)+Q}$ construit comme le "complémentaire" de $B_{\underline{i}}$.

En suivant la notation habituelle, nous notons h^i la fonction calculée par la LS-M-SVM sur l'ensemble d'apprentissage privé de l'exemple i. Nous admettons que $A_{E,E}$ et $A^{\underline{i}}$ sont inversibles.

Théorème 9 (Expression analytique de la fonction calculée par la LS-M-SVM lors de la procé-

dure de validation croisée leave-one-out). *Soit une LS-M-SVM à Q catégories entraînée sur d_m à l'aide du système* (3.10). *Considérons à présent la même machine entraînée sur $d_m \setminus \{(x_i, y_i)\}$. Alors, la valeur de la fonction correspondante au point x_i est*

$$h^i_{[\![1,Q]\!]\setminus y_i}(x_i) = -\frac{1}{Q-1}1_{Q-1} - \left(\left(A^{-1}_{E,E}\right)_{\underline{i},\underline{i}}\right)^{-1}\alpha_{\underline{i}}. \tag{3.11}$$

Démonstration. Le principe de cette démonstration est le même que celui de la démonstration de la proposition 8. Les $Q-1$ premières équations de (3.10) (celles associées à $\underline{i}$) permettent d'obtenir

$$\begin{aligned} a_{\underline{i},\underline{i}}B_{\underline{i}} + a^T_{\underline{i}}B^C_{\underline{i}} &= -\frac{1}{Q-1}1_{Q-1} \\ \Leftrightarrow a^T_{\underline{i}}B^C_{\underline{i}} &= -\frac{1}{Q-1}1_{Q-1} - a_{\underline{i},\underline{i}}B_{\underline{i}} \end{aligned} \tag{3.12}$$

alors que les $(Q-1)\,m + Q$ équations restantes donnent

$$b_{\underline{i}}B_{\underline{i}} + A^{\underline{i}}B^C_{\underline{i}} = \begin{pmatrix} -\frac{1}{Q-1}1_{(Q-1)(m-1)} \\ 0_Q \end{pmatrix}. \tag{3.13}$$

Soit $B^{(-i)} \in \mathbb{R}^{(Q-1)(m-1)}$ le vecteur des multiplicateurs de Lagrange de la LS-M-SVM entraînée sur l'ensemble d'apprentissage privé de i. Sa formule analytique est donnée par (3.9) et s'écrit :

$$\begin{aligned} A^{\underline{i}}B^{(-i)} &= \begin{pmatrix} -\frac{1}{Q-1}1_{(Q-1)(m-1)} \\ 0_Q \end{pmatrix} \\ \Leftrightarrow B^{(-i)} &= \left(A^{\underline{i}}\right)^{-1}\begin{pmatrix} -\frac{1}{Q-1}1_{(Q-1)(m-1)} \\ 0_Q \end{pmatrix}. \end{aligned} \tag{3.14}$$

D'après la définition de h^i, $h^i_{[\![1,Q]\!]\setminus y_i}(x_i)$ est un vecteur de $\mathbb{R}^{Q-1}$ constitué de la valeur de h^i en x_i pour toutes les catégories sauf y_i. Par définition, nous avons

$$\begin{aligned} h^i_{[\![1,Q]\!]\setminus y_i}(x_i) &= a^T_{\underline{i}}B^{(-i)} \\ &= a^T_{\underline{i}}\left(A^{\underline{i}}\right)^{-1}\begin{pmatrix} -\frac{1}{Q-1}1_{(Q-1)(m-1)} \\ 0_Q \end{pmatrix} && \text{avec (3.14)} \\ &= a^T_{\underline{i}}\left(A^{\underline{i}}\right)^{-1}\left(b_{\underline{i}}B_{\underline{i}} + A^{\underline{i}}B^C_{\underline{i}}\right) && \text{avec (3.13)} \\ &= a^T_{\underline{i}}B^C_{\underline{i}} + a^T_{\underline{i}}\left(A^{\underline{i}}\right)^{-1}b_{\underline{i}}B_{\underline{i}} \\ &= -\frac{1}{Q-1}1_{Q-1} - a_{\underline{i},\underline{i}}B_{\underline{i}} + a^T_{\underline{i}}\left(A^{\underline{i}}\right)^{-1}b_{\underline{i}}B_{\underline{i}} && \text{avec (3.12)} \\ &= -\frac{1}{Q-1}1_{Q-1} - \left(a_{\underline{i},\underline{i}} - a^T_{\underline{i}}\left(A^{\underline{i}}\right)^{-1}b_{\underline{i}}\right)B_{\underline{i}}. \end{aligned} \tag{3.15}$$

Dans (3.15) on reconnaît, devant $B_{\underline{i}}$, la formule d'inversion par blocs

$$\left(a_{\underline{i},\underline{i}} - a_{\underline{i}}^T \left(A^{\underline{i}}\right)^{-1} b_{\underline{i}}\right)^{-1} = \left(A_{E,E}^{-1}\right)_{\underline{i},\underline{i}} \tag{3.16}$$

avec $\left(A_{E,E}^{-1}\right)_{\underline{i},\underline{i}}$ la sous-matrice de $A_{E,E}^{-1}$ construite en sélectionnant seulement les indices des lignes et des colonnes dans $\underline{i}$. Par substitution de (3.16) dans (3.15) nous obtenons le résultat annoncé puisque $B_{\underline{i}} = \alpha_{\underline{i}}$. □

La LS-M-SVM classe mal un exemple lorsque $h_{y_i}^i(x_i) \leqslant \max_{k \in [\![1,Q]\!] \setminus y_i} h_k^i(x_i)$. Or puisque $\sum_k h_k(x_i) = 0$, nous obtenons naturellement

$$h_{y_i}^i(x_i) = -\sum_{k \neq y_i} h_k^i(x_i)$$

ce qui permet d'évaluer la qualité du classement pour l'exemple exclu de la procédure de validation croisée leave-one-out à l'aide du théorème 9.

Corollaire 2 (Erreur de validation croisée leave-one-out pour la LS-M-SVM)**.** *Soit une LS-M-SVM à Q catégories entraînée sur d_m. L'erreur de validation croisée leave-one-out de cette machine peut être obtenue sans avoir à effectuer cette procédure grâce au théorème 9.*

Une implantation naïve de (3.11) demande l'inversion de $A_{E,E}$ (en $O\left(((Q-1)m)^3\right)$ opérations) et m inversions de matrices de taille $Q-1$. Dans le cas des matrices de Gram de rang faible, nous montrons qu'il est possible de calculer cette erreur pour un coût beaucoup plus faible. En effet, les $(Q-1)m$ premières lignes et colonnes de $A_{E,E}^{-1}$ s'obtiennent par application de la formule d'inversion par bloc :

$$\left(A_{E,E}^{-1}\right)_{[\![1,(Q-1)m]\!],[\![1,(Q-1)m]\!]} = -H_{E,E}(\lambda)^{-1} - \nu(D - C\nu)^{-1} C H_{E,E}(\lambda)^{-1} \tag{3.17}$$

avec

- $D = \begin{pmatrix} 1_Q^T \\ 0_{Q-1,Q} \end{pmatrix} \in \mathcal{M}_{Q,Q}(\mathbb{R})$,
- $C = \begin{pmatrix} 0_{(Q-1)m}^T \\ M_\Delta \end{pmatrix} \in \mathcal{M}_{Q,(Q-1)m}(\mathbb{R})$.

Cette formule est particulièrement intéressante puisque seules des sous-matrices de la matrice constituée des $(Q-1)m$ premières lignes et colonnes de $A_{E,E}^{-1}$ sont manipulées par le théorème 9. L'utilisation de (3.17) et de (3.8) permet d'obtenir une estimation de l'erreur de généralisation pour une complexité linéaire en le nombre d'exemples : $O\left(mr^2(Q-1)^3\right)$. Nous retrouvons la complexité annoncée dans le cas bi-classe au facteur $(Q-1)^3$ près.

3.2 Parcours du chemin de régularisation pour la M-SVM^2

Maintenant que nous avons développé les principaux résultats pour la LS-M-SVM, nous présentons l'extension à la M-SVM^2 des travaux effectués sur la ℓ_2-SVM. Nous commençons donc par la partition de l'ensemble d'apprentissage. A partir de cette partition nous proposons une méthode de calcul des multiplicateurs de Lagrange en nous aidant de celle proposée pour la LS-M-SVM. En suivant la même démarche que celle du cas bi-classe, nous étudions le comportement de la M-SVM^2 pour les grandes valeurs de λ. Nous proposons ensuite une méthode pour calculer le prochain point d'arrêt de l'algorithme en étudiant les changements dans la partition de l'ensemble d'apprentissage. Finalement, nous présentons une synthèse des différences entre le cas bi-classe et le cas multi-classe.

3.2.1 Partition de l'ensemble d'apprentissage

La définition de l'ensemble $\mathcal{E}$ que nous proposons est similaire à celle du cas bi-classe au sens où elle se base sur les conditions de Kuhn-Tucker. L'objectif est d'obtenir un ensemble permettant de suivre l'évolution des multiplicateurs de Lagrange non nuls. Pour cela, nous faisons usage des conditions complémentaires qui sont :

$$\forall i \in [\![1,m]\!], \forall k \in [\![1,Q]\!] \setminus \{y_i\},\ \alpha_{ik}(\lambda) \left[h_{\lambda,k}(x_i) + \frac{1}{Q-1} - \xi_{ik}(\lambda) \right] = 0. \tag{3.18}$$

Si $\alpha_{ik}(\lambda)$ est égal à zéro, cette condition est toujours vérifiée. Si $\alpha_{ik}(\lambda)$ est diffèrent de zéro, il faut que le facteur de droite soit nul. Or, le choix de la matrice M (voir la table 1.1) permet d'exprimer, à l'optimum, les variables d'écart en fonction des multiplicateurs de Lagrange :

$$\forall i \in [\![1,m]\!], \forall k \in [\![1,Q]\!] \setminus \{y_i\},\ \xi_{ik}(\lambda) = \frac{1}{C} \left[\alpha_{ik}(\lambda) - \frac{1}{Q} \sum_l \alpha_{il}(\lambda) \right]. \tag{3.19}$$

En utilisant (3.18) et (3.19), nous proposons la définition de l'ensemble $\mathcal{E}(\lambda)$:

$$\mathcal{E}(\lambda) = \left\{ (j,k) \in [\![1,m]\!] \times [\![1,Q]\!] \setminus \{y_j\} / h_{\lambda,k}(x_j) + \frac{1}{Q-1} - \frac{1}{C} \left[\alpha_{jk}(\lambda) - \frac{1}{Q} \sum_l \alpha_{jl}(\lambda) \right] = 0 \right\}. \tag{3.20}$$

Cette définition de $\mathcal{E}(\lambda)$ correspond à prendre les couples (exemple, catégorie) associés à une sortie du classifieur à marge dure égale à $-\frac{1}{Q-1}$. Par analogie avec l'appellation vecteurs support, ces couples seront désignés sous le terme de "couples support". Par abus de notation, comme pour la LS-M-SVM, chacun de ces couples (i,k) sera confondu avec l'indice $(j-1)Q+k$. L'ensemble $\mathcal{I}(\lambda)$ est défini comme l'ensemble des couples restants à l'exception des couples correspondant aux variables artificielles. Contrairement à la définition de $\mathcal{E}$ donnée par l'équation (2.6), il n'est plus possible de travailler avec la machine à marge douce puisque les variables d'écart de la M-SVM^2 peuvent être négatives.

3.2.2 Calcul des multiplicateurs de Lagrange

Nous présentons ici les calculs nécessaires à l'obtention des valeurs des multiplicateurs de Lagrange de la M-SVM^2. Pour ce faire, nous nous appuyons entièrement sur les notations et les calculs présentés pour la LS-M-SVM (voir le problème 15). En effet, il suffit de remplacer E par $\mathcal{E}$ pour obtenir les relations de la M-SVM^2. Toutefois, nous présentons quand même leur obtention car le calcul pour obtenir un tel résultat n'est pas trivial. D'après (3.20), le système vérifié à l'optimum par les couples (exemple, catégorie) de $\mathcal{E}$ est

$$\forall (j,k) \in \mathcal{E}(\lambda),\ h_{\lambda,k}(x_j) + \frac{1}{Q-1} - \frac{1}{C}\left[\alpha_{jk}(\lambda) - \frac{1}{Q}\sum_l \alpha_{jl}(\lambda)\right] = 0.$$

Avec les notations introduites pour la LS-M-SVM, la réécriture matricielle de ce système est :

$$-H_{\mathcal{E},\mathcal{E}}(\lambda)\,\alpha_{\mathcal{E}} + M_I b = -\frac{1}{Q-1}1_{m_{\mathcal{E}}}$$

avec $m_{\mathcal{E}}$ le cardinal de $\mathcal{E}$. La reformulation matricielle des contraintes de sommation à zéro des fonctions composantes s'écrit à l'aide du système [1]

$$\begin{cases} M_\Delta \alpha_{\mathcal{E}} = 0_{Q-1} \\ 1_Q^T b = 0. \end{cases} \tag{3.21}$$

Nous utilisons les mêmes notation que celle de la proposition 5, c'est-à-dire :

$$A_{\mathcal{E}}(\lambda) = \begin{pmatrix} -H_{\mathcal{E},\mathcal{E}}(\lambda) & M_I \\ 0_{m_{\mathcal{E}}}^T & 1_Q^T \\ M_\Delta & 0_{Q-1,Q} \end{pmatrix},$$

$$C_{\mathcal{E}} = \begin{pmatrix} -\frac{1}{Q-1}1_{m_{\mathcal{E}}} \\ 0 \\ 0_{Q-1} \end{pmatrix}$$

et

$$\alpha_{\mathcal{E}}^a(\lambda) = \left(\alpha_0(\lambda)\ \ \alpha_{\mathcal{E}}(\lambda)^T\right)^T$$

avec les matrices M_I et M_Δ dépendant de $\mathcal{E}$. Nous obtenons ainsi la proposition suivante qui est l'extension de la proposition 5 :

Proposition 9. *Pour tout $\lambda \in \mathbb{R}_+^*$, le vecteur $\alpha_{\mathcal{E}}^a(\lambda)$ est la solution du système d'équations linéaires :*

$$A_{\mathcal{E}}(\lambda)\,\alpha_{\mathcal{E}}^a(\lambda) = C_{\mathcal{E}}. \tag{3.22}$$

La dependance à λ étant dans le terme de gauche, comme pour la ℓ_2-SVM, nous nous doutons

1. Ces notations sont issues de la LS-M-SVM, puisque partageant les mêmes contraintes.

que l'évolution des multiplicateurs de Lagrange à $\mathcal{E}$ constant ne sera pas linéaire. Nous utilisons alors la même approche que celui du cas bi-classe pour l'obtention des multiplicateurs de Lagrange. Il est intéressant de remarquer que le système à résoudre est très semblable à celle de la LS-M-SVM. Toutefois, la différence notable entre ces deux problèmes est la structure de $\mathcal{E}$. En effet, cet ensemble, contrairement à E, n'est pas "régulier" au sens où chaque exemple n'a pas exactement $Q-1$ multiplicateurs de Lagrange non nuls. Il est donc nécessaire, comme dans le cas bi-classe, de déterminer cet ensemble $\mathcal{E}$.

3.2.2.1 Calcul dans le cas général

La résolution de (3.22) peut se faire efficacement en adoptant la même démarche que pour la LS-M-SVM (voir section 3.1.2). Nous supposons que $H_{\mathcal{E},\mathcal{E}}(\lambda)$ est inversible. En posant $\nu \in \mathcal{M}_{m_\mathcal{E},Q}(\mathbb{R})$ et $\rho \in \mathbb{R}^{m_\mathcal{E}}$ tels que :

$$\begin{cases} H_{\mathcal{E},\mathcal{E}}(\lambda)\,\nu = -M_I \\ H_{\mathcal{E},\mathcal{E}}(\lambda)\,\rho = \frac{1}{Q-1} 1_{m_\mathcal{E}} \end{cases}, \tag{3.23}$$

la solution de (3.22) s'écrit

$$\begin{cases} b = \begin{pmatrix} 1_Q^T \\ M_\Delta \nu \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ M_\Delta \rho \end{pmatrix} \\ \alpha_\mathcal{E} = \rho - \nu b \end{cases}$$

en supposant que la matrice $\begin{pmatrix} 1_Q^T \\ M_\Delta \nu \end{pmatrix}$ de taille Q soit inversible. La résolution de (3.23) en utilisant une décomposition de Cholesky Gaxpy est d'une complexité en $\frac{1}{3}m_\mathcal{E}^3 + 2(Q+1)m_\mathcal{E}^2$ opérations. Lorsque la matrice de Gram est de rang faible (devant m), nous pouvons appliquer la même démarche que celle utilisée pour la LS-M-SVM (et donc la ℓ_2-SVM).

3.2.2.2 Cas d'une matrice de Gram de rang faible

Lorsque la matrice de Gram est de rang faible, le calcul de ν et ρ peut se faire à coût réduit (voir le résultat sur la LS-M-SVM dans la section 3.1.2). Supposons que K est de rang r, alors il existe une matrice $R \in \mathcal{M}_{m,r}(\mathbb{R})$ telle que $K = RR^T$. De manière analogue, puisque $I_Q - \frac{1}{Q}1_Q 1_Q^T$ est de rang $Q-1$, il existe une matrice $T \in \mathcal{M}_{Q,Q-1}(\mathbb{R})$ telle que $I_Q - \frac{1}{Q}1_Q 1_Q^T = TT^T$. [1]

De la même manière que pour la LS-M-SVM, nous définissons :

- la matrice $R_\mathcal{E} \in \mathcal{M}_{m_\mathcal{E},(Q-1)r}(\mathbb{R})$ égale à la sous-matrice de $R \otimes T$ constituée seulement des lignes d'indices dans $\mathcal{E}$,
- la matrice $D_\mathcal{E} \in \mathcal{M}_{m_\mathcal{E},(Q-1)m}(\mathbb{R})$ égale à la sous-matrice de $I_m \otimes \left(I_Q - \frac{1}{Q}1_Q 1_Q^T\right)$ constituée

1. La matrice $I_Q - \frac{1}{Q}1_Q 1_Q^T$ ayant pour spectre 0 de multiplicité 1 et 1 de multiplicité $Q-1$, cette factorisation s'obtient en diagonalisant.

seulement des lignes et des colonnes d'indice dans $\mathcal{E}$.[1]

Ces notations permettent de formuler $H_{\mathcal{E},\mathcal{E}}(\lambda)$ comme la somme d'une matrice de rang plein et d'une matrice de rang $(Q-1)r$. Nous obtenons

$$H_{\mathcal{E},\mathcal{E}}(\lambda) = R_{\mathcal{E}}R_{\mathcal{E}}^T + \lambda D_{\mathcal{E}}.$$

A nouveau la formule de Sherman-Morrison-Woodbury donne une méthode pour calculer son inverse :

$$H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} = \frac{1}{\lambda}D_{\mathcal{E}}^{-1} - \frac{1}{\lambda^2}D_{\mathcal{E}}^{-1}R_{\mathcal{E}}\left(I_{(Q-1)r} + \frac{1}{\lambda}R_{\mathcal{E}}^T D_{\mathcal{E}}^{-1}R_{\mathcal{E}}\right)^{-1}R_{\mathcal{E}}^T D_{\mathcal{E}}^{-1}. \tag{3.24}$$

Comme pour la LS-M-SVM, la matrice $D_{\mathcal{E}}^{-1}R_{\mathcal{E}}$ est obtenue en seulement $m_{\mathcal{E}}r(Q-1)$ opérations en exploitant la parcimonie de $D_{\mathcal{E}}^{-1}$. Le produit de matrice $R_{\mathcal{E}}^T D_{\mathcal{E}}^{-1}R_{\mathcal{E}}$ nécessite $O\left(m_{\mathcal{E}}r^2(Q-1)^2\right)$ opérations. Nous retrouvons la complexité du calcul du vecteur ρ en $O\left(m_{\mathcal{E}}r(Q-1) + r^3(Q-1)^3 + m_{\mathcal{E}}r^2(Q-1)^2\right)$. Il est important de se souvenir que $m_{\mathcal{E}} \leqslant (Q-1)m$. Les besoins en mémoire sont donc de l'ordre de $O(m_{\mathcal{E}}r(Q-1))$ ce qui reste raisonnable lorsque le rang est faible.

3.2.3 Calcul du point de départ du chemin de régularisation

Pour définir un point de départ du chemin de régularisation, nous exploitons une particularité du problème 13 lorsque λ tend vers l'infini. Asymptotiquement la matrice $H(\lambda)$ est équivalente à $\lambda I_m \otimes \left(I_Q - \frac{1}{Q}1_Q 1_Q^T\right)$ si bien que la contribution des exemples dans le problème d'apprentissage ne se fait plus que par les contraintes. Le problème d'apprentissage "équivalent" est

Problème 17 (Problème equivalent au problème 13 quand λ tend vers l'infini)**.**

$$\max_{\alpha}\left\{-\frac{\lambda}{2}\alpha^T\left(I_m \otimes \left(I_Q - \frac{1}{Q}1_Q 1_Q^T\right)\right)\alpha + \frac{1}{Q}1_{Qm}^T\alpha\right\}$$

$$s.c. \quad \begin{cases} \forall i \in [\![1,m]\!],\ \forall k \in [\![1,Q]\!]\setminus\{y_i\},\ \ \alpha_{ik} \geqslant 0 \\ \forall k \in [\![1,Q-1]\!],\ \ \sum_{i=1}^m \alpha_{ik} = \sum_{i=1}^m \alpha_{i(k+1)}. \end{cases}.$$

Seule la catégorie des exemples intervient puisque la description n'apparaît plus dans le problème d'apprentissage. Il en découle que

$$\forall (k,l) \in [\![1,Q]\!]^2,\ \exists c_{k,l} \in \mathbb{R}_+ / \ \forall i \in [\![1,m]\!],\ \alpha_{il} = c_{y_i,l}.$$

Nous imposons $c_{l,k} = 0$ pour $l = k$ (penser à la définition de α). Le nombre de variables inconnues est donc de $Q(Q-1)$. La substitution des Q contraintes dans le problème de maximisation permet de n'avoir plus que $(Q-1)^2$ inconnues restantes. Nous retrouvons le résultat du

1. La formule analytique de D_E n'est plus valable pour $\mathcal{E}$.

cas bi-classe, où la solution est immédiate. Toutefois ici, trouver ces $(Q-1)^2$ variables est plus complexe et ne présente pas un intérêt fondamental. Neanmoins, il est important de remarquer que puisque seules les catégories interviennent, la seule différence qui existe entre les catégories est leur cardinalité. Il apparaît ainsi naturellement que si deux classes ont la même cardinalité, alors ces deux classes seront indiscernables pour la M-SVM^2 lorsque λ est suffisamment grand [1].

3.2.4 Détection des changements dans la partition de l'ensemble d'apprentissage

Nous suivons le même raisonnement que celui du cas bi-classe. Soit $\left(\lambda^l\right)_{l\in\mathbb{N}^*}$ une suite strictement décroissante de valeurs du coefficient de régularisation associées au changement des ensembles $\mathcal{E}$ et $\mathcal{I}$. Si la séquence $\left(\lambda^l\right)_{l\in\mathbb{N}^*}$ est connue jusqu'à l'indice l, alors trouver λ^{l+1} revient à trouver la plus grande valeur de $\lambda \in]0, \lambda^l[$ telle qu'un couple entre ou sorte de $\mathcal{E}\left(\lambda^l\right)$. L'entrée du couple (i,k) est donnée par l'équation

$$h_{\lambda,k}(x_i) + \frac{1}{Q-1} - \lambda\left[\alpha_{ik}(\lambda) - \frac{1}{Q}\sum_u \alpha_{iu}(\lambda)\right] = \tilde{h}_{\lambda,k}(x_i) + \frac{1}{Q-1} = 0$$

tandis que la sortie d'un couple correspond à l'annulation de son multiplicateur de Lagrange [2]. De la même manière que pour la ℓ_2-SVM, nous choisissons une approximation de la valeur de λ^{l+1} en approchant $\tilde{h}_{\lambda,k}(x_i)$ par son développement en série de Taylor du premier ordre. Les valeurs de λ_{ik}^{l+1} sont alors obtenues par :

$$\begin{cases} \forall (i,k) \in \mathcal{I}, \lambda_{ik}^{l+1} = \lambda_{ik}^l - \frac{\frac{1}{Q-1} + \tilde{h}_{\lambda^l,k}(x_i)}{\frac{\partial \tilde{h}_{\lambda^l,k}(x_i)}{\partial \lambda}}. \\ \forall (i,k) \in \mathcal{E}, \lambda_{ik}^{l+1} = \lambda_{ik}^l - \frac{\alpha_{\mathcal{E}}(\lambda)}{\frac{\partial \alpha_{\mathcal{E}}(\lambda)}{\partial \lambda}} \end{cases}$$

La dérivée de $\tilde{h}_{\lambda^l,k}(x_i)$ par rapport à λ s'obtient facilement à partir de $\frac{\partial \alpha_{\mathcal{E}}(\lambda)}{\partial \lambda}$ et $\frac{\partial b}{\partial \lambda}$, elles mêmes dérivées de (3.22) en utilisant une inversion par bloc de $A'_{\mathcal{E}}(\lambda)$.

$$\begin{cases} \frac{\partial b_\lambda}{\partial \lambda} = \left(\begin{pmatrix} 1_Q^T \\ 0_{Q-1,Q} \end{pmatrix} - \begin{pmatrix} 0_{m_\mathcal{E}}^T \\ M_\Delta \end{pmatrix} \nu\right)^{-1} \begin{pmatrix} 0_{m_\mathcal{E}}^T \\ M_\Delta \end{pmatrix} H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} \alpha_{\mathcal{E}}(\lambda) \\ \frac{\partial \alpha_{\mathcal{E}}(\lambda)}{\partial \lambda} = -H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1} \alpha_{\mathcal{E}}(\lambda) - \frac{\partial b_\lambda}{\partial \lambda}\nu \end{cases} . \tag{3.25}$$

Cette équation est de structure similaire à celle obtenue dans le cas bi-classe (2.15). La stratégie de choix de λ^{l+1} est donc la même que celle du cas bi-classe, c'est-à-dire que nous choisissons λ^{l+1} tel qu'il y ait $Qm2/100$ valeurs de λ_{ik}^{l+1} comprises entre λ^l et λ^{l+1}. Ce choix correspond à avoir environ 2% des exemples qui changent d'ensemble à chaque pas, comme dans le cas bi-classe.

1. En pratique ce comportement n'est pas vu à cause de la précision numérique.
2. En effet, si ce couple ne sortait pas de $\mathcal{E}\left(\lambda^l\right)$ alors, puisque sa dérivée est non nulle, le multiplicateur de Lagrange deviendrait négatif.

3.2.5 Comparaison du parcours bi-classe et multi-classe et vue globale de l'algorithme de parcours du chemin

Le diagramme de l'algorithme donné par la figure 3.1 est identique à celui du cas bi-classe puisqu'ils sont identiques dans leurs conceptions. Nous avons choisi de représenter sur ce diagramme non pas le cas où K n'est pas de rang faible mais celui où l'on l'approche par une telle matrice. Toutefois, quelques différences entres ces deux algorithmes méritent une attention particulière :

- Pour la M-SVM^2, l'ensemble $\mathcal{E}$ est défini à l'aide de la formulation à marge dure. En effet, pour cette machine, il n'y a pas équivalence entre les définitions de cet ensemble à l'aide de la marge douce et de la marge dure. Ceci est dû à la négativité possible des variables d'écart. Cette différence se répercute sur la facilité d'utilisation de $\mathcal{E}$, par exemple lors de la détection de changement d'ensemble.
- Un exemple peut n'être vecteur support que pour quelques catégories et donc ne pas être vecteur support "complet". Cela impose d'introduire une notion de couple exemple-catégorie support.
- Le nombre d'exemples et le rang sont multipliés par un facteur $Q-1$ dans les calculs de complexité. Cette complexité nous donne une idée du prix à payer pour traiter les problèmes multi-classes.
- Dans le cas d'une approximation de K par une matrice de rang faible, l'évaluation du classifieur pendant la phase de test ne se fait plus comme dans le cas bi-classe. En effet, puisque la cardinalité de $\mathcal{E}$ est souvent très élevée, nous recommandons d'utiliser l'expression analytique des fonctions propres du noyau (voir par exemple [53]) plutôt que de résoudre un système linéaire (voir la section 2.2.6.3).

3.3 Sélection de modèle pour la M-SVM^2

Cette section présente nos contributions à la sélection de modèle pour la M-SVM^2. Tout d'abord nous présentons une nouvelle borne exacte sur l'erreur de validation croisée leave-one-out basée sur le concept de la borne Span [105]. Nous montrons que, contrairement au cas bi-classe, cette borne exacte ne permet pas d'obtenir une estimation précise de l'erreur de validation croisée leave-one-out. Nous nous intéressons ensuite à l'adaptation à la M-SVM^2 de l'expression analytique de cette erreur pour la LS-M-SVM, ceci afin d'en obtenir une estimation. Nous étudions alors son intégration dans le parcours du chemin de régularisation. Dans l'annexe A, nous présentons également un algorithme d'optimisation locale de la borne Rayon-Marge. Cet algorithme n'est pas intégré dans ce chapitre car hors du contexte du chemin de régularisation.

3.3.1 Borne exacte sur l'erreur de validation croisée leave-one-out

Une des contributions principales de cette section est une borne exacte sur l'erreur de validation croisée leave-one-out en utilisant l'approche des espaces engendrés par les vecteurs support

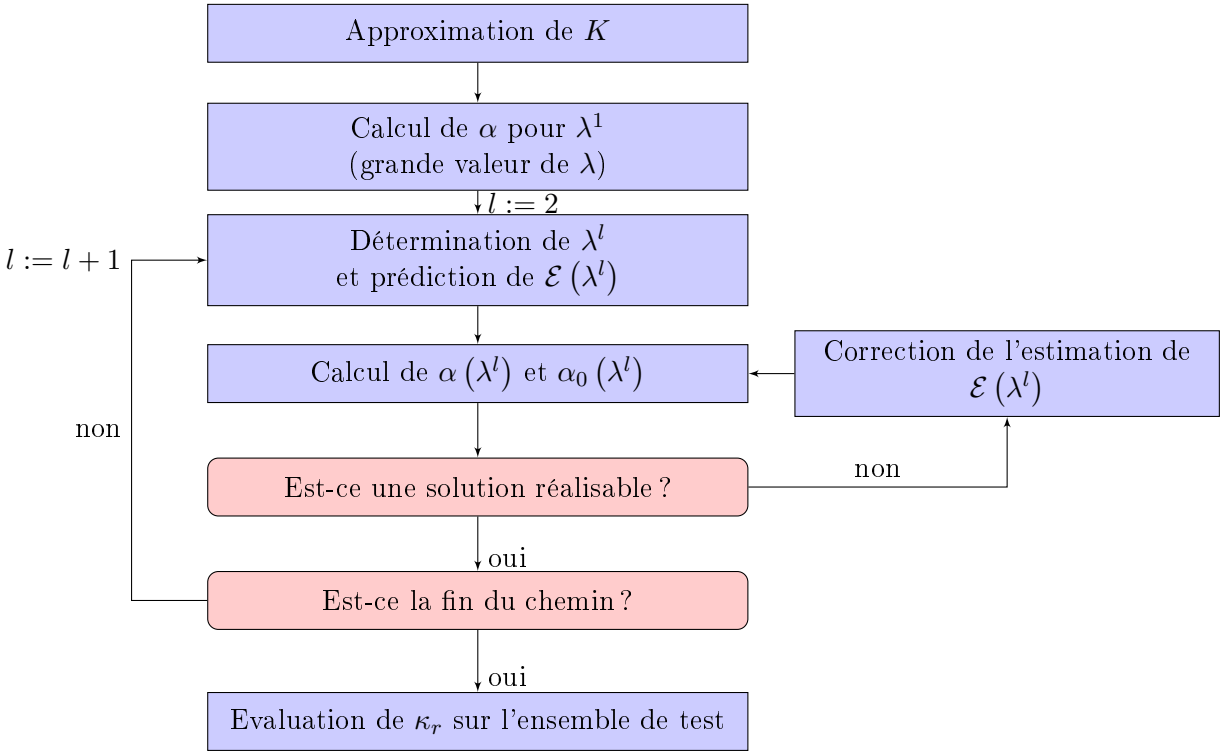


FIGURE 3.1 – Diagramme de l'algorithme de parcours du chemin de régularisation pour la M-SVM²

présentée dans [105]. Ensuite nous montrons que, contrairement au cas bi-classe, cette approche ne permet pas d'en obtenir une estimation précise.

Soit $\alpha^* \in \mathbb{R}_+^{Qm}(d_m)$ une solution optimale du problème 11[1]. Soit h^* la fonction qui lui est associée et soit $\alpha^*_{p,max} = \max_l \alpha^*_{pl}$.

Définition 14. *Soit $d\Lambda_{p,k}$ l'ensemble des normes au carré de chaque point de l'espace engendré par combinaison linaire contrainte des exemples relativement à la catégorie k et à l'exemple d'indice p :*

$$d\Lambda_{p,k} = \left\{ \left\| \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\frac{1}{Q} - \delta_{k,l} \right) \lambda_{il} \kappa_{x_i} \right\|_\kappa^2 \text{ tel que } \lambda \in \mathcal{C}_{pk} \right\}$$

avec le convexe des contraintes $\mathcal{C}_{pk}$ défini par

$$\begin{cases} \forall l \in [\![1, Q]\!], \lambda_{pl} = -\frac{\alpha^*_{pl}}{\alpha^*_{p,max}} \\ \forall i \in [\![1, m]\!], \forall l \in [\![1, Q]\!], 0 \leqslant -\lambda_{il} \alpha^*_{p,max} \leqslant \alpha^*_{il} \\ \forall k \in [\![1, Q-1]\!], \ \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\frac{1}{Q} - \delta_{k,l} \right) \lambda_{il} = 0 \end{cases} .$$

Le d de $d\Lambda_{p,k}$ fait référence au fait que les quantités de cet ensemble jouent le rôle de distances

1. Nous choisissons de noter ce vecteur α^* plutôt que β^* afin de garder les notations de [63].

au carré contrairement à l'ensemble Λ_p du cas bi-classe où ce sont des éléments de l'espace de représentation. Nous définissons à présent le span de manière analogue au cas bi-classe :

Définition 15 (Span de p relativement à la catégorie k)**.** *Soit $d\Lambda_{p,k}$ l'ensemble de la définition 14. Soit $S_{p,k}$ le* span de p relativement à la catégorie k *tel que :*

$$S_{p,k}^2 = \min_{\lambda} d\Lambda_{p,k}.$$

Nous énonçons à présent le lemme permettant d'obtenir la borne exacte sur l'erreur de validation croisée leave-one-out :

Lemme 8. *Soit une LLW-M-SVM à marge dure entraînée sur d_m. Considérons à présent la même machine entraînée sur $d_m \setminus \{(x_p, y_p)\}$. Si elle fait une erreur sur (x_p, y_p), alors*

$$\max_{1 \leqslant l \leqslant Q} {\alpha_{pl}^*}^2 \geqslant \frac{Q}{(Q-1)^4 \sum_{k=1}^{Q} S_{p,k}^2 D_m^2},$$

avec D_m le diamètre de la plus petite boule de $\mathbf{H}_\kappa$ contenant l'ensemble $\{\kappa_{x_i} : 1 \leqslant i \leqslant m\}$ et $S_{p,k}$ le span de p relativement à la catégorie k.

Ce lemme est très semblable au lemme 1 permettant d'obtenir la borne Rayon-Marge et au lemme 3 de [105]. Le tableau 3.1 met en évidence des dépendances similaires au span et au rayon.

	Borne Radius-Margin	Borne Span
ℓ_2-SVM	$\alpha_p \geqslant \frac{1}{D_m^2}$	$\alpha_p \geqslant \frac{1}{D_m S_p}$
M-SVM^2	$\max_{1 \leqslant l \leqslant Q} {\alpha_{pl}^*}^2 \geqslant \frac{Q^2}{(Q-1)^6 D_m^4}$	$\max_{1 \leqslant l \leqslant Q} {\alpha_{pl}^*}^2 \geqslant \frac{Q}{(Q-1)^4 \sum_{k=1}^{Q} S_{p,k}^2 D_m^2}$

TABLE 3.1 – Comparaison des lemmes des bornes Rayon-Marge et Span dans le cadre bi- et multi-classe

Démonstration. Soit $h^p \in \mathcal{H}_{\kappa,Q}$ la fonction calculée par la machine entraînée sur $d_m \setminus \{(x_p, y_p)\}$. Soit $\alpha^p = \left(\alpha_{ik}^p\right) \in \mathbb{R}_+^{Qm}(d_m)$ le vecteur des variables duales correspondantes avec $\left(\alpha_{pk}^p\right)_{1 \leqslant k \leqslant Q} = 0_Q$. Nous définissons deux solutions réalisables du problème 11 : δ^p et μ^p. Le vecteur δ^p est tel que $\alpha^* - \delta^p$ est une solution réalisable du problème 11 sous la contrainte additionnelle que $\left(\alpha_{pk}^* - \delta_{pk}^p\right)_{1 \leqslant k \leqslant Q} = 0_Q$, c'est-à-dire que $\alpha^* - \delta^p$ satisfasse les mêmes contraintes que α^p. Nous avons alors

$$\begin{cases} \forall k \in [\![1, Q]\!], \;\; \delta_{pk}^p = \alpha_{pk}^* \\ \forall i \in [\![1, m]\!] \setminus \{p\}, \; \forall k \in [\![1, Q]\!], \;\; 0 \leqslant \delta_{ik}^p \leqslant \alpha_{ik}^* \\ \forall k \in [\![1, Q-1]\!], \;\; \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\frac{1}{Q} - \delta_{k,l}\right) \delta_{il}^p = 0 \end{cases} . \tag{3.26}$$

Pour la suite de la démonstration, nous notons J au lieu de $J_{\text{LLW,d}}$. Par définition de μ^p, $\alpha^p + \mu^p$ est une solution réalisable du problème 11. Nous avons alors $J(\alpha^* - \delta^p) \leqslant J(\alpha^p)$ et $J(\alpha^p + \mu^p) \leqslant J(\alpha^*)$ et donc

$$J(\alpha^*) - J(\alpha^* - \delta^p) \geqslant J(\alpha^*) - J(\alpha^p) \geqslant J(\alpha^p + \mu^p) - J(\alpha^p). \tag{3.27}$$

En utilisant les notations de [105], les différences des fonctions objectif s'écrivent

$$I_2 = J(\alpha^*) - J(\alpha^* - \delta^p) = \frac{1}{2}\delta^{pT} H \delta^p + \nabla J(\alpha^*)^T \delta^p$$

et

$$I_1 = J(\alpha^p + \mu^p) - J(\alpha^p) = -\frac{1}{2}\mu^{pT} H \mu^p + \nabla J(\alpha^p)^T \mu^p.$$

La démonstration se déroule en deux étapes.

- La minoration de I_1 qui est strictement identique à celle effectuée dans le démonstration du lemme 1 de [63]. Cette étape sera donc traitée dans un second temps.
- Le calcul de I_2 qui est similaire dans l'idée à la démonstration du lemme de la borne Span [105].

Calcul de I_2 :
A la manière de [105], nous choisissons δ^p de terme général $\delta^p_{ik} = -\lambda_{ik}\alpha^*_{p,max}$ avec λ_{ik} tel que δ^p satisfasse (3.26) c'est-à-dire :

- $\forall l \in [\![1, Q]\!], \lambda_{pl} = -\frac{\alpha^*_{pl}}{\alpha^*_{p,max}}$,
- $\forall i \in [\![1, m]\!], \forall k \in [\![1, Q]\!], 0 \leqslant -\lambda_{ik}\alpha^*_{p,max} \leqslant \alpha^*_{ik}$,
- $\forall k \in [\![1, Q-1]\!]$, $\sum_{i=1}^m \sum_{l=1}^Q \left(\frac{1}{Q} - \delta_{k,l}\right) \lambda_{il} = 0$,

Nous remarquons que si $\alpha^*_{ik} = 0$ alors $\lambda_{ik} = 0$ grâce aux contraintes inégalités. Le vecteur λ possède les mêmes coefficients à zéro que α^*.

Montrons que le gradient de J en α^* dans la direction δ^p est égal au vecteur nul. Nous avons

$$\nabla J(\alpha^*)^T \delta^p = \sum_{i=1}^m \sum_{k=1}^Q \delta^p_{ik} \left(\bar{h}^*_k(x_i) + \frac{1}{Q-1}\right).$$

Or en utilisant $\sum_{i=1}^m \sum_{k=1}^Q \left(\frac{1}{Q} - \delta_{l,k}\right) \delta^p_{ik} = 0$ nous pouvons écrire $\sum_{l=1}^Q \left(\sum_{i=1}^m \sum_{k=1}^Q \left(\frac{1}{Q} - \delta_{l,k}\right) \delta^p_{ik}\right) b_l = 0$, ce qui, factorisé dans $\nabla J(\alpha^*)^T \delta^p$, donne

$$\begin{aligned}
\nabla J(\alpha^*)^T \delta^p &= \sum_{i=1}^m \sum_{k=1}^Q \delta^p_{ik} \left(\bar{h}^*_k(x_i) + \frac{1}{Q-1} - \sum_{l=1}^Q \left(\frac{1}{Q} - \delta_{l,k}\right) b_l\right) \\
&= \sum_{i=1}^m \sum_{k=1}^Q \delta^p_{ik} \left(\bar{h}^*_k(x_i) + \frac{1}{Q-1} + b_k\right) \text{ puisque } \textstyle\sum_k b_k = 0 \\
&= \sum_{i=1}^m \sum_{k=1}^Q \delta^p_{ik} \left(h^*_k(x_i) + \frac{1}{Q-1}\right).
\end{aligned}$$

Puisque les composantes nulles de δ^p sont au moins celles de α^*, les conditions de Kuhn-Tucker imposent que la dernière double somme soit nulle. Il en résulte que l'expression de I_2 se réduit à

la forme quadratique suivante :

$$\begin{aligned} I_2 &= \frac{1}{2}\delta^{pT} H \delta^p \\ &= \frac{1}{2}\sum_{k=1}^{Q}\left\|\sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q}-\delta_{k,l}\right)\delta_{il}^p \kappa_{x_i}\right\|_\kappa^2 \\ &= \frac{1}{2}\sum_{k=1}^{Q}\left\|\sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q}-\delta_{k,l}\right)\lambda_{il} \kappa_{x_i}\right\|_\kappa^2 {\alpha^*_{p,max}}^2. \end{aligned}$$

Nous reconnaissons, dans cette dernière équation, un élément de $d\Lambda_{p,k}$. Le terme I_2 étant le majorant de (3.27), le minimiser revient à obtenir une majoration plus précise. De plus minimiser I_2, revient à choisir la valeur de λ correspondant au span de p relativement à k, ce qui permet d'écrire l'égalité suivante :

$$I_2 = \frac{1}{2}\sum_{k=1}^{Q} S_{p,k}^2 {\alpha^*_{p,max}}^2 \tag{3.28}$$

et qui conclut la première partie de la démonstration.

Minoration de I_1 :

Ce calcul est exactement celui de la borne Rayon-Marge multi-classe [63] (équations (18) à (21)). Il s'appuie sur deux constatations :

- Si la M-SVM entraînée sur $d_m \setminus \{x_p, y_p\}$ (avec $\alpha^p \neq 0$) commet une erreur sur l'exemple x_p, alors il existe une catégorie $n \in [\![1, Q]\!] \setminus \{y_p\}$ tel que $h_n^p(x_p) > 0$ et la solution n'est pas optimale pour le problème d'origine.
- De plus, pour toute M-SVM à marge dure entraînée $d_m \setminus \{x_p, y_p\}$ (avec $\alpha^p \neq 0$) il existe une application $\mathcal{I}$ de $[\![1, Q]\!]$ sur $[\![1, m]\!] \setminus \{p\}$ telle que $\forall k \in [\![1, Q]\!],\ h_k^p\left(x_{\mathcal{I}(k)}\right) = -\frac{1}{Q-1}$.[1]

Nous choisissons le vecteur μ^p tel que :

$$\begin{cases} \mu_{pn}^p = K \\ \forall k \in [\![1, Q]\!] \setminus \{n\}, \quad \mu_{\mathcal{I}(k)k}^p = K \end{cases}$$

avec $K \in R_+^*$. Cette définition de μ^p satisfait les contraintes de la M-SVM à marge dure et

1. Penser aux conditions de Kuhn-Tucker et aux contraintes.

permet d'obtenir l'égalité suivante :

$$\begin{aligned}\nabla J\left(\alpha^p\right)^T \mu^p &= \sum_{i=1}^{m}\sum_{k=1}^{Q} \mu_{ik}^p \left(\langle h_k^p, \kappa_{x_i}\rangle_\kappa + \frac{1}{Q-1}\right) \\ &= K\left\{\langle h_n^p, \kappa_{x_p}\rangle_\kappa + \frac{1}{Q-1} + \sum_{k\neq n}\left(\langle h_k^p, \kappa_{x_{\mathcal{I}(k)}}\rangle_\kappa + \frac{1}{Q-1}\right)\right\} \\ &= K\left\{h_n^p\left(x_p\right) + \frac{1}{Q-1} - \sum_{k=1}^{Q} b_k^p\right\} = K\left\{h_n^p\left(x_p\right) + \frac{1}{Q-1}\right\}.\end{aligned}$$

Puisque $h_n^p\left(x_p\right) > 0$, il est possible de minorer I_1 :

$$I_1 \geqslant \frac{K}{Q-1} - \frac{K^2}{2}\sum_{k=1}^{Q}\left\|\sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q} - \delta_{k,l}\right)\nu_{il}^p \kappa_{x_i}\right\|_\kappa^2 \tag{3.29}$$

avec $\mu^p = K\nu^p$. La maximisation du terme de droite donne la valeur K :

$$K = \left[(Q-1)\sum_{k=1}^{Q}\left\|\sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q} - \delta_{k,l}\right)\nu_{il}^p \kappa_{x_i}\right\|_\kappa^2\right]^{-1}. \tag{3.30}$$

En substituant (3.30) dans (3.29), nous obtenons l'inégalité suivante :

$$I_1 \geqslant \frac{1}{2}\left[(Q-1)^2\sum_{k=1}^{Q}\left\|\sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q} - \delta_{k,l}\right)\nu_{il}^p \kappa_{x_i}\right\|_\kappa^2\right]^{-1}.$$

Or, comme le montre [63], $\sum_{k=1}^{Q}\left\|\sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q} - \delta_{k,l}\right)\nu_{il}^p \kappa_{x_i}\right\|_\kappa^2$ peut se majorer par une quantité faisant intervenir D_m le diamètre de la plus petite boule de $\mathbf{H}_\kappa$ contenant l'ensemble $\{\kappa_{x_i} : 1 \leqslant i \leqslant m\}$. En effet, le développement de la double somme donne :

$$\begin{aligned}\sum_{k=1}^{Q}\left\|\sum_{i=1}^{m}\sum_{l=1}^{Q}\left(\frac{1}{Q} - \delta_{k,l}\right)\nu_{il}^p \kappa_{x_i}\right\|_\kappa^2 &= \frac{1}{Q^2}\sum_{k=1}^{Q}\left\|\sum_{l=1,l\neq k}^{Q}\left(\sum_{i=1}^{m}\nu_{il}^p\kappa_{x_i} - \sum_{i=1}^{m}\nu_{ik}^p\kappa_{x_i}\right)\right\|_\kappa^2 \\ &\leqslant \frac{(Q-1)^2}{Q}D_m^2.\end{aligned}$$

Cette dernière équation conclut cette partie de la démonstration en fournissant une minoration de I_1 :

$$I_1 \geqslant \frac{1}{2}\left[\frac{(Q-1)^4}{Q}D_m^2\right]^{-1}.$$

Synthèse :

L'utilisation conjointe de la minoration de I_1 et la valeur I_2 fournit

$$\frac{(Q-1)^4}{Q}\sum_{k=1}^{Q} S_{p,k}^2 {\alpha_{p,max}^*}^2 D_m^2 \geqslant 1,$$

et donc

$${\alpha_{p,max}^*}^2 \geqslant \frac{Q}{(Q-1)^4 \sum_{k=1}^{Q} S_{p,k}^2 D_m^2},$$

concluant la démonstration. □

Soit $\mathcal{M}(d_m)$ le sous-ensemble de d_m constitué des exemples mal classés pendant la procédure de validation croisée. Le cardinal de $\mathcal{M}(d_m)$ correspond donc au nombre d'erreurs de la procédure de validation croisée $\hat{Err}^{(cv,1)}$. Posons $S = \max_{p,k} S_{p,k}$. L'obtention d'une majoration de l'erreur de validation croisée leave-one-out se fait simplement en sommant sur les indices des exemples pour lesquels le classifieur se trompe, soit

$$\sum_p \max_{1\leqslant l\leqslant Q} {\alpha_{pl}^*}^2 \geqslant \sum_{p:\ (x_p,y_p)\in\mathcal{M}(d_m)} \frac{Q}{(Q-1)^4 \sum_{k=1}^{Q} S_{p,k}^2 D_m^2} \geqslant \frac{Q\hat{Err}^{(cv,1)}}{(Q-1)^4 Q S^2 D_m^2}.$$

Théorème 10 (Borne exacte multi-classe de type *Span bound*)**.** *Considérons une LLW-M-SVM à marge dure entraînée sur d_m. Soit $\hat{Err}^{(cv,1)}$ le nombre d'erreur de la procédure de validation croisée leave-one-out pour cette machine. Soit D_m le diamètre de la plus petite boule de $\mathbf{H}_\kappa$ contenant l'ensemble $\{\kappa_{x_i} : 1 \leqslant i \leqslant m\}$. Soit S la plus grande valeur du span pour tous les points relativement à toutes les catégories. Alors nous avons :*

$$\hat{Err}^{(cv,1)} \leqslant (Q-1)^4 S^2 D_m^2 \sum_p \max_{1\leqslant l\leqslant Q} {\alpha_{pl}^*}^2.$$

Dans [105], une méthode d'estimation de l'erreur de validation croisée leave-one-out est présentée. Elle s'appuie sur l'hypothèse que les vecteurs support ne changent pas pendant la procédure de validation croisée leave-one-out. Nous utiliserons aussi cette hypothèse et montrerons qu'elle ne permet pas d'obtenir une estimation précise de l'erreur. Cette constatation nous amènera à proposer une approximation de cette erreur en utilisant l'approche utilisée pour la LS-M-SVM.

Théorème 11. *Soit h^* et h^p les fonctions calculées par la LLW-M-SVM à marge dure respectivement sur l'ensemble d'apprentissage d_m et sur le même ensemble privé de l'exemple d'indice p. Soit $\alpha^* \in \mathbb{R}_+^{Qm}(d_m)$ le vecteur des variables duales associé à la LLW-M-SVM à marge dure entraînée sur d_m et soit $\alpha_{p,max}^* = \max_l \alpha_{pl}^*$. Soit $S_{p,k}$ le span de p relativement à la catégorie k. Si les vecteurs support restent inchangés pendant la procédure de validation croisée leave-one-out,*

alors pour tout exemple (x_p, y_p), l'égalité suivante est vraie :

$$\sum_{k=1}^{Q} \alpha_{pk}^* \left(h_k^p (x_p) - h_k^* (x_p) \right) = \sum_{k=1}^{Q} S_{p,k}^2 {\alpha_{p,max}^*}^2.$$

Démonstration. Cette démonstration reprend les notations de celle du lemme 8. Puisque les vecteurs support restent inchangés, nous pouvons choisir $\delta^p = \mu^p = \alpha^* - \alpha^p$ puisque $\alpha^* - \alpha^p$ vérifie (3.26). Nous avons alors l'égalité entre I_1 et I_2 puisque

$$I_1 = J(\alpha^*) - J(\alpha^p) = I_2. \tag{3.31}$$

Notons λ^* le vecteur de coefficients associés à $S_{p,k}^2$ et posons le vecteur λ tel que $\delta^p = -\lambda \alpha_{p,max}^*$ alors, d'après le terme de gauche de (3.27), nous obtenons

$$S_{p,k}^2 \geqslant \left\| \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\frac{1}{Q} - \delta_{k,l} \right) \lambda_{il} \kappa_{x_i} \right\|_\kappa^2.$$

De plus, d'après la définition de $S_{p,k}^2$, nous avons $S_{p,k}^2 \leqslant \left\| \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\frac{1}{Q} - \delta_{k,l} \right) \lambda_{il} \kappa_{x_i} \right\|_\kappa^2$ et donc l'égalité des deux quantités. L'expression de I_2 en fonction de $S_{p,k}^2$ est donc immédiate

$$I_2 = \frac{1}{2} \sum_{k=1}^{Q} S_{p,k}^2 {\alpha_{p,max}^*}^2 + \sum_{k=1}^{Q} \delta_{pk}^p \left(h_k^* (x_p) + \frac{1}{Q-1} \right).$$

Il en est de même pour I_1

$$\begin{aligned} I_1 &= J(\alpha^p + \delta^p) - J(\alpha^p) \\ &= -\frac{1}{2} \delta^{pT} H \delta^p + \nabla J(\alpha^p)^T \delta^p \\ &= -\frac{1}{2} \sum_{k=1}^{Q} S_{p,k}^2 {\alpha_{p,max}^*}^2 + \sum_{i=1}^{m} \sum_{k=1}^{Q} \delta_{ik}^p \left(h_k^p (x_i) + \frac{1}{Q-1} \right). \end{aligned}$$

Pour tous les exemples (autres que celui d'indice p), en utilisant l'hypothèse sur les vecteurs support, les conditions de Kuhn-Tucker imposent que $\delta_{ik}^p \left(h_k^p (x_i) + \frac{1}{Q-1} \right) = 0$, d'où

$$I_1 = -\frac{1}{2} \sum_{k=1}^{Q} S_{p,k}^2 {\alpha_{p,max}^*}^2 + \sum_{k=1}^{Q} \delta_{pk}^p \left(h_k^p (x_p) + \frac{1}{Q-1} \right).$$

En substituant cette relation dans (3.31), nous obtenons

$$\sum_{k=1}^{Q} \delta_{pk}^{p} \left(h_k^p \left(x_p \right) - h_k^* \left(x_p \right) \right) = \sum_{k=1}^{Q} S_{p,k}^2 {\alpha_{p,max}^*}^2.$$

Cette relation conclut la démonstration puisque pour tout k dans $[\![1, Q]\!]$, $\delta_{pk}^{p} = \alpha_{pk}^*$. □

Pour diagnostiquer une erreur de classification, il faudrait au moins $Q-1$ équations, or ce théorème n'en fournit qu'une seule.

3.3.2 Approximation de l'erreur de validation croisée leave-one-out

Comme nous l'avons vu avec le théorème 11, le raisonnement sur la borne Span ne permet pas d'obtenir une approximation de l'erreur de validation croisée leave-one-out. Afin d'obtenir une telle approximation, nous utilisons la similitude existant entre la M-SVM^2 et la LS-M-SVM. En effet, la section 3.1.3 présente une manière de calculer l'erreur de validation croisée leave-one-out pour la LS-M-SVM. En remplaçant les matrices E par $\mathcal{E}$, le théorème 9 devient :

Théorème 12 (Expression analytique de la fonction calculée par la M-SVM^2 lors de la procédure de validation croisée leave-one-out)**.** *Soit une M-SVM^2 à Q catégories entraînée sur d_m. Sous l'hypothèse que les vecteurs support ne changent pas pour la machine entraînée sur $d_m \setminus \{(x_i, y_i)\}$, la fonction calculée par cette dernière M-SVM^2 sur l'exemple x_i est le vecteur de $\mathbb{R}^{m_{\underline{i}}}$*

$$\left(h_j^i \left(x_i \right) \right)_{j \in \hat{i}} = -\frac{1}{Q-1} 1_{m_{\underline{i}}} - \left(\left(A_{\mathcal{E},\mathcal{E}}^{-1} \right)_{\underline{i},\underline{i}} \right)^{-1} \alpha_{\mathcal{E}\underline{i}} \tag{3.32}$$

avec :

- *$\underline{i}$: L'ensemble des indices de $\alpha_{\mathcal{E}}$ tels que les multiplicateurs de Lagrange correspondants soient associés à l'exemple x_i et que ces multiplicateurs soient non nuls.*
- *$\hat{i}$: L'ensemble des catégories pour lesquelles les multiplicateurs de Lagrange associés à l'exemple i sont non nuls.*

La démonstration de ce théorème est strictement identique à celle du théorème 9. La version bi-classe de ce théorème suffit à obtenir l'approximation de l'erreur leave-one-out mais cela n'est plus le cas ici. En effet, pour un exemple donné, ce théorème permet d'obtenir $m_{\hat{i}}$ sorties du classifieur avec $m_{\hat{i}} \leqslant Q-1$. Dans le cas bi-classe, nous avons une sortie si l'exemple est vecteur support et zéro dans le cas contraire. De plus, nous savons que si l'exemple n'est pas vecteur support, alors il n'intervient pas. Dans tous les cas, nous pouvons vérifier si la SVM fait une erreur grâce aux contraintes de bon classement.

Pour le cas multi-classe, plusieurs possibilités surviennent :

- $m_{\hat{i}} = 0$: l'exemple n'intervient pas,
- $m_{\hat{i}} = Q-1$: toutes les sorties du classifieur sont disponibles (théorème 12),
- $m_{\hat{i}} \in [\![1, Q-2]\!]$: On ne peut pas calculer toutes les sorties.

Afin de pouvoir obtenir une approximation de l'erreur de validation croisée leave-one-out, nous proposons deux approches. La première consiste à majorer l'erreur faite sous l'hypothèse d'invariance des vecteurs support, c'est-à-dire considérer que le classifieur se trompe dès qu'une des sorties du classifieur d'indice différent de y_p est positive .

Corollaire 3 (Majoration de l'erreur de validation croisée leave-one-out)**.** *Soit une M-SVM^2 à Q catégories entraînée sur d_m. Si les vecteurs support de cette M-SVM ne changent pas pendant la procédure de validation croisée leave-one-out, alors une majoration de l'erreur de cette procédure est :*

$$M_{sv}^{(cv,1)} = \sum_{i=1}^{m} \min\left(\sum_{k\in\hat{i}} \mathbb{1}_{h_k^i(x_i)>0}, 1\right)$$

avec

- $\underline{i}$ *: L'ensemble des indices de $\alpha_{\mathcal{E}}$ tels que les multiplicateurs de Lagrange correspondants soient associés à l'exemple x_i et que ces multiplicateurs soient non nuls.*
- $\hat{i}$ *: L'ensemble des catégories pour lesquelles les multiplicateurs de Lagrange associés à l'exemple i sont non nuls.*

Lorsque l'hypothèse d'invariance des vecteurs support n'est pas vérifiée, cette quantité n'est plus une majoration de l'erreur. Elle sera tout de même appelée majoration de l'approximation *de l'erreur de validation croisée leave-one-out au lieu d'*approximation de la majoration de l'approximation *de l'erreur de validation croisée leave-one-out et sera notée $\hat{M}_{sv}^{(cv,1)}$.*

La fonction $\min(\cdot, 1)$ est introduite afin de ne pas compter plusieurs fois le même exemple si plusieurs composantes sont positives.

Une autre approximation utile repose sur la constatation expérimentale que, pour certains noyaux, les valeurs de la fonction calculée par la M-SVM se concentrent sur les bords des marges analytiques. Nous faisons donc l'hypothèse que tous les exemples de $d_{\mathcal{I}}$ sont sur la marge, c'est-à-dire que les sorties du classifieur associées à ces exemples sont égales à $-\frac{1}{Q-1}$. Il est évident que cette hypothèse n'est jamais verifiée, mais nous avons constaté expérimentalement que cette approximation est pertinente.

Corollaire 4 (Approximation de l'erreur de validation croisée leave-one-out (appelée Approx.))**.** *Soit une M-SVM^2 à Q catégories entraînée sur d_m. Si les vecteurs support de cette M-SVM ne changent pas pendant la procédure de validation croisée leave-one-out et que les exemples d'indice dans $\mathcal{I}$ sont sur la marge, alors l'erreur de cette procédure est :*

$$Err_{sv}^{(cv,1)} = \sum_{i=1}^{m} \mathbb{1}_{\sum_{k\in\hat{i}} h_k^i(x_i) + (Q-1-m_i)\frac{1}{Q-1} < \max_{k\in\hat{i}} h_k^i(x_i)}$$

avec

- $\underline{i}$ *: L'ensemble des indices de $\alpha_{\mathcal{E}}$ tels que les multiplicateurs de Lagrange correspondants soient associés à l'exemple x_i et que ces multiplicateurs soient non nuls.*

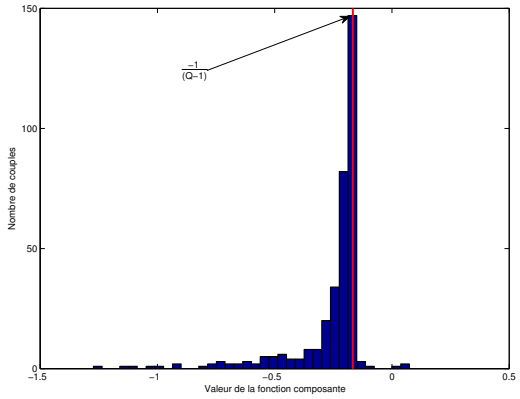


FIGURE 3.2 – Histogramme des valeurs du classifieur pour les couples de $\mathcal{I}$ sur le jeu de données *Image Segmentation*. Il y a 358 couples au total. La ligne rouge correspond à $-\frac{1}{Q-1}$.

- $\hat{\imath}$: *L'ensemble des catégories pour lesquelles les multiplicateurs de Lagrange associés à l'exemple i sont non nuls.*

En pratique, nous constatons que cette approximation est pertinente pour les noyaux stationnaires [51].

Définition 16 (Noyau stationnaire)**.** *Un noyau κ de $\mathcal{X}^2$ est stationnaire si il existe une fonction k de $\mathcal{X}$ telle que*

$$\forall \left(x, x'\right) \in \mathcal{X}^2,\ \kappa\left(x, x'\right) = k\left(x - x'\right).$$

Le noyau RBF en est un, et c'est celui que nous utilisons dans les expériences. La figure 3.2 présente un histogramme des valeurs du classifieur muni de ce noyau sur *Image segmentation*.

3.3.3 Intégration au chemin de régularisation

Ces deux approximations se basent sur le théorème 12, et leurs intégrations dans le parcours du chemin de régularisation s'effectuent par l'obtention de $\left(\left(A^{-1}_{\mathcal{E},\mathcal{E}}\right)_{\underline{i},\underline{i}}\right)^{-1}$ pour tous les $\underline{i}$ associés à i dans $[\![1, m]\!]$.

Un raisonnement identique à celui de la section 2.3.1 permet d'obtenir une expression analogue à (3.17) en remplaçant E par $\mathcal{E}$. En effet, nous obtenons

$$A^{-1}_{\mathcal{E},\mathcal{E}} = -H_{\mathcal{E},\mathcal{E}}\left(\lambda\right)^{-1} - \nu\left(D - C\nu\right)^{-1} C H_{\mathcal{E},\mathcal{E}}\left(\lambda\right)^{-1} \tag{3.33}$$

avec

- $D = \begin{pmatrix} 1_Q^T \\ 0_{Q-1,Q} \end{pmatrix} \in \mathcal{M}_{Q,Q}\left(\mathbb{R}\right)$,
- $C = \begin{pmatrix} 0_{m\mathcal{E}}^T \\ M_\Delta \end{pmatrix} \in \mathcal{M}_{Q,m_\mathcal{E}}\left(\mathbb{R}\right)$.

Cette matrice ne nous intéresse pas dans son intégralité, en effet seuls les blocs de sa diagonale sont utiles pour calculer les approximations de l'erreur. Comme la matrice $A_{\mathcal{E},\mathcal{E}}$ n'est pas symétrique, cette équation est moins facile à manipuler que l'équation (2.24), mais il est toutefois possible, moyennant quelques calculs supplémentaires, d'obtenir les quantités qui nous intéressent. Pour cela, nous posons $\Gamma \in \mathcal{M}_{m_\mathcal{E},m}$ tel que

$$C = (D - C\nu)\,\Gamma^T H_{\mathcal{E},\mathcal{E}}(\lambda)\,.$$

En supposant que $(D - C\nu)$ est de rang plein, l'unicité de Γ est assurée. La complexité pour obtenir Γ est en $O\left((Q-1)\,r^3 m_\mathcal{E}\right)$ puisque le système impliquant $H_{\mathcal{E},\mathcal{E}}(\lambda)$ est déjà partiellement résolu. [1] En adoptant la même notation que dans la section 3.1.3, nous notons $\nu_{\underline{i}}$ (respectivement $\Gamma_{\underline{i}}$) la sous matrice de ν (respectivement Γ) dont les lignes (respectivement colonnes) sont associées aux multiplicateurs non nuls de l'exemple i. Le terme $\left(A^{-1}_{\mathcal{E},\mathcal{E}}\right)_{\underline{i},\underline{i}}$ du théorème 12 s'écrit donc :

$$\left(A^{-1}_{\mathcal{E},\mathcal{E}}\right)_{\underline{i},\underline{i}} = -\left(H_{\mathcal{E},\mathcal{E}}(\lambda)^{-1}\right)_{\underline{i},\underline{i}} - \nu_{\underline{i}}\Gamma^T_{\underline{i}}.$$

En profitant du fait que la décomposition de Cholesky de $\left(I_{(Q-1)r} + \frac{1}{\lambda}R^T_\mathcal{E} D^{-1}_\mathcal{E} R_\mathcal{E}\right)$ est déjà calculée, et en utilisant la même démarche que pour obtenir (2.25), on calcule la sous-matrice en $O\left(m_\mathcal{E}\,(Q-1)^2\right)$ opérations.

3.4 Résultats expérimentaux

Cette section est dédiée aux résultats expérimentaux de la sélection de modèle pour la M-SVM². Tout d'abord, nous présentons les conditions expérimentales de l'évaluation. Ensuite, nous présentons le comportement relatif des différents critères de sélection de modèle que nous avons présentés. Finalement, nous comparons l'efficacité de ces critères. Cette section expérimentale ne traite pas de la LS-M-SVM, mais les expérimentations menées sur celle-ci nous montrent que les taux de reconnaissance obtenus sont similaires à ceux de la M-SVM². [2]

3.4.1 Conditions expérimentales

A l'image de ce que nous avons fait dans la section expérimentale 2.4, nous utilisons une méthode de Nÿstrom pour calculer la décomposition de l'approximation de la matrice de Gram. Les jeux de données sont présentés, avec leur paramétrage, dans le tableau 3.2. Le coefficient de régularisation balaie une plage allant de 10^4 à 10^{-4}. Tous les critères sont évalués pour chaque valeur de λ^l. L'algorithme de parcours du chemin de régularisation pour la M-SVM² est implanté en MATLAB (les temps de calcul ne sont pas donnés car le code n'a pas été optimisé). A notre

1. Des systèmes impliquant $H_{\mathcal{E},\mathcal{E}}(\lambda)$ ont déjà été résolu pour ν et ρ.

2. La littérature (par exemple [98]) annonce des taux similaires pour la ℓ_1-SVM, la ℓ_2-SVM et la LS-SVM. La LS-M-SVM étant construite comme une LLW-M-SVM des moindres carrés, ce résultat est donc prévisible, et par suite ne sera pas illustré ici.

connaissance, il n'existe pas de méthode automatique de sélection de modèle pour les M-SVM (y compris pour la M-SVM^2). Pour obtenir une performance de référence, nous utilisons une procédure de descente en gradient sur la borne Rayon-Marge multi-classe (voir annexe A). Pour cette minimisation, dans le cas du noyau gaussien, l'optimisation se fait aussi par rapport à la largeur de bande.

Jeu de données	# descripteurs	#apprentissage	#test	Normalisé
Banana	2	400	4900	Oui
Image Segmentation	19	210	2100	Oui
USPS_500	256	500	500	Non
Toy	2	1000	1000	Non
Iris	4	30	120	Oui

TABLE 3.2 – Caractéristiques des jeux de données utilisés pour l'évaluation des performances de la sélection de modèle par parcours du chemin de régularisation pour la M-SVM^2

L'utilisation du jeu de données *Banana* permet de vérifier l'identité entre la majoration de l'approximation et l'approximation dans le cas bi-classe. *Image Segmentation* et *Iris* sont issus de l'*UCI repository* alors que *USPS_ 500* (sous-ensemble de la base USPS réduite à 500 exemples) et *Toy* sont fournis avec MSVMpack [75].

3.4.2 Comparaison des critères de sélection de modèle

La figure 3.3 illustre le comportement des différents critères dédiés à la sélection de modèle présentés dans ce chapitre. Nous remarquons notamment que :

- la borne Rayon-Marge tend à choisir une valeur de C inférieure à celle correspondant au minimum de l'erreur de test,
- l'approximation valable pour les noyaux stationnaires est proche de l'erreur de test,
- les variations de la majoration de l'approximation sont semblables à celles de l'erreur de test.

3.4.3 Qualité de la sélection de modèle

Nous évaluons la qualité de ces critères en sélection automatique de modèle. Les tableaux 3.3 et 3.4 reportent les erreurs de test à l'optimum des critères présentés précédemment. Le tableau 3.3 illustre le comportement de ces critères pour une machine munie d'un noyau linéaire tandis que le tableau 3.4 présente les résultats pour un noyau gaussien et propose une comparaison à la borne Rayon-Marge. Le parcours du chemin de régularisation est parcouru à noyau fixé alors que l'on laisse à l'optimisation de la borne Rayon-Marge le choix de la largeur de bande. Un tel degré de liberté supplémentaire permet d'obtenir le meilleur de cet algorithme.

La première ligne du tableau confirme les résultats de la figure 3.3 : l'utilisation du parcours du chemin de régularisation multi-classe pour deux catégories est identique à celui du cas bi-classe. L'approximation de l'erreur (lors de l'utilisation d'un noyau stationnaire) reste très proche de l'erreur, aussi bien en tendance qu'en valeur. La majoration de l'approximation quant à elle

Jeu de données	Test	Majoration de l'approx.	Approximation	RM
Toy	9.1	9.1	10.6	9.9
Iris	3.3	3.3	13.3	13.3

TABLE 3.3 – Comparaison des erreurs de test au minimum de différents critères multi-classes (majoration de l'approximation, approximation et borne Rayon-Marge) pour le noyau linéaire.

Jeu de données	Chemin				Gradient		
	Test	Majoration de l'approx.	Approximation	RM	RM	C	σ
Banana	10.1	11.2	11.2	10.5	10.7	0.17	0.46
Image Segmentation	10.7	10.9	10.8	11.1	11.52	0.42	1.57
USPS_500	12.2	12.4	12.6	17.2	12.6	1.77	2.44
Toy	0.4	0.6	0.7	1.1	1.0	0.38	0.10

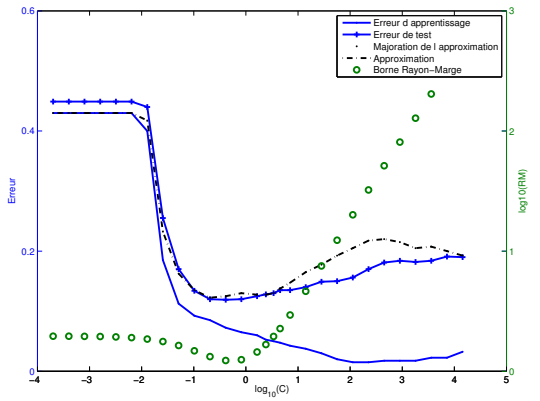
TABLE 3.4 – Comparaison des erreurs de test au minimum de différents critères multi-classes (majoration de l'approximation, approximation et borne Rayon-Marge) pour un noyau gaussien par parcours du chemin de régularisation et par une méthode de descente en gradient (voir annexe A).

suit l'erreur en tendance y compris pour les noyaux non stationnaires. Même si ses résultats sont un peu moins bons, la borne Rayon-Marge offre tout de même une alternative intéressante de par son coût de calcul très faible.

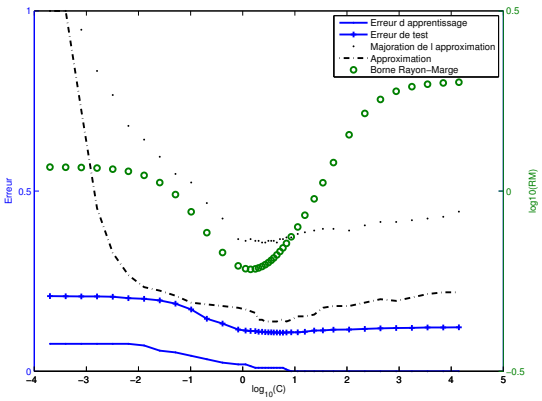
Ces constatations ainsi que l'analyse de la complexité du parcours de chemin et des calculs des approximations nous amène à proposer le diagramme 3.4 explicitant nos recommandations pour le choix de l'algorithme de sélection de modèle.

3.5 Conclusion

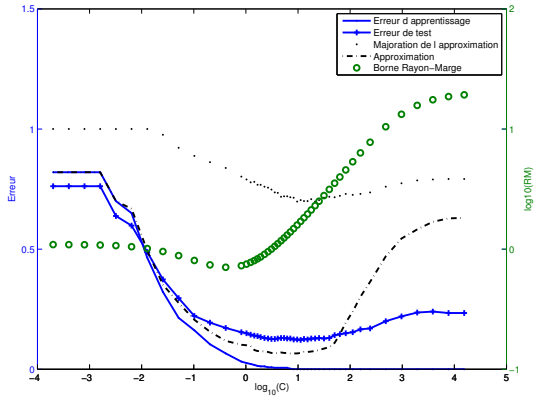
Ce chapitre contient des contributions originales de différentes natures. Dans un premier temps, nous avons introduit une nouvelle M-SVM, la M-SVM des moindres carrés. Ensuite, nous avons étendu l'algorithme de parcours du chemin de la ℓ_2-SVM à la M-SVM^2. Cette extension conserve les propriétés de l'algorithme bi-classe : les trajectoires des multiplicateurs de Lagrange ne sont pas linéaires par morceaux et la complexité peut être linéaire en le nombre d'exemples. De plus, nous avons introduit une nouvelle borne sur l'erreur de validation croisée leave-one-out pour la LLW-M-SVM (et donc pour la M-SVM^2) en s'inspirant de la borne Span. Enfin, nous avons aussi introduit deux approximations de l'erreur de validation croisée leave-one-out pour la M-SVM^2. Des résultats expérimentaux illustrent le bon comportement en pratique de ces approximations.



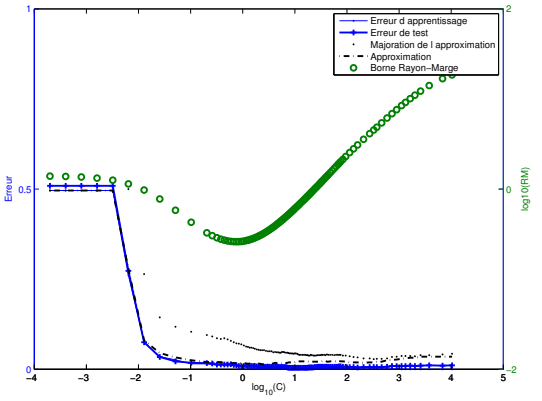
(a) : *Banana*



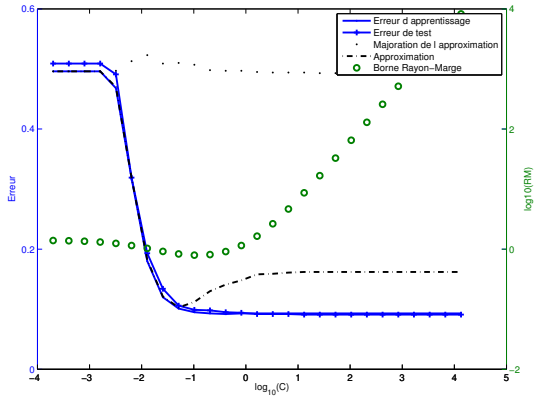
(b) *Image Segmentation*



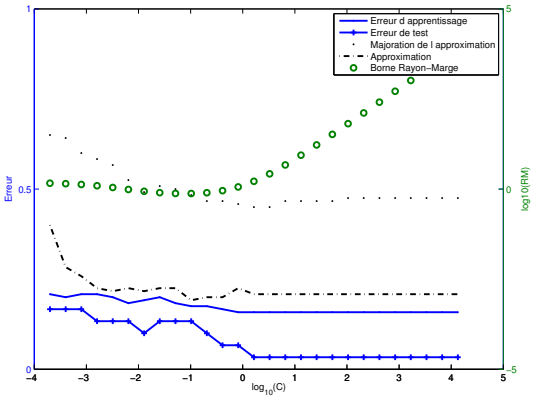
(c) : *USPS_ 500*



(d) : *Toy (gaussien)*



(e) : *Toy (linéaire)*



(e) : *Iris*

FIGURE 3.3 – Comparaison des critères de sélection de modèles pour la M-SVM^2 le long du chemin de régularisation. L'ordonnée de la borne Rayon-Marge est affichée sur une autre échelle en $\log_{10}(RM)$.

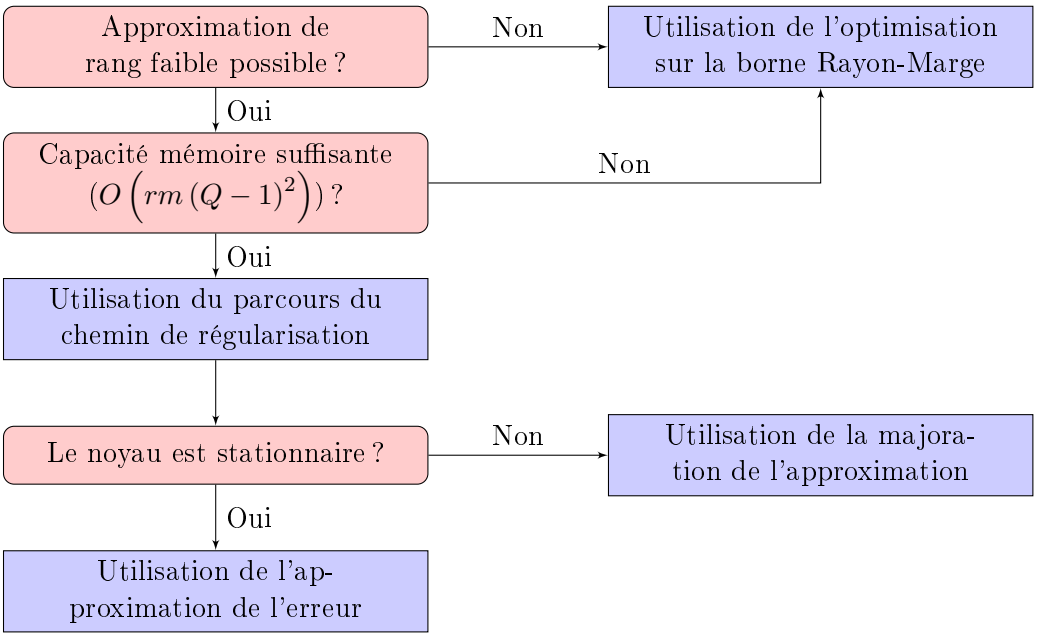


FIGURE 3.4 – Diagramme d'aide au choix de l'algorithme de sélection de modèle pour la M-SVM^2.

Chapitre 4

Sélection d'hyperparamètres par maximisation régularisée de l'alignement noyau/cible centré

« Calvin : You can't just turn on creativity like a faucet. You have to be in the right mood.
Hobbes : What mood is that ?
Calvin : Last-minute panic. »
—Calvin and Hobbes, Bill Watterson

Les méthodes à noyau ont été utilisées avec succès dans différents domaines de la reconnaissance des formes. Généralement, le choix du noyau est laissé à l'utilisateur, alors qu'il est pourtant crucial pour l'efficacité de ces méthodes. Dans le cas de la sélection de modèle par parcours du chemin de régularisation, le choix du noyau, effectué en amont, joue un rôle crucial puisqu'il doit être adapté au problème. Ce chapitre traite donc exclusivement des méthodes de sélection indépendantes de la suite du traitement. Ainsi, nous excluons toutes les méthodes évaluant l'adéquation du noyau à la tâche par un critère lié au taux de reconnaissance d'un système discriminant : les utilisations de la procédure de validation croisée, de bornes ou d'approximations (comme la méthode de Chapelle [27]) sortent du cadre de notre propos. Le chapitre commence par la présentation de l'alignement noyau/cible [35], puis se poursuit par une adaptation de cette méthode en utilisant l'alignement centré proposé dans [30]. Il s'achève sur des résultats expérimentaux obtenus sur un jeu de données jouet et sur des données du monde réel.

4.1 Alignement noyau/cible

4.1.1 Alignement noyau/cible standard

La sélection de noyau en amont permet d'effectuer la sélection de modèle de manière efficace en temps de calcul puisque le parcours du chemin de régularisation ne se fait qu'une seule fois. Les performances en test mises à part, la mise en œuvre du parcours du chemin doit guider le choix du noyau, à travers le rôle central joué par la valeur du rang de la matrice de Gram. Comme la section 2.4 l'a mis en évidence, le rang de la matrice de Gram dépend du noyau utilisé. Il en résulte qu'un choix approprié du noyau permet aussi bien de minimiser l'erreur de test que le temps de calcul.

L'alignement de noyaux a été présenté dans [35], puis proposé sous forme suivante dans [34] :

Définition 17 (Alignement noyau/cible [34]). *Soient κ et κ' deux fonctions de type positif sur $\mathcal{X}^2$ telles que $0 < \langle \kappa, \kappa \rangle_P < \infty$ et $0 < \langle \kappa', \kappa' \rangle_P < \infty$. L'*alignement *entre κ et κ' est*

$$A\left(\kappa, \kappa'\right) = \frac{\langle \kappa, \kappa' \rangle_P}{\sqrt{\langle \kappa, \kappa \rangle_P \langle \kappa', \kappa' \rangle_P}}$$

avec $\langle \kappa, \kappa' \rangle_P = \int_{\mathcal{X}^2} \kappa\left(x, x'\right) \kappa'\left(x, x'\right) dP_{\mathcal{X}}\left(x\right) dP_{\mathcal{X}}\left(x'\right)$ et $P_{\mathcal{X}}$ la distribution marginale de P sur $\mathcal{X}$.

Cette quantité permet de capturer la notion de bonne classification. La distribution des exemples étant inconnue, on lui substitue l'alignement empirique :

Définition 18 (Alignement empirique). *Soient κ et κ' deux fonctions de type positif sur $\mathcal{X}^2$ et soient K et K' leurs matrices de Gram associées à D_m vérifiant $\|K\|_F \neq 0$ et $\|K'\|_F \neq 0$. Alors,* l'alignement empirique *entre κ et κ' est défini par*

$$A_m\left(\kappa, \kappa'\right) = \frac{\langle K, K' \rangle_F}{\sqrt{\langle K, K \rangle_F \langle K', K' \rangle_F}}$$

avec $\langle K, K' \rangle_F = \sum_{i,j=1}^{m} \kappa\left(x_i, x_j\right) \kappa'\left(x_i, x_j\right)$ le produit de Frobenius de K et K'.

En plus d'être calculable à partir des données d'apprentissage, l'alignement empirique noyau/cible possède quelques propriétés intéressantes. Cristianini et ses coauteurs [34] ont montré qu'il était fortement concentré autour de son espérance : sa valeur empirique est stable en fonction du choix des ensembles d'apprentissage. Toujours d'après [34], il est possible de le relier à la "marge moyenne" et de montrer qu'un alignement optimal correspond à une séparation de marge maximale. Ces observations signifient qu'il est possible d'optimiser cette quantité afin d'obtenir une meilleure erreur en généralisation.

Le principe de la sélection de noyau par alignement noyau/cible consiste à choisir celui maximisant l'alignement empirique avec une cible. Dans [62], plusieurs cibles sont proposées pour le cadre multi-classe. Nous choisissons une cible (qui sera associé à la matrice cible K_t) de terme

général

$$\kappa_t\left(x,x'\right)=\left(-\frac{1}{Q-1}\right)^{1-\delta_{y,y'}}.$$

Les expériences réalisées ne mettent pas en évidence une variation significative des performances en fonction du choix de la cible.

4.1.2 Alignement noyau/cible centré

Une interprétation possible de l'alignement de noyaux est le cosinus d'un angle. Lorsque les données sont loin de l'origine de l'espace de représentation mais regroupées dans la même zone de cet espace, alors la valeur du produit scalaire (et donc l'alignement) est élevée. Sur ces données, la valeur de l'alignement est grande tandis que le pouvoir discriminant du noyau est faible. En conséquence, le centrage dans l'espace de représentation a été recommandé dans plusieurs articles [82, 86, 30].

Définition 19 (Noyau centré κ_c associé à κ [30])**.** *Soit κ une fonction de type positif sur $\mathcal{X}^2$ à valeurs réelles, alors le noyau centré κ_c associé à κ est défini pour tout $(x,x')\in\mathcal{X}^2$ par*

$$\begin{aligned}\kappa_c\left(x,x'\right)&=\langle\kappa_x-\mathbb{E}_X[\kappa_X],\kappa_{x'}-\mathbb{E}_{X'}[\kappa_{X'}]\rangle_\kappa\\&=\kappa\left(x,x'\right)-\mathbb{E}_X[\kappa\left(X,x'\right)]-\mathbb{E}_{X'}[\kappa\left(x,X'\right)]-\mathbb{E}_{(X,X')}[\kappa\left(X,X'\right)].\end{aligned}$$

La matrice de Gram associée au noyau centré κ_c est défini pour tout i et j dans $[\![1,m]\!]$ par :

$$[K_c]_{i,j}=K_{i,j}-\frac{1}{m}\sum_{k=1}^{m}K_{k,j}-\frac{1}{m}\sum_{k=1}^{m}K_{i,k}+\frac{1}{m^2}\sum_{k,l=1}^{m}K_{k,l}.$$

L'alignement noyau/cible centré se définit de la même manière que l'alignement standard :

Définition 20 (Alignement de noyaux centrés)**.** *Soient κ et κ' deux fonctions de type positif sur $\mathcal{X}^2$ telles que $0<\langle k_c,k_c\rangle_P<\infty$ et $0<\langle k'_c,k'_c\rangle_P<\infty$ avec k_c et k'_c les noyaux centrés respectifs. Alors,* l'alignement de noyaux centrés *entre κ et κ' est défini par*

$$\rho\left(\kappa,\kappa'\right)=A\left(\kappa_c,\kappa'_c\right)=\frac{\langle k_c,k'_c\rangle_P}{\sqrt{\langle k_c,k_c\rangle_P\langle k'_c,k'_c\rangle_P}}.$$

On peut définir l'alignement empirique de noyaux centrés à partir du m-échantillon D_m :

Définition 21 (Alignement empirique de noyaux centrés)**.** *Soient κ et κ' deux fonctions de type positif et soient K et K' leurs matrices de Gram associées à D_m vérifiant $\|K_c\|_F\neq 0$ et $\|K'_c\|_F\neq 0$. Alors,* l'alignement empirique de noyaux centrés *entre κ et κ' est défini par*

$$\rho_m\left(\kappa,\kappa'\right)=A_m\left(\kappa_c,\kappa'_c\right)=\frac{\langle K_c,K'_c\rangle_F}{\sqrt{\langle K_c,K_c\rangle_F\langle K'_c,K'_c\rangle_F}}.$$

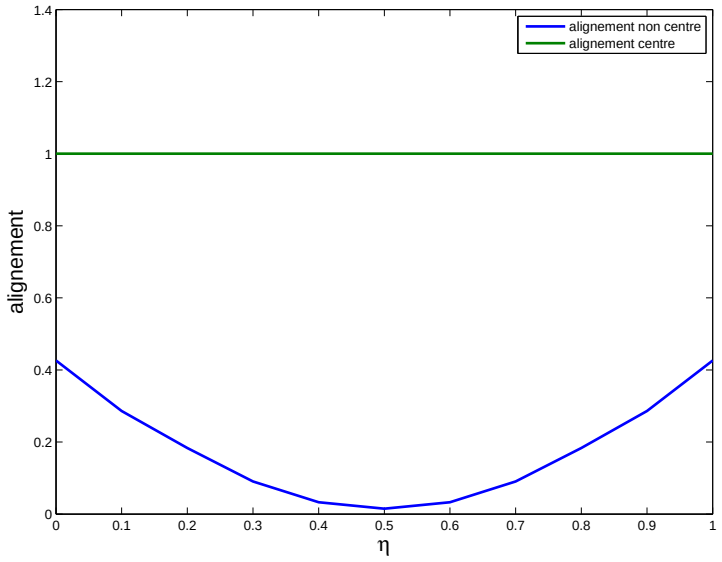


FIGURE 4.1 – Evolution de l'alignement empirique de noyau et de l'alignement empirique de noyau centré en fonction de η

Il est intéressant de remarquer que l'alignement empirique de noyaux centrés converge en probabilité vers l'alignement de noyaux centrés (voir le théorème 12 de [30]). Afin d'illustrer l'intérêt du centrage du noyau dans l'espace de représentation, nous choisissons une extension du problème de la Figure 1 de [30]. Cet exemple à trois catégories et en trois dimensions illustre les différences de comportement entre les deux alignements. La catégorie 1 est localisée en $(1, 0, 0)$ et contient $\frac{1}{3}$ des exemples. Les catégories 2 et 3 se situent respectivement en $(0, 1, 0)$ et $(0, 0, 1)$ et sont en proportion $\eta\frac{2}{3}$ (respectivement $(1-\eta)\frac{2}{3}$) avec $\eta \in [0, 1]$. Le noyau utilisé est $\kappa(x, x') = \langle x, x' \rangle + 1$. La figure 4.1 illustre que, sur cet exemple séparable et trivial, l'alignement de noyaux non centré ne présente pas un comportement optimal et que pour des classes équireparties l'alignement est nul.

4.2 Maximisation régularisée de l'alignement noyau/cible centré

Le choix du noyau peut se faire de multiples manières, par exemple par comparaison de l'alignement de différents noyaux, par combinaison linéaire de noyaux ou encore par paramétrisation d'une famille de noyaux. C'est cette dernière approche que nous considérons dans la suite.

Quel que soit la méthode utilisée pour maximiser l'alignement, il est nécessaire de recalculer des termes de la matrice de Gram à chaque pas. C'est pour cela que l'approche la plus couramment utilisée est la combinaison linéaire de noyaux [1] [34, 30]. Dans [30], il est montré que l'optimisation de l'alignement d'une telle combinaison se réduit à résoudre un problème de programmation

1. Le recalcul de la matrice de Gram dans ce cas précis correspond à une somme pondérée de matrices inchangées.

quadratique convexe. L'alignement d'un noyau gaussien paramétré présente l'inconvénient de devoir évaluer la fonction exponentielle pour chaque composante de la matrice de Gram à chaque itération. Malgré ce coût, l'alignement de ce noyau a déjà été utilisé avec succès dans plusieurs applications dont la prédiction de la structure secondaire des protéines [62].

L'alignement noyau/cible gaussien paramétré présente plusieurs avantages :

- Pour un noyau elliptique, il permet d'obtenir un sélection de variables *douce* directement interprétable par le praticien.
- La complexité de calcul des sorties du classfieur lors de l'evaluation en test reste faible : il n'y a qu'un seul noyau à évaluer.

Par la suite, bien que la méthode soit applicable à d'autres noyaux, nous ne nous intéressons qu'au noyau gaussien elliptique défini de la manière suivante :

Définition 22 (Noyau gaussien elliptique)**.** *Soient x et x' deux vecteurs de $\mathbb{R}^d$. Alors le noyau gaussien elliptique paramétrisé par $\mu = (\mu_l) \in \mathbb{R}^d$ est défini par*

$$\kappa_\mu\left(x, x'\right) = e^{-\sum_{l=1}^{d} \mu_l^2 \left(x_l - x'_l\right)^2}.$$

Intuitivement, la maximisation de l'alignement d'un noyau elliptique sur un petit jeu de données en haute dimension est sujette au sur-apprentissage. Pour pallier ce problème nous introduisons un terme de régularisation :

Problème 18 (Problème d'apprentissage d'un noyau gaussien elliptique par maximisation régularisée de l'alignement noyau/cible centré)**.** *Pour $Q \in \mathbb{N} \setminus \{0, 1\}$ et pour $(m, d) \in \mathbb{N}^{*2}$, soit $d_m = ((x_i, y_i))_{1 \leqslant i \leqslant m} \in \left(\mathbb{R}^d \times [\![1, Q]\!]\right)^m$, soit $\nu \in \mathbb{R}_+^*$ et soit κ_μ un noyau gaussien elliptique paramétré par $\mu \in \mathbb{R}^d$. L'apprentissage d'un noyau gaussien elliptique est obtenu par maximisation régularisée de l'alignement noyau/cible centré* [1] *:*

$$\mu_\nu^* = \operatorname*{argmax}_{\mu} J_a\left(\mu\right),$$

avec $J_a\left(\mu\right) = \rho_m\left(\kappa_\mu, \kappa_t\right) - \nu \|\mu\|_p^p$ et $p \in \{1, 2\}$.

Le choix $p = 1$ permet de chercher une solution parcimonieuse. Le terme de régularisation empêche de choisir une largeur de bande trop étroite, c'est-à-dire qu'il contrôle la vitesse de décroissance des valeurs propres [96, 114]. Indirectement, il permet donc de contrôler le rang qu'il faudra choisir pour approcher la matrice de Gram induite par le noyau [2].

La dérivation de J_a par rapport à μ s'obtient par dérivations successives des quantités impliquées dans cette fonctionelle. Soit K la matrice de Gram de d_m associée à κ_μ, soit K_c la matrice associée au noyau centré de κ_μ et soit K_{tc} la matrice associée à la cible centrée κ_t. En utilisant la notation habituelle (voir par exemple [54]), la dérivée d'une matrice par rapport à une variable

1. Par abus de langage, pour avoir une écriture pus légère, nous désignerons parfois cette méthode par le nom d'*alignement régularisé de noyaux centrés*.
2. Lorsque le critère de troncature porte sur la taille des valeurs propres.

est donnée par la dérivée de chaque terme de la matrice par rapport à cette variable. La dérivée de l'alignement centré par rapport à μ_k vaut

$$\frac{\partial \rho_m(\kappa_\mu, \kappa_t)}{\partial \mu_k} = \frac{\langle \frac{\partial K_c}{\partial \mu_k}, K_{tc} \rangle_F}{\|K_c\|_F \|K_{tc}\|_F} - \frac{\langle K_c, K_{tc} \rangle_F \langle K_c, \frac{\partial K_c}{\partial \mu_k} \rangle_F}{\|K_c\|_F^5 \|K_{tc}\|_F}. \tag{4.1}$$

La dérivée de K_c par rapport à μ_k est

$$\frac{\partial K_c}{\partial \mu_k} = \frac{\partial K}{\partial \mu_k} - \Gamma^T \frac{\partial K}{\partial \mu_k} - \frac{\partial K}{\partial \mu_k}\Gamma + \Gamma^T \frac{\partial K}{\partial \mu_k}\Gamma \tag{4.2}$$

avec $\Gamma = \frac{1}{m} 1_m 1_m^T$.

La dérivée de K par rapport à μ_k vaut donc

$$\frac{\partial K}{\partial \mu_k} = -2\mu D_k \odot K \tag{4.3}$$

avec $\odot$ le produit d'Hadamar et D_k la matrice des distances suivant la dimension k : $D_k = \left[(x_k - x'_k)^2\right]_{1 \leqslant i,j \leqslant m}$.

La complexité du calcul, en temps et en mémoire, du gradient de $\rho_m(\kappa_\mu, \kappa_t)$ par rapport à μ est $O\left(dm^2\right)$. Comme annoncé précédemment, cette complexité est beaucoup trop élevée pour que l'on puisse envisager une résolution classique du problème 18. Nous proposons de l'aborder par une variante de la descente en gradient stochastique à pas variable proposée dans [4]. Pour cela, à chaque itération, nous choisissons, suivant une loi uniforme, un petit nombre d'exemples qui serviront à calculer le gradient et appliquons l'équation (4.10) de [4] pour l'adaptation du pas de gradient. Il est important de noter que si la fonction objectif n'est pas convexe, elle ne présente toutefois qu'un seul maximum [1].

4.3 Résultats expérimentaux

Même si l'alignement de noyaux est bien fondé, il ne semble pas être entré dans l'usage des praticiens. Les résultats exposés dans ce chapitre montre en quoi cette méthode est prometteuse. Nous commençons par traiter un jeu de données jouet, puis des données du monde réel. Ces dernières données proviennent d'une part d'une étude effectuée avec l'ENSTA Bretagne et d'autre part d'une application de prédilection de l'équipe ABC : la prédiction de la structure secondaire des protéines globulaire. Les taux de reconnaissance annoncés sont obtenus avec une M-SVM^2 (implementation du paquet MSVMpack) dont les valeurs des hyperparamètres (dilatation du noyau et coefficient de régularisation) sont calculées par la sur-couche logicielle d'optimisation de la borne Rayon-Marge (présentée en annexe A). En effet, nous avons décidé de découpler le coefficient de dilatation du noyau et les pondérations des descripteurs, ce qui nous fait travailler

1. Penser à la graphe d'un cosinus sur une demi-période.

avec le noyau suivant :

$$\kappa_\mu\left(x, x'\right) = e^{-\mu_0 \sum_{l=1}^{d} \mu_l^2 \left(x_l - x'_l\right)^2}$$

avec μ_0 le coefficient de dilatation et $(\mu_i)_{i \in [\![1,d]\!]}$ les pondérations des descripteurs. Nous avons constaté expérimentalement que cette approche convergeait plus rapidement que l'enchaînement de l'optimisation des pondérations relatives seules puis de la recherche du coefficient de régularisation. Cette dernière approche fonctionne bien lorsque peu de pondérations sont cherchées, mais n'est pas efficace pour une valeur de d élevée [1].

4.3.1 Données jouet

Nous limitons l'étude des données jouet à un seul jeu : Image Segmentation de l'*UCI repository*[47]. Ce jeu de données en dimension 19 est particulièrement intéressant puisque chacun de ses descripteurs correspond à une quantité interprétable [2]. Chaque exemple est une vignette de trois sur trois pixels dont la catégorie correspond à la nature de ce qui est représenté. Les catégories sont : brique, ciel, feuillage, ciment, fenêtre, chemin et herbe.

Nous utilisons le découpage apprentissage-test proposé par défaut, c'est-à-dire 210 exemples pour faire l'apprentissage et la sélection de modèle et 2100 exemples pour le test. Le seul prétraitement effectué est le centrage et la réduction des données dans l'espace de description. L'intérêt de l'alignement de noyaux est évalué en deux étapes. La première consiste uniquement à quantifier l'apport en termes de taux de reconnaissance alors que la seconde s'intéresse à la sélection de variables.

4.3.1.1 Apport de l'alignement noyau/cible gaussien elliptique à la classification

Dans un premier temps, nous nous intéressons à l'apport du noyau gaussien elliptique dans la classification à l'aide de SVM. L'optimum de la borne Rayon-Marge multi-classe pour le noyau RBF standard (aussi dénomme sphérique par opposition à elliptique) est obtenu pour $\sigma = \frac{1}{\sqrt{2\mu_0}} = 1.48$ et $C = 0.47$. Le taux de reconnaissance avec ces hyperparamètres est de 88.43%, alors que Guermeur et Monfrini [63] obtiennent 90.20%. Une telle différence est probablement due à l'utilisation de la borne Rayon-Marge pour la sélection de modèle et non à une validation croisée. Ce taux de reconnaissance nous sert de référence. La minimisation de la borne Rayon-Marge multi-classe pour ce noyau est effectuée pour les configurations suivantes : maximisation non régularisée de l'alignement noyau/cible non centré, maximisation non régularisée de l'alignement noyau/cible centré, et ensuite les deux versions régularisées de cette méthode (la première de norme 1 ($p = 1$) et la seconde de norme 2 ($p = 2$)). Le coefficient de régularisation ν est fixé à 10^{-3}. Le tableau 4.1 présente les taux de reconnaissance obtenus. L'augmentation du taux de reconnaissance induite par l'utilisation du noyau elliptique apparaît statistiquement significative au sens du test de comparaison de deux pourcentages ("two sample proportion test" avec z=5.79)

1. On est confronté à un phénomène de vallée étroite.
2. Le 3-ieme descripteurs est constant puisqu'il s'agit du nombre de pixels de la mini-image.

Noyau	Régularisation	Taux de reconnaissance
RBF standard	-	88.48%
Gaussien elliptique non centré	sans	93.62 %
Gaussien elliptique centré	sans	94.19%
Gaussien elliptique centré	ℓ_1	94.19%
Gaussien elliptique centré	ℓ_2	94.05%

TABLE 4.1 – Taux de reconnaissance obtenus par minimisation de la borne Rayon-Marge multi-classe après alignement noyau/cible pour Image Segmentation

tandis que l'influence de la régularisation semble secondaire. Sur ce jeu de données, le centrage des données dans l'espace de représentation n'apparaît pas comme nécessaire [1].

Cette étude sur l'utilisation d'un noyau gaussien elliptique étant réalisée sur un seul jeu de données, elle est bien évidemment non représentative des gains que l'on peut atteindre en général. Cependant, la sélection de variables a largement prouvé son utilité par ailleurs (voir par exemple le très bon article [65]). Notre méthode peut aussi se voir comme la recherche d'une métrique adaptée au problème de classification considéré.

4.3.1.2 Etude comparative de la sélection de variables pour Image Segmentation

Puisque la mise en œuvre du principe de maximisation de l'alignement noyau/cible sur un noyau gaussien elliptique correspond à une sélection de variables douce, il est naturel de la comparer à des méthodes classiques effectuant cette tâche. En nous refusant à toute méthode de sélection liée au taux de reconnaissance, nous nous orientons vers les méthodes de type filtre ("filter" en anglais, voir par exemple la section 4.4 de [65]). Nous avons décidé de comparer notre méthode à une sous-famille de ces méthodes : celle basée sur l'information mutuelle (voir par exemple [17]).

Dans cette famille nous avons choisi quatre méthodes :

- Mutual Information Maximisation (MIM, [36]) : la première méthode développée qui consiste à choisir les descripteurs dont l'information mutuelle avec la classe est la plus importante. Le nombre de descripteurs est choisi a priori.
- Mutual Information Feature Selection (MIFS, [11]) : cette méthode choisit, de manière séquentielle, les descripteurs maximisant l'information mutuelle tout en limitant la redondance avec ceux déjà choisis. Le paramètre lié à la redondance est réglé par le praticien.
- Conditional Mutual Information Maximisation (CMIM, [46]) : cette méthode s'appuie sur l'information mutuelle conditionnelle.
- minimum-Redundancy Maximum-Relevance (mRMR, [84]) : cette méthode est très similaire à MIFS, mais le paramètre est fixé (et non choisi).

Afin de vérifier que ces méthodes fournissent des résultats satisfaisants, nous avons aussi utilisé une méthode de type "wrapper" de recherche séquentielle en avant (voir section 4.1 de

1. Pour la prédiction de la structure secondaire ne pas centrer peut conduire à l'obtention d'une pondération quasi-uniforme et donc de peut d'intérêt.

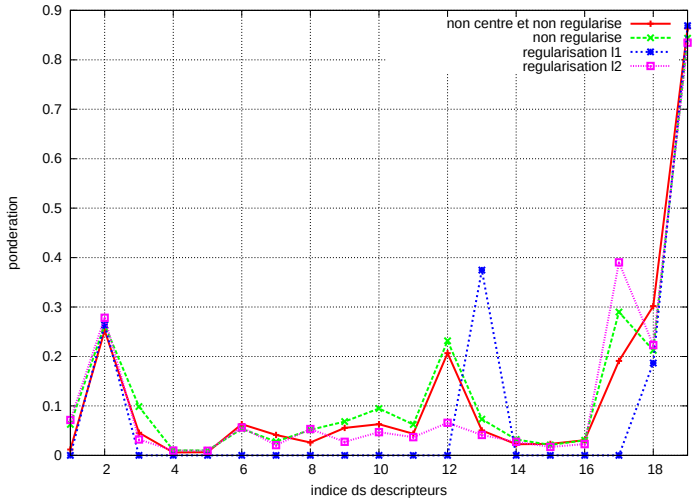


FIGURE 4.2 – Pondération obtenue pour Image Segmentation par maximisation de l'alignement noyau/cible, par maximisation de l'alignement noyau/cible centré et par maximisation régularisée de l'alignement noyau/cible centré (normes ℓ_1 et ℓ_2)

[65]).

La méthode d'alignement dotée d'une régularisation de norme ℓ_1 offre une pondération parcimonieuse (voir Figure 4.2) avec seulement 4 prédicteurs retenus. En nous basant sur ce résultat nous fixons le nombre de prédicteurs à 5 pour toutes les autres méthodes.

Le tableau 4.2 présente une comparaison des variables sélectionnées par les méthodes MIM, MIFS, CMIM, mRMR, notre méthode avec les normes ℓ_1 et ℓ_2 ainsi que la recherche séquentielle en avant utilisant une validation croisée à 5 pas [1]. Il est intéressant de noter que cette méthode s'arrête lorsqu'il trouve 6 variables.

La dernière ligne du tableau 4.2 contient les taux de reconnaissance obtenus par minimisation de la borne Rayon-Marge d'une M-SVM^2 (entraînée avec seulement 5 descripteurs). Les méthodes de type filtre fonctionnent globalement moins bien que les autres. Seuls mRMR, la méthode "wrapper" et les alignements de noyaux présentent des taux supérieurs à celui d'un apprentissage sur l'ensemble des descripteurs.

Intéressons-nous à présent aux variables sélectionnées par ces quatre méthodes. Le descripteur systématiquement sélectionné est le second : "ligne du pixel central". Ce choix, a priori surprenant, est en fait légitime, il permet de séparer ce qui se trouve en bas (herbe et chemin), en haut (ciel et feuillage) et au centre (ciment, fenêtre et brique) de l'image. Un autre descripteur qui revient souvent est la teinte de l'image. Cet attribut est particulièrement intéressant puisqu'il permet de couvrir toutes les couleurs par un seul scalaire. La méthode ne le sélectionnant pas choisit toutefois deux intensités de couleurs (rouge et bleu). La sélection réalisée par ces méthodes

1. Le couple (C, λ) est fixé à $(1.1, 1.48)$ pendant la procédure.

Descripteur :		Méthode						
#	Définition	ℓ_1	ℓ_2	MIM	MIFS	CMIM	mRMR	"wrapper"
1	Colonne du pixel central			2		3		
2	Ligne du pixel central	3	3	1	1	1	1	2
3	Nombre de pixels (=9)				2		2	
4	Nombre de lignes de faible contraste (longueur 5)				3			
5	Nombre de lignes de fort contraste (longueur 5)				4			6
6	Moyenne de la mesure de contraste entre pixels horizontalement adjacents							
7	Écart type de la mesure de contraste entre pixels horizontalement adjacents							
8	Moyenne de la mesure de contraste entre pixels verticalement adjacents							
9	Écart type de la mesure de contraste entre pixels verticalement adjacents							
10	Intensité moyenne sur la région $(R+V+B)/3$			4				
11	Moyenne du rouge sur la région					4		1
12	Moyenne du bleu sur la région		4	5		5		
13	Moyenne du vert sur la région	2					4	
14	Excès de rouge : 2R -V+B					2		5
15	Excès de bleu : 2B -V+R							
16	Excès de vert : 2V -B+R							4
17	Valeur moyenne de la région		2	3				
18	Saturation moyenne de la région	4	5		5		5	
19	Teinte moyenne de la région	1	1				3	3
	Taux de reconnaissance (%)	94.19	94.05	79.38	78.86	88.33	92.29	92.71

TABLE 4.2 – Comparaison des variables sélectionnées par méthodes de type filtre pour Image Segmentation

apparaît logique au vu du problème.

4.3.2 Données maritimes : inversion du front d'Ouesssant

L'application de la maximisation régularisée de l'alignement noyau/cible centré s'est faite dans le cadre d'un étude menée avec l'ENSTA Bretagne pour le projet "Passive Acoustics for Coastal Oceanography" (voir par exemple [23, 19]). Ce projet vise à fournir des méthodes de caractérisation des milieux marins côtiers par acoustique passive. Notre contribution réside dans le développement d'un noyau et d'une méthode adaptés à la classification des spectres enregistrés pour obtenir un résultat opérationnel.

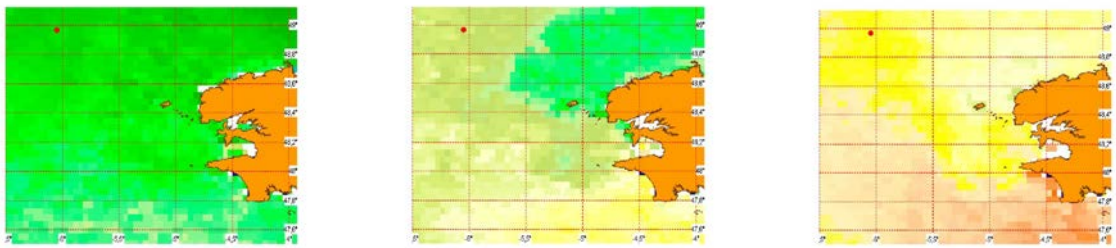
Contexte :

Le front d'Ouessant est un phénomène saisonnier qui a lieu de mai à octobre en mer d'Iroise. Il correspond à la frontière entre deux masses d'eau au large de l'île d'Ouessant :

- une masse d'eau côtière dont le profil de température est homogène (10-11°C),
- une autre masse d'eau, plus au large, dont le profil de température est stratifié (autour de 14°C en surface et 10-11°C au fond).

Dans la suite, on dira que le front est absent si le profil est homogène et qu'il est présent si le profil est hétérogène. La figure 4.3 présente l'évolution de la température de surface du 2 mai au 25 juin 2011. Au début de l'étude, le front est absent. Il s'installe progressivement jusqu'à être présent fin juin. Cette étude a pour objet de déterminer la présence du front d'Ouessant. Dans ce but, il est nécessaire de connaître précisément la localisation de la source sonore d'intérêt et d'installer un hydrophone équipé d'un système de mouillage silencieux. La source sonore utilisée dans cette étude est le rail d'Ouessant sur lequel passent en moyenne 165 navires par jour. L'hydrophone est situé au point A de la figure 4.4 tandis que le rail est représenté par les traits. Le choix d'un

hydrophone comme dispositif de mesure se justifie puisque les propriétés du fond marin et de la colonne d'eau influent sur la propagation du bruit rayonné par la voie maritime. Dans le cadre de l'étude, il est admis que seul le profil de la colonne d'eau change au cours du temps.



(a) : homogène (02/05/2010) (b) : transitoire (21/05/2010) (c) : stratifié (25/06/2012)

FIGURE 4.3 – Vue satellite de la température des eaux du front d'Ouessant. La couleur verte correspond à de l'eau dont le profil est homogène et la rouge à un profil stratifié.

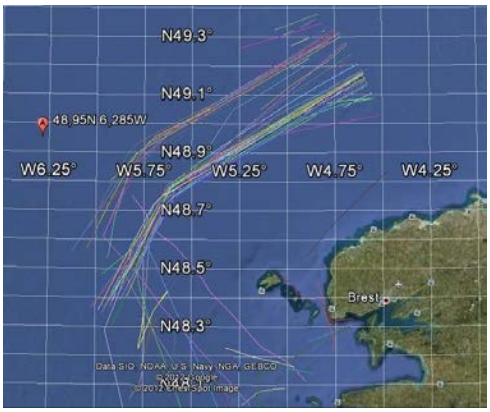


FIGURE 4.4 – Carte représentant les 24 heures de trafic maritime du 24 avril 2012. Le point A matérialise l'hydrophone.

Descriptions du jeu de données :

Chaque exemple correspond au spectre d'une heure de mesures. Ce spectre couvre une plage de 0 à $16KHz$ avec 2048 descripteurs (ce qui correspond à une résolution d'environ $7.8Hz$). Le jeu de données est décomposé en trois parties : une première correspondant à la présence du front (53 exemples), une deuxième à son absence (48 exemples), et une dernière à des spectres pour lesquels l'affectation à l'une des deux catégories n'est pas directe (25 exemples). L'étiquetage était réalisé à partir d'images satellite (dont sont tirées les images de la figure 4.3).

Résultats :

Le fait d'introduire une classe de données correspondant à une transition du phénomène plutôt que des données non-étiquetées, nous a conduit à ne pas utiliser une méthode d'apprentissage semi-supervisé. Nous procédons à la maximisation régularisée de l'alignement noyau/cible centré suivi par le parcours du chemin de régularisation. La figure 4.5 présente la pondération obtenue

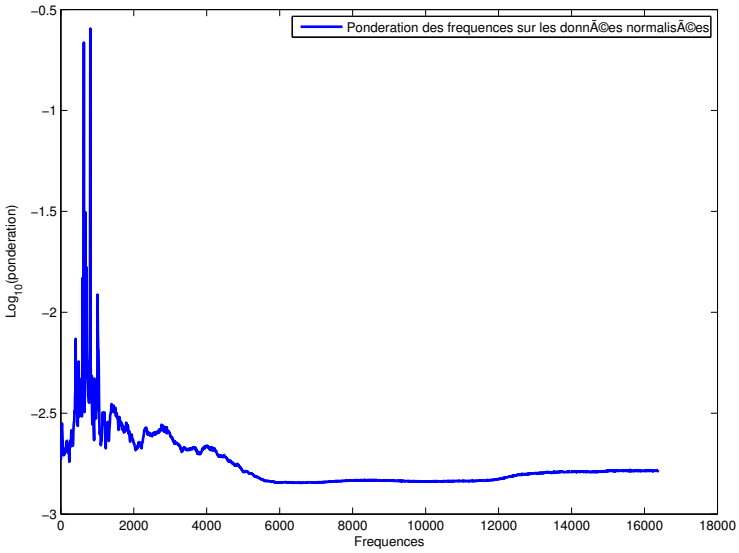


FIGURE 4.5 – Pondération des fréquences de l'hydrophone obtenue par maximisation régularisée de l'alignement noyau/cible centré.

par l'alignement de noyaux. La méthode attribue de fortes pondérations à la plage de fréquences comprise entre $500Hz$ et $1000Hz$. Ce résultat est en accord avec la théorie [23] puisque les basses fréquences ($500Hz$ à $1000Hz$) correspondent aux fréquences rayonnées par les navires tandis les très basses fréquences ($0Hz$ à $500Hz$) sont liées au bruit de mouillage de l'hydrophone. Pour cette pondération, l'erreur de validation croisée est d'environ 3%. Les sorties du classifieur sont satisfaisantes puisqu'elles fournissent des résultats cohérents avec les mesures satellites de températures de surface. De plus, ces résultats ont permis de mettre en évidence une propriété du profil stratifié : le son interagit plus avec le plancher océanique et s'atténue plus rapidement dans une bande de fréquences donnée. Ce résultat a été mis en évidence par simulation dans une autre étude [19]. Une autre campagne de mesures est prévue afin de confirmer cette première étude. En effet, il est nécessaire de s'assurer que c'est bien le front d'Ouessant qui justifie cette baisse d'intensité à ces fréquences. Toutefois, la possibilité d'interprétation de l'alignement de noyaux combinée au bon taux de reconnaissance sont très prometteurs pour une installation opérationnelle de ce système de monitorage.

4.3.3 Données protéiques : prédiction de la structure secondaire des protéines

Dans le cadre d'une étude exposée dans un article figurant dans l'annexe B, l'utilisation de la maximisation régularisée de l'alignement noyau/cible centré s'est révélée nécessaire puisque l'alignement noyau/cible non-centré ne permet pas d'obtenir un noyau satisfaisant.

Contexte :

Connaître la structure d'une protéine est un prérequis pour comprendre précisément sa fonction.

On distingue différentes manières de décrire cette structure. Tout d'abord, la structure primaire qui correspond à sa séquence. Cette séquence peut être écrite dans un alphabet de 20 symboles : les acides aminés. Le terme "résidu" est utilisé pour décrire ce qui reste de ces acides aminés dans une protéine. Ensuite, nous avons sa structure tertiaire qui désigne sa structure 3D, c'est-à-dire sa conformation spatiale. C'est cette dernière conformation qui aide le biologiste à inférer la fonction de la protéine. Toutefois, la détermination expérimentale de cette structure est complexe et n'est pas toujours possible. Il en résulte un très grand écart entre le nombre de séquences connues et celui des structures connues ($21.2\ 10^6$ contre $82\ 10^3$ en juin 2012). Pour combler cet écart, on fait appel à la prédiction. Celle-ci s'appuie ordinairement sur une approche de type *diviser pour régner*. C'est essentiellement dans ce contexte qu'intervient la prédiction de la structure secondaire. Cette structure correspond à une projection de la structure tridimentionelle sur une unique dimension, c'est-à-dire une succession d'états conformationnels associés à chaque résidu. Il existe deux principales structures régulières périodiques qui sont l'hélice α et le brin β. Le reste correspond à des structures apériodiques appelées boucles (ou *coil* en anglais). L'hélice est la structure la plus représentée puisqu'elle correspond à environ 30% des résidus. De plus, sa période est de 3.6 résidus. Le brin concerne environ 20% des résidus.

Cette prédiction de la structure de la structure secondaire est une tâche de reconnaissance des formes. En effet, elle correspond à une problème de discrimination à trois catégories consistant à affecter à chaque résidu d'une séquence protéique son état conformationnel : hélice α, brin β ou structure apériodique. Toutefois, plutôt que d'utiliser directement la séquence pour effectuer la classification, on préfère utiliser des alignements multiples qui permettent d'incorporer de l'information d'évolution et de fournir ainsi de meilleurs taux. La description des résidus d'une séquence de protéine correspond alors à une matrice appelée PSSM (Position-Specific Scoring Matrix) obtenue par PSI-BLAST [5]. L'utilisation de tels descripteurs a permis une augmentation significative du taux de reconnaissance (voir par exemple [69]).

Nous nous intéressons à une approche locale de la prédiction. L'utilisation d'une fenêtre d'analyse est une des méthode les plus intuitives pour la mettre en œuvre (voir par exemple [37]). Toutefois, nous souhaitons de plus obtenir l'importance relative des positions dans cette fenêtre d'analyse. De plus, ces pondérations permettent de surmonter le problème du choix de la taille de la fenêtre, à conditions que la fenêtre d'origine soit assez grande, en attribuant des faibles poids aux extrémités. Plusieurs méthodes de recherche de la pondération de la fenêtre d'analyse ont été proposées (voir [62, 55, 50]). Toutefois, aucune ne permet de calculer les pondérations à partir des PSSM. Nous proposons d'appliquer l'alignement noyau/cible centré à cette problématique en nous inspirant du noyau spécifique développé dans [62]. L'approche d'alignement noyau/cible développée dans [62] s'applique aux séquences seules en introduisant un produit scalaire entre acides aminés et une pondération de la fenêtre d'analyse. Une fois ce produit scalaire "enlevé", ce noyau correspond au noyau gaussien elliptique faisant le coeur de ce chapitre. L'apprentissage des paramètres de ce noyau sur les PSSM par un alignement noyau/cible ne fournit qu'une

pondération uniforme triviale [1] ce qui appelé la mise en œuvre d'une approche alternative.

Les données :

Les données (PSSM) appartiennent à $\mathbb{R}^{260}$: chaque description est fournie par le contenu d'une fenêtre de taille 13 dont chaque position est codé sur 20 valeurs réelles. Nous souhaitons obtenir la valeur des pondérations des positions dans la fenêtre. La base de données contient plus de 80000 résidus.

Résultats :

La maximisation régularisée de l'alignement noyau/cible centré fournit des pondérations très satisfaisantes. La figure 4.6 les présente ainsi que celles disponibles dans la littérature. La méthode de [50] repose sur une approche statistique, celle de [55] sur les poids virtuels dans un réseau de neurones et celle de [62] sur l'alignement de noyaux sur des séquences seules. Notre méthode est la première à fonctionner directement sur les alignements multiples. On remarque le maximum au centre et légèrement plus de poids du côté de l'extrémité C-terminale de la séquence.

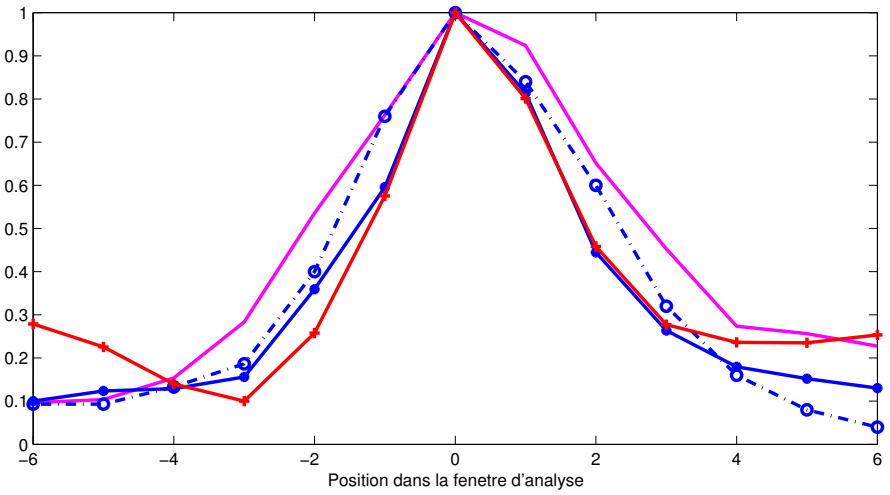


FIGURE 4.6 – Pondération des positions de la fenêtre d'analyse pour la prédiction de la structure secondaire de protéines obtenues par différentes méthodes : en magenta [50], en bleu et en continu [55], en bleu et en pointillés [62] et en rouge la maximisation de l'alignement noyau/cible centré. Les courbes sont remises à l'échelle par une transformation affine pour faciliter la lisibilité.

Cette pondération a été utilisée dans l'article présenté en annexe B pour le calcul des probabilités a posteriori d'appartenance aux différentes classes.

Résultat complémentaire :

Une application intéressante de l'alignement noyau/cible centré est son utilisation pour calculer un noyau adapté à une classe. En effet, cela permet de trouver les descripteurs qui sont discriminants pour cette classe. Il apparaît naturel de prendre une approche bi-classe un-contre-tous

1. Les données sont devenues trop éloignées de l'origine de l'espace de représentation, posant des problèmes pour l'algorithme de descente en gradient.

pour l'apprentissage des coefficients. Pour réaliser cette tâche pour la classe $k \in [\![1, Q]\!]$, il suffit de choisir la cible définie par

$$\kappa_{t,k}\left(x, x'\right) = (-1)^{\delta_{y,k}\left(1-\delta_{y',k}\right)+\delta_{y',k}\left(1-\delta_{y,k}\right)} .$$

La figure 4.7 présente les pondérations obtenues par classe. Ces résultats concordent avec ceux obtenus dans [50, 55].

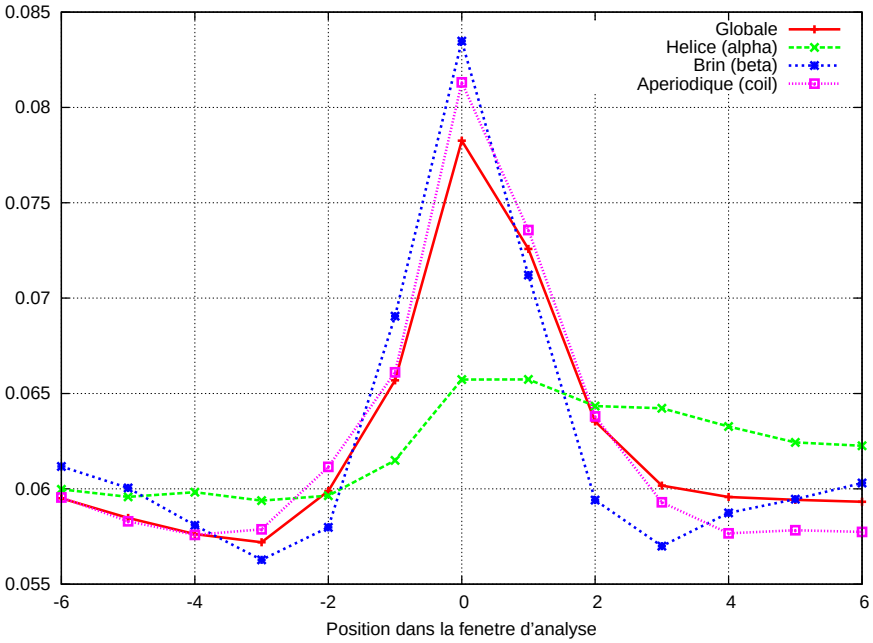


FIGURE 4.7 – Pondération des positions de la fenêtre d'analyse par alignement noyau/cible centré par classe pour la prédiction de la structure secondaire des protéines.

4.4 Conclusion

Ce chapitre présente une nouvelle méthode de d'alignement noyau/cible. La sélection de noyau par maximisation de l'alignement est tout d'abord présentée avant d'aborder le centrage dans l'espace de représentation. Nous avons ensuite proposé une méthode de maximisation de l'alignement noyau cible/centré et d'un terme de régularisation. En plus d'une mise en œuvre sur un jeu de données jouet, nous avons présenté deux applications de la méthode à des données du monde réel. Les expériences sur les données jouet ont montré que cette approche était compétitive autant pour l'amélioration de la classification que pour la sélection de variables. L'alignement de noyau sur les données de l'ENSTA Bretagne a permis d'obtenir un noyau adapté à ce problème. La plage de fréquences retenue est en accord avec les résultats théoriques sur la propagation des ondes

sonores en milieu sous-marin. De plus, notre méthode est la première à permettre l'apprentissage d'un noyau dédié à la prédiction de la structure secondaire des protéines directement exploitant des PSSM.

Conclusion générale et perspectives

Conclusion

Dans ce manuscrit, nous avons proposé des méthodes de sélection de modèle dédiées aux SVM à coût quadratique. Elles combinent des techniques de parcours du chemin de régularisation et de nouveaux critères de sélection. La non-linéarité de la trajectoire des multiplicateurs de Lagrange a requis la conception de solutions originales pour la mise au point des algorithmes de parcours du chemin de régularisation de la ℓ_2-SVM et de la M-SVM^2. Des critères de sélection de modèle pour ces deux machines sont obtenus par extension de résultats de la littérature comme la borne Span, la borne Rayon-Marge multi-classe et des approximations de l'erreur en généralisation. Nous résumons ces contributions avant de discuter des perspectives.

Dans le chapitre 2, nous avons proposé une méthode de parcours du chemin de régularisation pour la ℓ_2-SVM. Une fois l'ensemble des vecteurs support déterminé, les valeurs des multiplicateurs de Lagrange s'obtiennent comme solution d'un système linéaire. La résolution de ce système est particulièrement efficace lorsque la matrice de Gram est de rang faible. Nous avons ensuite présenté une heuristique permettant d'identifier les valeurs de λ pour lesquelles il est important de calculer les changements d'ensemble. Nous proposons l'intégration d'un critère de sélection de modèle qui est une approximation de l'erreur de validation croisée leave-one-out. Son calcul s'effectue pour un coût réduit dans la mesure où il s'appuie sur ceux du parcours du chemin. En conséquence du rang faible de la matrice de Gram (qu'il s'agisse de la matrice d'origine ou d'une approximation choisie de manière adéquate), le calcul du chemin de régularisation et celui du critère de sélection ont une complexité linéaire en le nombre d'exemples. Nous avons également proposé deux approximations de l'erreur en généralisation, l'une relative à la procédure de validation croisée à V pas, l'autre au bootstrap .632+, afin de fournir des critères de sélection associés à des estimateurs de variance plus faible. Les résultats expérimentaux obtenus à ce jour ne permettent pas de trancher entre les différents critères.

Au chapitre 3, nous avons étendu l'algorithme de parcours du chemin de la ℓ_2-SVM à la M-SVM^2. Pour ce faire, nous avons introduit une nouvelle M-SVM des moindres carrés, la LS-M-SVM, qui est à la M-SVM^2 ce que la LS-SVM est à la ℓ_2-SVM. Parmi les principales difficultés posées par l'extension de l'algorithme, on note le fait que dans le cas multi-classe, les variables d'écart ne sont plus contraintes en signe et la notion de vecteur support doit être liée à la classe. Nous avons aussi étendu la borne Span à la LLW-M-SVM et introduit deux approximations de

l'erreur de la procédure de validation croisée leave-one-out. Comme dans le cas bi-classe, ces approximations utilisent l'hypothèse d'invariance des vecteurs support. Cependant, l'obtention de résultats satisfaisants nécessite l'introduction d'hypothèses additionnelles. Les expériences établissent la pertinence des solutions que nous avons développées au sens où les erreurs en test correspondantes s'avèrent très proches du minimum de l'erreur sur le chemin. Cette méthode de sélection de modèle par parcours du chemin de régularisation est ici encore particulièrement efficace pour les problèmes où la matrice de Gram est de rang faible. Pour les autres cas, l'espace mémoire peut constituer un facteur limitant. Nous recommandons alors l'optimisation de la borne Rayon-Marge, pour laquelle une solution efficace est fournie par la descente en gradient présentée dans l'annexe A.

Le dernier chapitre de la thèse porte sur l'autre aspect de la sélection de modèle pour les SVM : le choix du noyau. Ce sujet est naturellement intimement liée au précédent dans la mesure où un choix approprié du noyau est un prérequis pour une utilisation efficace du parcours du chemin de régularisation. Nous avons proposé une méthode permettant de choisir un noyau indépendamment de la machine. L'approche utilisée est celle de la maximisation de l'alignement du noyau d'intérêt (après centrage) et d'un noyau cible. Nous l'avons évaluée dans le cadre de la détermination des valeurs des paramètres d'un noyau gaussien elliptique. Les résultats expérimentaux, obtenus sur des données jouet et des données issues de problèmes du monde réel, valident la méthode. Elle produit conjointement un augmentation du taux de reconnaissance et une sélection de variables douce.

Pendant la finalisation de ce manuscrit, notre attention a été attirée sur un article présentant une approche que l'on peut considérer comme étant la duale de la nôtre. Alors que la méthode de parcours du chemin de régularisation présentée ici se fonde sur une résolution exacte d'une approximation du problème, à l'aide de l'approximation de la matrice de Gram, celle de [52] utilise une résolution approchée du problème exact. La détection des transitions du chemin de régularisation, dans cette approche, utilise des critères de précision plutôt que des critères de changement d'ensemble.

Perspectives

L'une des principales contributions de cette thèse est la dérivation d'estimations de l'erreur en généralisation des SVM à coût quadratique. Une perspective naturelle est leur extension à d'autres machines à noyau à coût quadratique. Celle qui s'impose d'abord à l'esprit est une extension de la Kernel Projection Machine (KPM) [14]. Le lien central avec nos travaux réside dans la méthode de construction de l'approximation de la matrice de Gram. Précisément, nous pensons spécifier une variante de la KPM multi-classe [101] pour laquelle la sélection de modèle se fonde sur notre approche de l'estimation de l'erreur. Une autre de nos contributions, la LS-M-SVM, appelle un développement naturel, son extension au cadre de l'apprentissage "multi-label". Ce développement pourrait suivre l'approche proposée dans [74]. De plus, le calcul de l'erreur de la

procédure de validation croisée qui est proposé dans la section 3.1.3 est directement transposable à ce cadre. Cette adaptation consiste en effet à abandonner la contrainte de sommation à zéro des fonctions composantes tout en imposant des contraintes de bon classement pour chaque couple exemple-label. Nous obtenons donc une formulation plus simple de la LS-M-SVM qui reste en forme fermée, nous assurant la possibilité d'utiliser la même technique de calcul de l'erreur.

Un second point intéressant soulevé par cette thèse concerne l'approximation de la matrice de Gram. En effet, sa construction nous assure qu'elle est définie positive, que le noyau le soit ou pas. Elle doit donc permettre de relâcher les contraintes sur le noyau, ce dont nous avons tiré parti en considérant l'emploi d'un noyau conditionnellement défini positif : le noyau "thinplate" étudié dans [107]. Il est presque invariant par dilatation, une propriété particulièrement intéressante pour l'apprentissage puisqu'une dilatation des données peut être corrigée par une modification de la valeur du coefficient de régularisation. On en mesure ainsi l'avantage dans le cadre du parcours du chemin de régularisation. L'approximation du noyau proposée dans le manuscrit (voir la section 2.2.6.1) est telle que l'apprentissage n'est pas affecté négativement par ce choix, tandis que les performances s'avèrent à peine inférieures à celles du noyau RBF (voir Table 2.5). Nous souhaitons approfondir notre compréhension de ce phénomène. Toujours dans le cadre de l'ingénierie du noyau, nous projetons de nous inspirer de [37], et plus particulièrement des travaux sur les fenêtres glissantes récurrentes [9], afin de développer un noyau "récurrent" dédié au données séquentielles. Sur un exemple synthétique, l'intégration des prédictions dans la fenêtre glissante a permis d'obtenir une augmentation significative du taux de reconnaissance. Cette approche soulève un certain nombre de questions théoriques auxquelles il conviendra de répondre avant de pouvoir effectuer une sélection de modèle.

Le dernier axe de ces perspectives, mais non le moindre, concerne le développement d'une M-SVM dont chaque fonction composante est associée à un noyau différent. L'objectif est naturellement d'exploiter au mieux les spécificités des distributions des différentes catégories. Les espoirs que nous plaçons dans une telle machine sont étayés par les bons résultats fournis par l'alignement régularisé de noyaux centrés pour l'apprentissage de noyaux inspirés par la méthode de décomposition "un-contre-tous" (voir le résultat complémentaire de la section 4.3.3). L'expression analytique de la classe de fonctions sous-jacente est $(\mathbf{H}_{\kappa_1} \oplus 1) \times \ldots \times \big(\mathbf{H}_{\kappa_Q} \oplus 1\big)$, où κ_k est le noyau associé à la catégorie k.

Annexe A

Optimisation de la borne Rayon-Marge multi-classe pour la sélection de modèle de la M-SVM^2

L'optimisation de la borne Rayon-Marge de la ℓ_2-SVM est une méthode de sélection de modèle reconnue comme efficace (voir par exemple dans [27, 71, 39, 24]). Nous proposons dans cette annexe une méthode permettant de sélectionner les hyperparamètres de la M-SVM^2 en utilisant la borne Rayon-Marge multi-classe développée dans [63]. La majeure partie de l'annexe est dédiée aux calculs des dérivées de cette borne. Ces calculs sont fortement inspiré de [27]. Nous concluons sur une discussion liée à la mise en œuvre de cette méthode.

L'utilisation directe du théorème 5 ne permet pas le calcul efficace des dérivées de la borne. La réécriture de cette borne en utilisant (1.14), permet d'obtenir :

$$\hat{Err}^{(cv,1)} \leqslant \frac{(Q-1)^4}{Q} \alpha(\lambda)^T H(\lambda) \alpha(\lambda) D_m^2.$$

La présence de la forme quadratique va nous permettre une simplification très intéressante.

Toujours dans une optique de faciliter les calculs, nous posons :

$$\begin{cases} w = \alpha(\lambda)^T H(\lambda) \alpha(\lambda) \\ c = \ln(1/\lambda) \\ s = \ln(\sigma). \end{cases}$$

L'utilisation du logarithme est une astuce couramment utilisée pour simplifier les calculs de dérivées (voir par exemple dans [27, 71]). Toutefois c et s ne seront utilisés que plus tard puisque nous travaillerons avec un hyperparamètre quelconque θ dont dépend κ_λ.[1] Par simplicité, nous

1. Ne pas oublier que la borne Rayon-Marge n'est valable que pour les machines à marge dure. Si l'on travaille avec une M-SVM^2, le "noyau" à utiliser dans la borne Rayon-Marge est κ_λ.

omettons le coefficient $\frac{(Q-1)^4}{Q}$ dans les calculs. Nous cherchons donc à évaluer la dérivée de $R^2 w$, c'est-à-dire :

$$\frac{\partial R^2 w}{\partial \theta} = \frac{\partial R^2}{\partial \theta} w + R^2 \frac{\partial w}{\partial \theta}.$$

A.1 Dérivée du rayon

Le calcul de la plus petite sphère contenant l'ensemble des exemples d'apprentissage s'obtient en résolvant le problème de programmation quadratique suivant (voir section 10.7 de [104]) :

Problème 19 (Rayon de la plus petite sphère contenant l'ensemble des exemples d'apprentissage)**.**

$$R^2 = \max_{\beta} \sum_{i=i}^{m} \beta_i \kappa_\lambda (x_i, x_i) - \sum_{i,j=1}^{m} \beta_i \beta_j \kappa_\lambda (x_i, x_j)$$

$$s.c. \begin{cases} \sum_{i=1}^{m} \beta_i = 1 \\ \forall i \in [\![1, m]\!] \beta_i \geqslant 0 \end{cases}$$

D'après le lemme 2 de [27], si le maximum est unique alors la dérivée de R^2 par rapport à un hyperparamètre s'exprime en fonction de β et de la dérivée de la matrice de Gram par rapport à cet hyperparamètre. Soit β^* le vecteur de multiplicateurs de Lagrange optimaux alors nous avons :

$$\frac{\partial R^2}{\partial \theta} = \sum_{i=i}^{m} \beta_i^* \frac{\partial \kappa_\lambda (x_i, x_i)}{\partial \theta} - \sum_{i,j=1}^{m} \beta_i^* \beta_j^* \frac{\partial \kappa_\lambda (x_i, x_j)}{\partial \theta}. \tag{A.1}$$

Dans le cas des noyaux RBF, il est possible d'utiliser l'approximation du rayon proposé dans [28] : $R^2 = 1 + \lambda$. Dans ce cas l'obtention de la dérivée est triviale.

A.2 Dérivée de w

La dérivée du terme de marge w s'obtient par l'intermédiaire de la dérivée de $\alpha_\mathcal{E}$. Or d'après la proposition 9, en supposant que $A_\mathcal{E}(\lambda)$ soit inversible, nous avons

$$\alpha_\mathcal{E}^a (\lambda) = A_\mathcal{E} (\lambda)^{-1} C_\mathcal{E}.$$

Et d'après la formule de dérivation de l'inverse d'une matrice, la dérivée de $\alpha_\mathcal{E}^a(\lambda)$ par rapport à θ est

$$\frac{\partial \alpha_\mathcal{E}^a (\lambda)}{\partial \theta} = -A_\mathcal{E} (\lambda)^{-1} \frac{\partial A_\mathcal{E} (\lambda)}{\partial \theta} \alpha_\mathcal{E}^a (\lambda). \tag{A.2}$$

En utilisant les notations de l'équation (3.33), $\alpha_\mathcal{E}^a (\lambda)^T A_\mathcal{E} (\lambda) \alpha_\mathcal{E}^a (\lambda)$ s'écrit :

$$\alpha_\mathcal{E}^a (\lambda)^T A_\mathcal{E} (\lambda) \alpha_\mathcal{E}^a (\lambda) = -\alpha_\mathcal{E} (\lambda)^T H_{\mathcal{E},\mathcal{E}} (\lambda) \alpha_\mathcal{E} (\lambda) + \alpha_\mathcal{E} (\lambda)^T M_I b(\lambda) + \underbrace{b(\lambda)^T C \alpha_\mathcal{E} (\lambda) + b(\lambda)^T D b(\lambda)}_{=0 \text{ d'après (3.21)}}.$$

La quantité $\alpha_{\mathcal{E}}(\lambda)^T M_I b(\lambda)$ est elle aussi égale à zéro. En effet, chaque composante de $\alpha_{\mathcal{E}}(\lambda)^T M_I$ correspond à une somme par catégorie des multiplicateurs de Lagrange. Or d'après les contraintes de la LLW-M-SVM, on sait que ces sommes sont identiques. Soit η cette somme, nous avons alors :

$$\alpha_{\mathcal{E}}(\lambda)^T M_I b(\lambda) = \eta 1_Q^T b(\lambda) = 0$$

puisque la somme des biais est égale à zéro. Il en résulte que

$$\alpha_{\mathcal{E}}^a(\lambda)^T A_{\mathcal{E}}(\lambda) \alpha_{\mathcal{E}}^a(\lambda) = -\alpha_{\mathcal{E}}(\lambda)^T H_{\mathcal{E},\mathcal{E}}(\lambda) \alpha_{\mathcal{E}}(\lambda). \tag{A.3}$$

En utilisant (A.2) et (A.3), la dérivée de w par rapport à θ exprimée en fonction de $\alpha_{\mathcal{E}}^a(\lambda)$ est :

$$\begin{aligned}
\frac{\partial w}{\partial \theta} &= -\frac{\partial \alpha_{\mathcal{E}}^a(\lambda)^T A_{\mathcal{E}}(\lambda) \alpha_{\mathcal{E}}^a(\lambda)}{\partial \theta} \\
&= -\alpha_{\mathcal{E}}^a(\lambda)^T \frac{\partial A_{\mathcal{E}}(\lambda)}{\partial \theta} \alpha_{\mathcal{E}}^a(\lambda) + 2\alpha_{\mathcal{E}}^a(\lambda)^T A_{\mathcal{E}}(\lambda) A_{\mathcal{E}}(\lambda)^{-1} \frac{\partial A_{\mathcal{E}}(\lambda)}{\partial \theta} \alpha_{\mathcal{E}}^a(\lambda) \\
&= \alpha_{\mathcal{E}}^a(\lambda)^T \frac{\partial A_{\mathcal{E}}(\lambda)}{\partial \theta} \alpha_{\mathcal{E}}^a(\lambda) \\
&= -\alpha_{\mathcal{E}}(\lambda)^T \frac{\partial H_{\mathcal{E},\mathcal{E}}(\lambda)}{\partial \theta} \alpha_{\mathcal{E}}(\lambda) = -\alpha(\lambda)^T \frac{\partial H(\lambda)}{\partial \theta} \alpha(\lambda)
\end{aligned}$$

La dernière ligne s'obtient en remarquant que $\frac{\partial A_{\mathcal{E}}(\lambda)}{\partial \theta} = \begin{pmatrix} -\frac{\partial H_{\mathcal{E},\mathcal{E}}(\lambda)}{\partial \theta} & 0_{m_{\mathcal{E}},Q} \\ 0_{Q,m_{\mathcal{E}}} & 0_{Q,Q} \end{pmatrix}$. De plus, puisque $H = K_\lambda \otimes \left(1_Q - \frac{1}{Q} 1_Q 1_Q^T\right)$, la dérivée de w par rapport à θ s'exprime en fonction de $\frac{\partial K_\lambda}{\partial \theta}$:

$$\frac{\partial w}{\partial \theta} = -\alpha_{\mathcal{E}}^T(\lambda) \left[\frac{\partial K_\lambda}{\partial \theta} \otimes \left(1_Q - \frac{1}{Q} 1_Q 1_Q^T\right)\right] \alpha_{\mathcal{E}}(\lambda). \tag{A.4}$$

A.3 Dérivée de K_λ

La dérivée de la borne Rayon-Marge dépend principalement de la dérivée de la matrice de Gram. Il est possible de calculer les dérivées pour n'importe quel noyau différentiable par rapport à ces hyperparamètres. Nous choisissons le noyau RBF sphérique habituel $\kappa(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2e^{2s}}}$. Le noyau κ_λ est donc

$$\kappa_\lambda(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2e^{2s}}} + e^{-c}\delta_{i,j}$$

et ses dérivées par rapport à c et à s sont :

$$\begin{cases} \frac{\partial \kappa_\lambda(x_i,x_j)}{\partial s} = e^{-2s}\|x_i - x_j\|^2 \kappa_\lambda(x_i, x_j) \\ \frac{\partial \kappa_\lambda(x_i,x_j)}{\partial c} = -e^{-c}\delta_{i,j} \end{cases}.$$

La combinaison de (A.1) et de (A.4) permet d'exprimer les dérivées de la borne Rayon-Marge

pour un coût particulièrement faible. Les opérations les plus coûteuses sont

- le calcul des formes quadratiques en $\alpha\,(\lambda)$
- la résolution du problème quadratique pour trouver le diamètre.

Il est possible d'aborder le premier point en parallélisant, sans aucun effort, ce calcul. Le second point peut être évité en approchant la valeur du rayon par la formule donnée dans [28]. Toutefois, en pratique, le temps nécessaire pour ces calculs est négligeable devant le temps d'apprentissage de la M-SVM^2. En effet, l'apprentissage de cette machine est relativement long. Elle comporte beaucoup de vecteurs support et l'évaluation de la fonction objectif du primal, lors de l'apprentissage, est complexe (voir [63] pour plus de détails).

A.4 Remarque sur la mise en œuvre

Pour le cas bi-classe, les simulations ont montré que la borne Rayon-Marge n'était pas convexe. En effet, la borne Rayon-Marge peut présenter des plateaux pour les très grandes ou très faibles valeurs de λ rendant l'optimisation difficile. De plus, lorsque σ est suffisamment grand, il y a possibilité d'obtenir un minimum local de la borne. Ces résultats sont très bien illustrés dans [28]. Pour la borne Rayon-Marge multi-classe, les comportements sont similaires. Toutefois les résultats obtenu en appliquant des algorithmes de descente en gradient sont très satisfaisants et c'est cette approche qui a été retenue par la communauté. Le choix de l'initialisation est donc très important pour s'assurer des résultats de la procédure d'optimisation. Par défaut, nous choisissons ceux de [71]. Afin de réaliser cette procédure, l'algorithme de Broyden-Fletcher-Goldfarb-Shanno (BFGS) a été retenu, comme dans [71], puisque nécessitant peu d'itérations. Pendant nos expériences, l'algorithme convergeait au bout d'une trentaine d'itérations ce qui est un nombre du même ordre que celui obtenu pour la borne Rayon-Marge bi-classe. Dans le cas d'un noyau gaussien elliptique, l'optimisation de la borne Rayon-Marge se fait sur le coefficient de régularisation et sur celui de dilatation du noyau elliptique. Cette méthode permet de garder les poids relatifs des descripteurs trouvés par l'alignement de noyau tout en choisissant la largeur de bande la plus adaptée au classifieur. En effet, l'alignement de noyau à tendance à fournir un noyau requérant une petite valeur de λ pour obtenir de bonnes performances, ce qui rend l'apprentissage difficile [1].

La procédure de minimisation de la borne Rayon-Marge a été implantée en utilisant la librairie de BFGS écrite par Okazaki `http://www.chokkan.org/software/liblbfgs/` et le solveur de M-SVM MSVMpack[76].

1. Les solveurs classiques convergent plus lentement avec les petites valeurs de λ

Annexe B

Estimating the Class Posterior Probabilities in Biological Sequence Segmentation

Abstract :
To tackle segmentation problems on biological sequences, we advocate the use of a hybrid architecture combining discriminant and generative models in the framework of a hierarchical approach. Multi-class support vector machines and neural networks provide a set of initial predictions. These predictions are post-processed by classifiers estimating the class posterior probabilities. The outputs of this cascade of classifiers, named MSVMpred, are then used to derive the emission probabilities of a hidden Markov model performing the final prediction. This article deals with the evaluation of MSVMpred both as a stand alone classifier, i.e., according to the recognition rate, and as part of a hybrid architecture, i.e., with respect to the quality of the probability estimates.

Introduction

We are interested in problems of bioinformatics which can be specified as follows : a biological sequence must be split into consecutive segments belonging to different categories given a priori. This class contains problems of central importance in biology such as protein secondary structure prediction, alternative splicing prediction, or the search for the genes of non-coding RNAs. To tackle it, we advocate the use of a hybrid architecture combining discriminant and generative models in the framework of a hierarchical approach. It was first outlined in [56], and was later developed by the community (see for instance [81]). Discriminant models compute estimates of the class posterior probabilities based on local information extracted from the sequence (or a multiple alignment). These estimates are then post-processed by generative models, to produce the final prediction. The class of problems of interest further suggests a specific organization of

the discriminant models : a first set of classifiers performs predictions based on the content of a window sliding on the sequence, and a second set combines and filters these predictions. The most successful instance of this "cascade" architecture, MSVMpred [64], is built as follows : the first level predictions are made by multi-class support vector machines (M-SVMs) [61] and neural networks (NNs) [88], the second level ones by "combiners" selected according to their capacity.

The present work aims at assessing the potential of MSVMpred through comparative studies. The assessment regards both the recognition rate and the quality of the probability estimates. Experiments are performed on synthetic data and in protein secondary structure prediction. The performance of reference is provided by the standard pairwise coupling procedure [67] applied to the (post-processed) outputs of bi-class support vector machines (SVMs) [32]. The results highlight the potential of MSVMpred and by way of consequence our hybrid architecture. The organization of the paper is as follows. Section B.1 is a general introduction to MSVMpred. The basic experimental protocol is detailed in Section B.2. Section B.3 provides results obtained on synthetic data. Experimental results in protein secondary structure prediction are exposed in Section B.4, and we draw conclusions in Section B.5.

B.1 MSVMpred

Let $\mathcal{X}$ be a non empty set and $Q \in \mathbb{N} \setminus [\![0, 2]\!]$. $\mathcal{X}$ and $[\![1, Q]\!]$ are respectively the description space and the set of categories of a discrimination problem characterized by a $\mathcal{X} \times [\![1, Q]\!]$-valued random pair (X, Y) whose distribution is unknown. All the knowledge regarding this distribution is provided by a set of labelled data. This set is used to select in a given class of functions a function assigning a category to the descriptions with minimal probability of error (risk). When the learning problem is formulated in that way, MSVMpred is simply defined as a two-layer cascade of classifiers with domain $\mathcal{X}$ and range the unit $(Q-1)$-simplex. M-SVMs and NNs produce initial predictions which, after a possible post-processing, are exploited by classifiers of appropriate capacity estimating the class posterior probabilities.

B.1.1 Multi-class support vector machines

M-SVMs are multi-class extensions of the (bi-class) SVM which do not rely on a decomposition method. As models of pattern recognition, they are characterized by the specification of a class of functions and a learning problem.

Définition 23 (Class of functions $\mathcal{H}_{\kappa,Q}$)**.** *Let κ be a real-valued positive type function [13] on $\mathcal{X}^2$ and let $(\mathbf{H}_\kappa, \langle \cdot, \cdot \rangle_{\mathbf{H}_\kappa})$ be the corresponding reproducing kernel Hilbert space. The class of vector-valued functions on which a Q-category M-SVM with kernel κ is based is the class*

$$\mathcal{H}_{\kappa,Q} = (\mathbf{H}_\kappa \oplus \{1\})^Q$$

where $\{1\}$ is the space of real-valued constant functions on $\mathcal{X}$.

$\mathcal{H}_{\kappa,Q}$ is a class of mulivariate affine functions on $\mathbf{H}_\kappa$. Indeed,

$$\forall h \in \mathcal{H}_{\kappa,Q},\ \forall x \in \mathcal{X},\ \ h\left(x\right) = \bar{h}\left(x\right) + b = \left(\langle \bar{h}_k, \kappa\left(x,\cdot\right)\rangle_{\mathbf{H}_\kappa} + b_k\right)_{1\leqslant k\leqslant Q},$$

where $\bar{h} = \left(\bar{h}_k\right)_{1\leqslant k\leqslant Q} \in \mathbf{H}_\kappa^Q$ and $b = \left(b_k\right)_{1\leqslant k\leqslant Q} \in \mathbb{R}^Q$.

Définition 24 (Generic model of M-SVM, Definition 4 in [61]). *Let $\mathcal{X}$ be a non empty set and $Q \in \mathbb{N}\setminus[\![0,2]\!]$. Let κ be a real-valued positive type function on $\mathcal{X}^2$ and let $\mathcal{H}_{\kappa,Q}$ be the class of functions induced by κ according to Definition 23. For $m \in \mathbb{N}^*$, let $d_m = \left(\left(x_i, y_i\right)\right)_{1\leqslant i\leqslant m} \in \left(\mathcal{X}\times[\![1,Q]\!]\right)^m$ and $\xi \in \mathbb{R}^{Qm}$ with $\left(\xi_{(i-1)Q+y_i}\right)_{1\leqslant i\leqslant m} = 0_m$. A Q-*category M-SVM *with kernel κ and training set d_m is a discriminant model trained by solving a convex quadratic programming problem of the form*

Problème 20 (Learning problem of an M-SVM, primal formulation).

$$\min_{h,\xi}\left\{\|M\xi\|_p^p + \lambda\sum_{k=1}^{Q}\left\|\bar{h}_k\right\|_{\mathbf{H}_\kappa}^2\right\}$$

$$s.t.\begin{cases}\forall i\in[\![1,m]\!],\ \forall k\in[\![1,Q]\!]\setminus\{y_i\},\ \ K_1 h_{y_i}\left(x_i\right) - h_k\left(x_i\right) \geqslant K_2 - \xi_{(i-1)Q+k}\\ \forall i\in[\![1,m]\!],\ \forall\left(k,l\right)\in\left([\![1,Q]\!]\setminus\{y_i\}\right)^2,\ \ K_3\left(\xi_{(i-1)Q+k} - \xi_{(i-1)Q+l}\right) = 0\\ \forall i\in[\![1,m]\!],\ \forall k\in[\![1,Q]\!]\setminus\{y_i\},\ \ \left(2-p\right)\xi_{(i-1)Q+k} \geqslant 0\\ \left(1-K_1\right)\sum_{k=1}^{Q} h_k = 0\end{cases}$$

where $\lambda \in \mathbb{R}_+^*$, $\left(K_1, K_3\right) \in \{0,1\}^2$, and $K_2 \in \mathbb{R}_+^*$. M is a $Qm\times Qm$ matrix of rank $\left(Q-1\right)m$ such that for all i in $[\![1,m]\!]$, its column of index $\left(i-1\right)Q + y_i$ is equal to 0_{Qm}. $p \in \{1,2\}$ and if $p = 1$, then M is a diagonal matrix.

If the problem of deriving class posterior probability estimates from the outputs of an SVM, in the bi-class case, or a set of SVMs, in the multi-class case, has been frequently addressed, this is not the case for the M-SVMs.

B.1.2 Second level discriminant models

Several options are available to perform the second level predictions. Both the polytomous logistic regression (PLR) model [68] and the linear ensemble methods (LEMs) [59] estimate the class posterior probabilities. Under mild hypotheses, this property is shared by many NNs, among which the most common one, the multi-layer perceptron (MLP) (see for instance [88]). All three models are assessed separately in our experiments. We have introduced them in order of increasing capacity [60]. Indeed, the PLR is a linear separator. An LEM combines Q-category classifiers taking their values in the unit $(Q-1)$-simplex. To combine classifiers that do not exhibit this property, such as the M-SVMs, the introduction of an intermediate step of post-processing is required, which can give birth to a nonlinear separator (in the space where the

M-SVM	M	p	K_1	K_2	K_3
WW-M-SVM	$I_{Qm}(d_m)$	1	1	1	0
CS-M-SVM	$\frac{1}{Q-1} I_{Qm}(d_m)$	1	1	1	1
LLW-M-SVM	$I_{Qm}(d_m)$	1	0	$\frac{1}{Q-1}$	0
M-SVM^2	$M^{(2)}$	2	0	$\frac{1}{Q-1}$	0

TABLE B.1 – Specifications of the four M-SVMs used as base classifiers in MSVMpred

outputs of the base classifiers live). At last, an MLP using a softmax activation function for the output units and the cross-entropy (CE) loss (a sufficient condition for its outputs to be class posterior probability estimates) is an extension of the PLR obtained by adding a hidden layer. The boundaries it computes are nonlinear in its input space. The availability of classifiers of different capacities for the second level of the cascade is an important feature of MSVMpred. It makes it possible to cope with one of the main limiting factors to the performance of modular architectures : overfitting.

B.2 Experimental protocol

When computing polytomies with SVMs, the standard way to derive class posterior probability estimates consists in combining the one-against-one decomposition scheme with pairwise coupling [67]. For each $x \in \mathcal{X}$, the $\binom{Q}{2}$ outputs of the SVMs are post-processed by a parameterized sigmoid [85] to provide estimates of the probabilities $\mathbb{P}(Y = k \mid Y \in \{k, l\}, X = x)$. These estimates are then used to derive estimates of the probabilities $\mathbb{P}(Y = k \mid X = x)$ by means of a maximum likelihood procedure.

The implementation of MSVMpred involves the four main models of M-SVMs as base classifiers, i.e., the models of Weston and Watkins (WW), Crammer and Singer (CS), Lee and co-authors (LLW), and the M-SVM^2 (see [61] for a survey). Let $I_{Qm}(d_m)$ and $M^{(2)}$ designate two instances of M whose general term $m_{(i-1)Q+k,(j-1)Q+l}$ is respectively $\delta_{i,j}\delta_{k,l}(1 - \delta_{y_i,k})$ and $(1 - \delta_{y_i,k})(1 - \delta_{y_j,l})\left(\delta_{k,l} + \frac{\sqrt{Q}-1}{Q-1}\right)\delta_{i,j}$, where δ is the Kronecker symbol. The formulations of the aforementioned M-SVMs as instances of the generic model correspond to the values of the hyperparameters reported in Table B.1.

Unless otherwise specified, the SVMs and M-SVMs implemented use a spherical Gaussian kernel. To post-process the outputs of the M-SVMs prior to presenting them in input of an LEM, we resorted to the PLR. Besides the pairwise coupling, the set of models providing the performance of reference is made up of the PLR, the MLP, and an M-SVM (precisely the CS-M-SVM) post-processed by the PLR. The MLP, as a universal approximator, has the potential to approximate the probabilities, and thus the Bayes classifier, arbitrarily well. Its performance is limited by the efficiency of the back-propagation algorithm. For a classifier g from $\mathcal{X}$ into the unit $(Q-1)$-simplex, the empirical CE measured on $((x_i, y_i))_{1 \leqslant i \leqslant m}$ is $-\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{Q}\mathbb{P}(Y = k \mid X = x_i)\ln(g_k(x_i))$ when the probabilities are

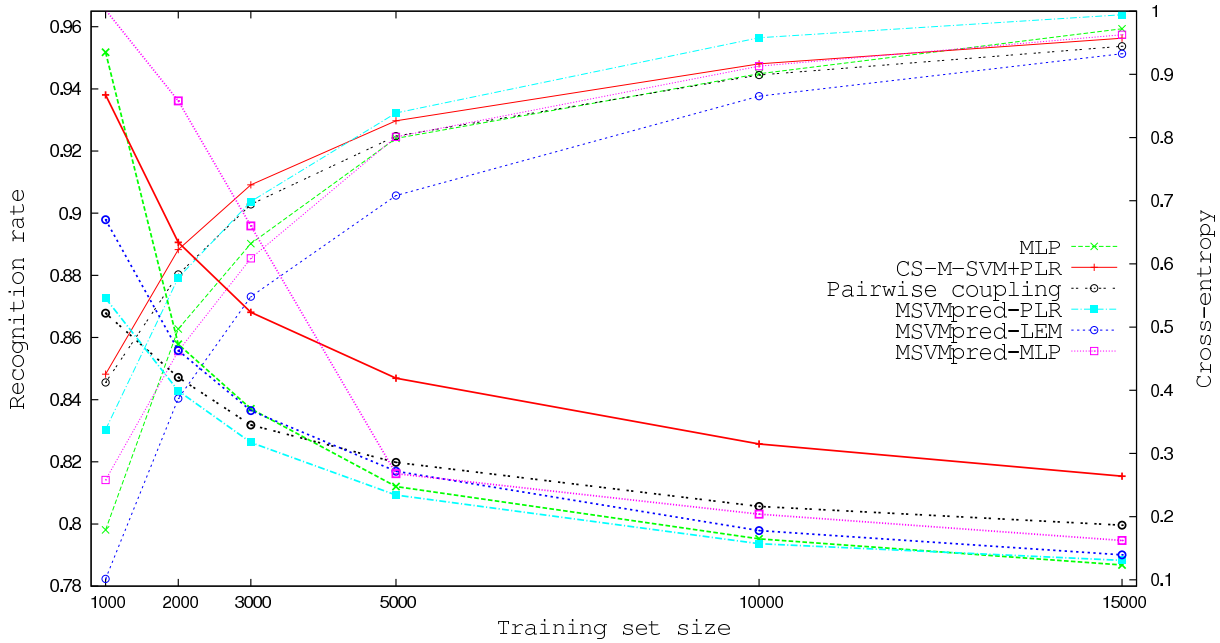


FIGURE B.1 – Prediction accuracy as a function of the training set size

known, $-\frac{1}{m}\sum_{i=1}^{m}\ln\left(g_{y_i}\left(x_i\right)\right)$ otherwise. This quantity is used to assess the accuracy of the probability estimates. The significance of the differences in recognition rate is measured by means of the two sample proportion test.

B.3 Experiments on synthetic data

We used an instance of the 10-category problem studied in [48]. In this problem, the categories are equiprobable and their probability density distributions are 20-dimensional Gaussians. The means of the Gaussians are randomly generated from a uniform distribution on $[0,1]^{20}$. The covariance matrices are also random. For each of them, the eigenvectors are generated from a uniform distribution on the unit sphere of $\mathbb{R}^{20}$ subject to orthogonality constraints. The square-roots of the eigenvalues are drawn from a uniform distribution on $[0.01, 1.01]$. Our Monte Carlo estimates of the risk and entropy of the Bayes classifier : $\mathbb{E}_{(X,Y)}\left[-\ln\left(\mathbb{P}\left(Y \mid X\right)\right)\right] = \mathbb{E}_X\left[-\sum_{k=1}^{Q}\mathbb{P}\left(Y=k \mid X\right)\ln\left(\mathbb{P}\left(Y=k \mid X\right)\right)\right]$ are respectively 0.50% and 0.0142. For these experiments, the NN used as base classifier, in parallel with the M-SVMs, is an MLP. To train MSVMpred, the training set is split into two subsets. Three quarters of the data are used for the base classifiers and their possible post-processing, the rest for the combiner. This decomposition is also the one used for the pairwise coupling. On the contrary, in the case when the CS-M-SVM alone is post-processed by the PLR, since no combiner is to be trained, the M-SVM and its post-processing are respectively trained on the first and the second subset. The test set is made up of 60000 examples. Figure B.1 displays the prediction accuracy as a function of the training set size m, for all the classifiers but the PLR (alone), discarded for lack of performance. For m

large enough (over 5000), the three variants of MSVMpred are uniformly superior to the pairwise coupling in terms of CE. The most efficient variant is the one of lowest capacity (PLR). For $m \geqslant 10000$, its recognition rate is superior to that of the second best classifier with confidence exceeding 0.95. MSVMpred seems capable of making the best of the complementarity of the base classifiers, since this gain is not obtained at the expense of the quality of the probability estimates. Indeed, the CE of the best variant is inferior to that of the MLP (alone), until both values become indistinguishable (for $m \geqslant 15000$).

B.4 Protein secondary structure prediction

Predicting the secondary structure of a protein is a three-category discrimination task consisting in assigning a conformational state α-helix, β-strand or aperiodic (coil), to each residue (amino acid) of its sequence. In that context, the two levels of classification performed in cascade correspond respectively to a sequence to structure prediction and a structure to structure prediction. State-of-the-art prediction methods exhibit a recognition rate around 80%, whereas the best cascades published so far remain slightly below 78%.

B.4.1 Sequence-to-structure classification

For this classification, deriving the predictors from a multiple alignment rather than from the sole sequence of interest makes it possible to incorporate some evolutionary information. The descriptions of the residues of a protein sequence are thus derived from the corresponding position-specific scoring matrix (PSSM) produced by PSI-BLAST [5]. To generate the PSSMs, the version 2.2.25 of the BLAST package is used. Choosing BLAST in place of the more recent BLAST+ offers the facility to extract more precise PSSMs. Three iterations are performed against the nr database with an E-value inclusion threshold of 0.005. The nr database is filtered by pfilt [70] to remove low complexity regions, transmembrane spans and coiled coil regions. Since a sliding window is used, the description of a residue is thus obtained by appending consecutive rows of the PSSM associated with the sequence to which it belongs. To dedicate the M-SVMs to the task of interest, we chose an elliptic Gaussian kernel weighting differently the positions of this window. To set the values of the weights, we applied a straightforward multi-class extension of the kernel target alignment. We had to center the data in the feature space according to the formula given in [29], and to penalize the corresponding objective function with the ℓ_2 norm of the weighting vector. This kernel was also used by the SVMs involved in the pairwise coupling. To exploit the sequential nature of the data, the NN we used as additional base classifier was a recurrent one : the bidirectional recurrent neural network (BRNN) [10].

B.4.2 Structure-to-structure classification

As usual when applying structure-to-structure classification, the predictors are provided by the content of a window sliding on the outputs of the sequence-to-structure classifiers. For the

	PLR	MLP	BRNN	CS-M-SVM + PLR	Pairwise coupling	MSVMpred PLR	MSVMpred LEM	MSVMpred MLP	MSVMpred BRNN
Q_3 (%)	72.5	76.1	77.0	77.0	76.6	78.2	77.9	78.1	78.1
CE	0.674	0.585	0.568	0.578	0.581	0.542	0.551	0.537	0.548
C_α	0.62	0.71	0.72	0.72	0.71	0.74	0.74	0.74	0.74
C_β	0.55	0.61	0.62	0.62	0.61	0.64	0.64	0.64	0.64
C_{coil}	0.54	0.57	0.58	0.59	0.58	0.60	0.60	0.60	0.60
Sov'99 (%)	67.7	68.7	71.3	71.2	71.4	74.6	74.3	73.8	73.7

TABLE B.2 – Performance of MSVMpred in protein secondary structure prediction

PLR and the MLP specifically, they are weighted according to their position in their window. The weighting is derived in the same way as the one of the elliptic Gaussian kernel.

B.4.3 Experimental results

The data set is CB513, a standard benchmark made up of 513 sequences for a total of 84119 residues. The secondary structure assignment was performed by the DSSP program, with the reduction from 8 to 3 conformational states following the CASP method. The first and second sliding window, centered on the residue of interest, have a size of 13 and 15 respectively. A seven-fold cross-validation procedure was implemented. At each step, two thirds of the training set were used to train the sequence-to-structure classifiers and their possible post-processing, and one third to train the combiner. This decomposition was also used for the pairwise coupling. Since a secondary structure prediction method must fulfill specific requirements in order to be useful for the biologist, two performance measures were added to the recognition rate (Q_3) and the CE : the Pearson correlation coefficients $C_{\alpha/\beta/\text{coil}}$ and the segment overlap measure (Sov'99). Results are gathered in Table B.2.

The Q_3 of each variant of MSVMpred is statistically superior to that of the pairwise coupling and all the single classifiers with confidence exceeding 0.95. The value of the CE confirms this superiority. In comparison with the results obtained on the synthetic data, a nice feature is that all variants of MSVMpred perform equally well. Model selection is not an issue here.

B.5 Conclusions and ongoing research

This article has illustrated the potential of MSVMpred as discriminant model performing the low-level processing of the data in our hybrid architecture dedicated to biological sequence segmentation. In protein secondary structure prediction, its performance is slightly superior to that of the best cascades published so far, and significantly superior to that of pairwise coupling, or single classifiers. An appropriate choice of the combiner, governed by the concern of capacity control, could optimize jointly the recognition rate and the quality of the class posterior probability estimates. We are currently working on increasing the scope of MSVMpred in biological

sequence segmentation without loss of control on its generalization performance.

The authors would like to thank C. Magnan for providing them with the 1D-BRNN package.

Notations

Cadre théorique

Q ..... Nombre de catégories

$\mathcal{X}$ ..... Espace de description des données

$\mathcal{Y}$ ... Ensemble de catégorie. $\{-1, 1\}$ dans le cas bi-classe et $[\![1, Q]\!]$ dans multi-classe à Q catégories.

P ..... Mesure de probabilité sur l'espace produit $\mathcal{X} \times \mathcal{Y}$

$P_{\mathcal{X}}$ ..... Distribution marginale de P sur $\mathcal{X}$

(X, Y) ..... Couple aléatoire distribué suivant P

$\mathcal{G}$ ..... Classe de fonction de $\mathcal{X}$ dans $\mathbb{R}$ dans le cas bi-classe) et dans $\mathbb{R}^Q$ pour le cas multi-classe

f ..... Fonction de décision allant de $\mathcal{X}$ dans $\mathcal{Y}$ et associée à une fonction de $\mathcal{G}$

D_m ..... m-échantillon constitué de copies de (X, Y) indépendantes

m ..... Taille du m-échantillon

$\{(x_i, y_i) \ : 1 \leqslant i \leqslant m\}$ ..... Réalisation de D_m

i,j ..... Indices d'exemples

k ..... Indice de catégorie

$\kappa(\cdot, \cdot)$ ..... Noyau symétrique défini positif

$(\mathbf{H}_\kappa, \langle \cdot, \cdot \rangle_{\mathbf{H}_\kappa})$ ..... RKHS associé au noyau κ

$\bar{\mathcal{H}}$ ..... Espace vectoriel correspondant à $(\mathbf{H}_\kappa, \langle \cdot, \cdot \rangle_{\mathbf{H}_\kappa})$ dans le cas bi-classe et à $(\mathbf{H}_\kappa, \langle \cdot, \cdot \rangle_{\mathbf{H}_\kappa})^Q$ dans le cas multi-classe

$\mathcal{H}$ ..... Espace affine constitués de $\bar{\mathcal{H}}$ en somme direct avec les fonctions constantes

h ..... Fonction de $\mathcal{H}$

$\bar{h}$ ..... Fonction de $\bar{\mathcal{H}}$

Élément d'une SVM

λ ..... Coefficient de régularisation

C ..... Coefficient de marge douce ($C = \frac{1}{\lambda}$)

ξ ..... Vecteur de variable d'écart

$(\cdot)_+$ ..... Fonction troncature : $(\cdot)_+ = max(0, \cdot)$

β .. Vecteur des variables duales associées au contrainte de bon classement du problème d'apprentissage exprimé avec λ

α .. Vecteur des variables duales associées au contrainte de bon classement du problème d'apprentissage exprimé avec C ($\alpha = \frac{\beta}{\lambda}$)

γ ..... Marge de la SVM

$m_+(m_-)$ ..... Nombre d'exemples de la catégorie positive (respectivement négative)

Chemin de régularisation

$\mathcal{E}(\lambda)$ Ensemble des indices des variables duales susceptibles de changer de valeur à un λ donné.

$\beta_{\mathcal{E}}(\lambda), \alpha_{\mathcal{E}}(\lambda)$ .. Vecteur des multiplicateurs de Lagrange (α ou β) associés aux exemples d'indice dans E

$A_{\mathcal{E}}$ ..... Matrice du système de Kuhn-Tucker (Notée $A_{\mathcal{E}}(\lambda)$ lorsqu'elle dépend de λ)

ν, ρ ..... Variables intermédiaires pour l'obtention des multiplicateurs de Lagrange

R_r ..... Matrice de rang r servant à approcher $H(\epsilon)$

r ..... Rang de l'approximation $H_r(\epsilon)$

Sélection de modèle

$\tilde{Err}^{(cv,1)}$ ..... Erreur de validation croisée leave-one-out

S_p Span du point p : distance, dans l'espace de représentation, de l'exemple p à l'espace engendré par les vecteurs support (privés de p)

$S_{p,k}$ ..... Span de p relativement à la catégorie k

$M_{sv}^{(cv,1)}$ ..... Majoration de l'erreur d'approximation de l'erreur de validation croisée leave-one-out

Alignement de noyau

$A(\cdot,\cdot)$ ..... Alignement de noyau/cible

$A_m(\cdot,\cdot)$ ..... Alignement empirique de noyau/cible

κ_c ..... noyau centré associé à κ

$\rho(\kappa,\kappa')$ ..... Alignement de noyaux centrés

ν ..... Coefficient de régularisation de l'alignement régularisé de noyaux centrés

Matrices

1_n ..... Vecteur de $\mathbb{R}^n$ dont chaque composante vaut 1

$V_{\mathcal{J}}$ ..... Vecteur défini par $(V_i)_{i\in\mathcal{J}}$

$M_{\mathcal{I},\mathcal{J}}$ Matrice définie par $(M_{i,j})_{i\in\in\mathcal{I},j\in\in\mathcal{J}}$. Si un des 2 indices est remplacé par un point ($\cdot$), ceci correspond à toute la ligne/colonne en question.

$\odot$ ..... Produit de Hadamart

$\otimes$ ..... Produit de Kronecker

Bibliographie

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Proceedings of the 2nd International Symposium on Information Theory*, pages 267–281, 1973.

[2] E.L. Allgower and K. Georg. Continuation and path following. *Acta Numerica*, 2 :1–64, 1993.

[3] E.L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[4] L.B. Almeida, T. Langlois, J. Amaral, and A. Plakhov. On-line learning in neural networks. In David Saad, editor, *Parameter adaptation in stochastic optimization*, pages 111–134. Cambridge University Press, New York, NY, 1998.

[5] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST : a new generation of protein database search programs. *Nucleic Acids Research*, 25(17) :3389–3402, 1997.

[6] D. Anguita, A. Boni, and S. Ridella. Evaluating the generalization ability of support vector machines through the bootstrap. *Neural Processing Letters*, 11(1) :51–58, February 2000.

[7] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4 :40–79, 2010.

[8] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3) :337–404, 1950.

[9] G. Bakiri and T.G. Dietterich. Achieving high-accuracy text-to-speech with machine learning. *Data mining in speech synthesis*, 10, 1999.

[10] P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11) :937–946, 1999.

[11] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on*, 5(4) :537 –550, jul 1994.

[12] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5 :1089–1105, 2004.

[13] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, Boston, 2004.

[14] G. Blanchard, P. Massart, R. Vert, and L. Zwald. Kernel projection machine : a new tool for pattern recognition. In *NIPS 17*, pages 1649–1656, 2005.

[15] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *COLT'92*, pages 144–152, 1992.

[16] R.A. Bradley and M.E. Terry. Rank analysis of incomplete block designs. i. the method of paired comparisons. *Biometrika*, 39(3/4) :324–345, 1952.

[17] G. Brown, A. Pocock, M.J. Zhao, and M. Luján. Conditional likelihood maximisation : A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13 :27–66, 2012.

[18] P. Burman. A comparative study of ordinary cross-validation, ν-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3) :503–514, 1989.

[19] O. Carriere, J.P. Hermand, and Y. Stéphan. Acoustic monitoring of the ushant front : a feasibility study. In *OCEANS 2009-EUROPE*, pages 1–5. IEEE, 2009.

[20] G.C. Cawley and N.L.C Talbot. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Network*, 17(10) :1467–1475, 2004.

[21] G.C. Cawley and N.L.C. Talbot. Preventing over-fitting during model selection using bayesian regularisation. *Journal of Machine Learning Research*, 8 :841–861, 2007.

[22] G.C. Cawley and N.L.C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11 :2079–2107, 2010.

[23] C. Chailloux, B. Kinda, C. Gervaise, J. Bonnel, Y. Stéphan, J. Mars, and JP Hermand. Modelling of ambient noise created by a shipping lane to prepare passive inversion, application to ushant case. In *Underwater Acoustics Measurements*, 2011.

[24] M.W. Chang and C.J. Lin. Leave-one-out bounds for support vector regression model selection. *Neural Computation*, 17(5) :1188–1222, 2005.

[25] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin. Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research*, 11 :1471–1490, 2010.

[26] O. Chapelle. Training a support vector machine in the primal. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large-Scale Kernel Machines*, chapter 2, pages 29–50. The MIT Press, Cambridge, MA, 2007.

[27] O. Chapelle, V.N. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1) :131–159, 2002.

[28] K.-M. Chung, W.-C. Kao, C.-L. Sun, L.-L. Wang, and C.-J. Lin. Radius margin bounds for support vector machines with the RBF kernel. *Neural Computation*, 15(11) :2643–2681, 2003.

[29] C. Cortes, M. Mohri, and A. Rostamizadeh. Two-stage learning kernel algorithms. In *ICML*, pages 239–246, 2010.

[30] C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13 :795–828, 2012.

[31] C. Cortes, M. Mohri, and A. Talwalkar. On the impact of kernel approximation on learning accuracy. In *AISTATS 2010*, pages 113–120, 2010.

[32] C. Cortes and V.N. Vapnik. Support-vector networks. *Machine Learning*, 20(3) :273–297, 1995.

[33] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2 :265–292, 2001.

[34] N .Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel target alignment. In Dawn Holmes E. and Lakhmi Jain C., editors, *Innovations in Machine Learning*, volume 194 of *Studies in Fuzziness and Soft Computing*, pages 205–256. Springer Berlin Heidelberg, 2006.

[35] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems 14*, pages 367–373. MIT Press, 2002.

[36] D.L. David. Feature selection and feature extraction for text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 212–217. Morgan Kaufmann, 1992.

[37] T. Dietterich. Machine learning for sequential data : A review. *Structural, syntactic, and statistical pattern recognition*, pages 227–246, 2002.

[38] T.G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7) :1895–1923, 1998.

[39] K. Duan, S.S. Keerthi, and A.N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51 :41–59, 2003.

[40] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley Interscience, New York, second edition, 2001.

[41] B. Efron and R. Tibshirani. Improvements on cross-validation : The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438) :548–560, 1997.

[42] Bradley Efron. Estimating the error rate of a prediction rule : Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382) :pp. 316–331, 1983.

[43] D. Efron and G. Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1) :pp. 36–48, 1983.

[44] S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2 :243–264, 2001.

[45] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, Chichester, second edition, 1987.

[46] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5 :1531–1555, December 2004.

[47] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[48] J. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.

[49] K. Fukunaga. *Introduction to Statistical Pattern Recognition. Second Edition.* Academic Press, New York, 1990.

[50] O. Gascuel and J.L. Golmard. A simple method for predicting the secondary structure of globular proteins : implications and accuracy. *Computer applications in the biosciences : CABIOS*, 4(3) :357–365, 1988.

[51] M.G. Genton. Classes of kernels for machine learning : A statistics perspective. *Journal of Machine Learning Research*, 2 :299–312, 2001.

[52] J. Giesen, M. Jaggi, and S. Laue. Approximating parameterized convex optimization problems. *ACM Transactions on Algorithms*, 9(1) :10 :1–10 :17, 2012.

[53] M. Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 14(13) :669–688, 2002.

[54] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[55] Y. Guermeur. *Combinaison de classifieurs statistiques, application à la prédiction de la structure secondaire des protéines.* Thèse, Université Paris 6, 1997.

[56] Y. Guermeur. Combining discriminant models with new multi-class SVMs. *Pattern Analysis and Applications*, 5(2) :168–179, 2002.

[57] Y. Guermeur. *SVM multiclasses, théorie et applications.* Habilitation à diriger des recherches, Université Nancy 1, 2007.

[58] Y. Guermeur. VC theory of large margin multi-category classifiers. *Journal of Machine Learning Research*, 8 :2551–2594, 2007.

[59] Y. Guermeur. Ensemble methods of appropriate capacity for multi-class support vector machines. *SMTDA'10*, pages 311–318, 2010.

[60] Y. Guermeur. Sample complexity of classifiers taking values in $\mathbb{R}^Q$, application to multi-class SVMs. *Communications in Statistics - Theory and Methods*, 39(3) :543–557, 2010.

[61] Y. Guermeur. A generic model of multi-class support vector machine. *International Journal of Intelligent Information and Database Systems (IJIIDS)*, 6(6) :555–577, 2012.

[62] Y. Guermeur, A. Lifchitz, and R. Vert. A kernel for protein secondary structure prediction. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*, chapter 9, pages 193–206. The MIT Press, Cambridge, MA, 2004.

[63] Y. Guermeur and E. Monfrini. A quadratic loss multi-class SVM for which a radius-margin bound applies. *Informatica*, 22(1) :73–96, 2011.

[64] Y. Guermeur and F. Thomarat. Estimating the class posterior probabilities in protein secondary structure prediction. In *PRIB'11*, pages 261–271, 2011.

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

[66] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5 :1391–1415, 2004.

[67] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(2) :451–471, 1998.

[68] D.W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. Wiley, 1998.

[69] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2) :195–202, 1999.

[70] D.T. Jones and M.B. Swindells. Getting the most from psi–blast. *Trends in biochemical sciences*, 27(3) :161–164, 2002.

[71] S.S. Keerthi. Efficient tuning of svm hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, 13(5) :1225–1229, September 2002.

[72] S.S. Keerthi, O. Chapelle, and D. DeCoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 7 :1493–1515, 2006.

[73] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited : A stepwise procedure for building and training a neural network. In F. Fogelman-Soulié and J. Hérault, editors, *Neurocomputing : Algorithms, Architectures and Applications*, volume F68 of *NATO ASI Series*, pages 41–50. Springer-Verlag, 1990.

[74] C.H. Lampert. Maximum margin multi-label structured prediction. In *NIPS*, pages 289–297, 2011.

[75] F. Lauer and Y. Guermeur. MSVMpack : a multi-class support vector machine package. *Journal of Machine Learning Research*, 2011.

[76] F. Lauer and Y. Guermeur. MSVMpack : a multi-class support vector machine package. *Journal of Machine Learning Research*, 12 :2269–2272, 2011. `http://www.loria.fr/~lauer/MSVMpack`.

[77] Y. Lee and Z. Cui. Characterizing the solution path of multicategory support vector machines. *Statistica Sinica*, 16(2) :391–409, 2006.

[78] Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines : Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465) :67–81, 2004.

[79] Z. Li, S. Tang, and S. Yan. Multi-class SVM classifier based on pairwise coupling. In *SVM'02*, pages 321–333, 2002.

[80] H.T. Lin, C.J. Lin, and R.C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3) :267–276, 2007.

[81] K. Lin, V.A. Simossis, W.R. Taylor, and J. Heringa. A simple and fast secondary structure prediction method using hidden neural networks. *Bioinformatics*, 21(2) :152–159, 2005.

[82] M. Meila. Data centering in feature space. In *Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[83] M. Moreira and E. Mayoraz. Improved pairwise coupling classification with correcting classifiers. In *ECML'98*, pages 160–171, 1998.

[84] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information : criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8) :1226–1238, 2005.

[85] J.C. Platt. Probabilities for SV Machines. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, chapter 5, pages 61–73. The MIT Press, Cambridge, MA, 2000.

[86] J.B. Pothin and C. Richard. Optimizing kernel alignment by data translation in feature space. In *ICASSP*, pages 3345–3348. IEEE, 2008.

[87] Gunnar Rätsch, Takashi Onoda, and K-R Müller. Soft margins for adaboost. *Machine learning*, 42(3) :287–320, 2001.

[88] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation*, 3(4) :461–483, 1991.

[89] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5 :101–141, 2004.

[90] S. Rosset. Following curved regularized optimization solution paths. In *NIPS 17*, pages 1153–1160, 2005.

[91] G. Saporta. *Probabilités Analyse des Données et Statistique*. Technip, France edition, 1990.

[92] B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *KDD'95*, pages 252–257, 1995.

[93] B. Scholkopf, A. Smola, and K.R. Müller. Kernel principal component analysis. In *Advances In Kernel Methods - Support Vector Learning*, pages 327–352. MIT Press, 1999.

[94] B. Schölkopf and A.J. Smola. *Learning with Kernels - Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Cambridge, MA, 2002.

[95] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2) :461–464, 1978.

[96] J. Shawe-Taylor, C.K.I. Williams, N. Cristianini, and J. Kandola. On the eigenspectrum of the Gram matrix and the generalization error of kernel-PCA. *IEEE Transactions on Information Theory*, 51(7) :2510–2522, 2005.

[97] M. Stone. Asymptotics for and against cross-validation. *Biometrika*, 64(1) :29–35, 1977.

[98] J.A.K. Suykens, L. Lukas, P. Van Dooren, B. De Moor, and J. Vandewalle. Least squares support vector machine classifiers : a large scale algorithm. *Proceeding of the European Conference on Circuit Theory and Design*, pages 839–842, 1999.

[99] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3) :293–300, 1999.

[100] J.A.K. Suykens and J. Vandewalle. Multiclass least squares support vector machines. In *IJCNN'99*, pages 900–903, 1999.

[101] S. Takerkart and L. Ralaivola. MKPM : A multiclass extension to the kernel projection machine. In *CVPR*, pages 2785–2791, 2011.

[102] A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston & Sons, 1977.

[103] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[104] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, 1998.

[105] V.N. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12(9) :2013–2036, 2000.

[106] G. Wahba. Multivariate function and operator estimation, based on smoothing splines and reproducing kernels. In *Santa Fe institute studies in the sciences of complexity*, volume 12, pages 95–95. Addison-Wesley, 1992.

[107] C. Walder and O. Chapelle. Learning with transformation invariant kernels. *Advances in Neural Information Processing Systems*, 20 :1561–1568, 2007.

[108] G. Wang, D.-Y. Yeung, and F.H. Lochovsky. A kernel path algorithm for support vector machines. In *Proceedings of the 24th international conference on Machine learning*, volume 227, pages 951–958, 2007.

[109] L. Wang, P. Xue, and K.L. Chan. Generalized radius-margin bounds for model selection in multi-class SVMs. Technical report, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639798, 2005.

[110] J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Royal Holloway, University of London, Department of Computer Science, 1998.

[111] H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *The Journal of Machine Learning Research*, 10 :1485–1510, 2009.

[112] K. Zhang and J.T. Kwok. Density-weighted Nyström method for computing large kernel eigensystems. *Neural Computation*, 21(1) :121–146, 2009.

[113] K. Zhang, I.W. Tsang, and J.T. Kwok. Improved Nyström low-rank approximation and error analysis. In *ICML'08*, pages 1232–1239, 2008.

[114] H. Zhu, C.K.I. Williams, R.J. Rohwer, and M. Morciniec. Gaussian regression and optimal finite dimensional linear models. In C. M. Bishop, editor, *Neural Networks and Machine Learning*. Springer-Verlag, Berlin, 1998.

Résumé

La sélection de modèle est un thème majeur de l'apprentissage statistique. Dans ce manuscrit, nous introduisons des méthodes de sélection de modèle dédiées à des SVM bi-classes et multi-classes. Ces machines ont pour point commun d'être à coût quadratique, c'est-à-dire que le terme empirique de la fonction objectif de leur problème d'apprentissage est une forme quadratique. Pour les SVM, la sélection de modèle consiste à déterminer la valeur optimale du coefficient de régularisation et à choisir un noyau approprié (ou les valeurs de ses paramètres).

Les méthodes que nous proposons combinent des techniques de parcours du chemin de régularisation avec de nouveaux critères de sélection. La thèse s'articule autour de trois contributions principales. La première est une méthode de sélection de modèle par parcours du chemin de régularisation dédiée à la ℓ_2-SVM. Nous introduisons à cette occasion de nouvelles approximations de l'erreur en généralisation. Notre deuxième contribution principale est une extension de la première au cas multi-classe, plus précisément à la M-SVM^2. Cette étude nous a conduits à introduire une nouvelle M-SVM, la M-SVM des moindres carrés. Nous présentons également de nouveaux critères de sélection de modèle pour la M-SVM de Lee, Lin et Wahba à marge dure (et donc la M-SVM^2) : un majorant de l'erreur de validation croisée leave-one-out et des approximations de cette erreur. La troisième contribution principale porte sur l'optimisation des valeurs des paramètres du noyau. Notre méthode se fonde sur le principe de maximisation de l'alignement noyau/cible, dans sa version centrée. Elle l'étend à travers l'introduction d'un terme de régularisation.

Les évaluations expérimentales de l'ensemble des méthodes développées s'appuient sur des benchmarks fréquemment utilisés dans la littérature, des jeux de données jouet et des jeux de données associés à des problèmes du monde réel.

Mots-clés: Apprentissage, Discrimination, Machine à vecteurs support (SVM), Machine à vecteurs support multi-classe (M-SVM), Sélection de modèle, Chemin de régularisation

Abstract

Model selection is of major interest in statistical learning. In this document, we introduce model selection methods for bi-class and multi-class support vector machines. We focus on quadratic loss machines, i.e., machines for which the empirical term of the objective function of the learning problem is a quadratic form. For SVMs, model selection consists in finding the optimal value of the regularization coefficient and choosing an appropriate kernel (or the values of its parameters).

The proposed methods use path-following techniques in combination with new model selection criteria. This document is structured around three main contributions. The first one is a method performing model selection through the use of the regularization path for the ℓ_2-SVM. In this framework, we introduce new approximations of the generalization error. The second main contribution is the extension of the first one to the multi-category setting, more precisely the M-SVM^2. This study led us to derive a new M-SVM, the least squares M-SVM. Additionally, we present new model selection criteria for the M-SVM introduced by Lee, Lin and Wahba (and thus the M-SVM^2). The third main contribution deals with the optimization of the values of the kernel parameters. Our method makes use of the principle of kernel-target alignment with centered kernels. It extends it through the introduction of a regularization term.

Experimental validation of these methods was performed on classical benchmark data, toy data and real-world data.

Keywords: Machine learning, Classification, Support Vector Machine (SVM), Multi-category Support Vector Machine (M-SVM), Model selection, Regularization path