Support de cours

Machines à Vecteurs Supports

Didacticiel

Hervé Frezza-Buet, Supélec

23 octobre 2013

Herve.Frezza-Buet@supelec.fr

Table des matières

Chapitre 1

Introduction

1.1 Objectif

L'objectif de ce document est de fournir une introduction pratique aux machines à vecteurs supports [1]. Le lecteur qui voudrait plus de développements mathématiques et pratiques pourra se reporter aux textes cités en bibliographie, ce document n'étant qu'une entrée en matière, qui pose les bases du problème. Néanmoins, on trouvera ici de quoi comprendre « intuitivement » les tenants et les aboutissants des SVM, avec une vision ingénieur permettant une mise en pratique rapide et raisonnée de ces techniques.

Les SVM impliquent plusieurs notions mathématiques, dont la théorie de la généralisation, peu abordée ici, la théorie de l'optimisation, et les méthodes d'apprentissage basées sur des fonctions noyau. Nous ne donnerons de ces théories que les éléments nécessaires à comprendre ce que sont les SVM, sans les développer plus avant.

1.2 Au fait, qu'est-ce qu'une SVM ?

Une SVM est un algorithme d'apprentissage, permettant d'apprendre un séparateur. Ceci ramène le problème à savoir ce qu'est un séparateur... Donnons nous un ensemble fini de vecteurs de $\mathbb{R}^n$, séparés en deux groupes, ou dit autrement en deux classes. L'appartenance à un groupe ou un autre est défini par une étiquette, associée à chacun des vecteurs, sur laquelle est inscrite « groupe 1 » ou « groupe 2 ». Trouver un séparateur revient à construire une fonction, qui prend un vecteur de notre ensemble, et peut dire de quel groupe il est. Les SVM sont une solution à ce problème, comme le serait un simple apprentissage par cœur des classes associées aux vecteurs de notre ensemble. Mais avec les SVM, on attend de bonnes propriétés de généralisation, à savoir que si un nouveau vecteur de présente, qui n'était pas dans l'ensemble, la SVM saura dire à quel groupe il est vraisemblable qu'il appartient, au regard des attributions de classes des vecteurs présents au départ.

Nous étendrons le cas du séparateur au cas où la SVM effectue un régression. Dans ce cas, une valeur numérique constitue l'étiquette des vecteurs de l'ensemble étudié, ce qui est un autre problème puisqu'il ne s'agit plus de savoir à quel groupe appartient un vecteur, groupe 1 ou groupe 2, mais de savoir « combien vaut » un vecteur.

1. Support Vector Machines, soit SVM en anglais.

1.3 Comment ça marche ?

L'idée est de poser, à partir des vecteurs dont on connaît les classes, un problème d'optimisation, du style « optimiser telle grandeur en s'assurant que ... ». Il y a deux difficultés. La première, c'est poser le bon problème d'optimisation. Cette notion de « bon » problème est celle qui fait référence aux théories mathématiques de la généralisation, et fait des SVM un outils d'un abord parfois difficile. La deuxième difficulté est de résoudre ce problème d'optimisation une fois posé, et là on tombe plus dans des subtilités informatiques, dont nous retiendrons l'algorithme SMO.

Chapitre 2

Séparateur linéaire

Nous allons d'abord traiter le cas d'un séparateur simple, quoi que pas si simple que ça finalement, le séparateur linéaire. En fait, c'est le cœur des SVM, même si les SVM fournissent des séparateurs bien plus puissants que celui que nous allons étudier.

2.1 Données du problème et notations

2.1.1 Les exemples

Nous avons dit en introduction que l'on partait d'un ensemble fini de vecteurs étiquetés. Nous noterons x un réel, alors que $\vec{x}$ désigera un vecteur de $\mathbb{R}^n$. Dans notre cas, nous dirons que l'ensemble des vecteurs étiquetés que nous nous donnons est l'ensemble des *exemples* noté S, qui contient p éléments.

$$S = \{(\vec{x}_l, y_l)\}_{1 \leq l \leq p} \text{ avec } \forall l,\ y_l \in \{-1, 1\}$$

L'appartenance d'un vecteur à une classe où à l'autre est matérialisée ici par la valeur -1 ou 1 de l'étiquette y, ce qui nous arrangera pour les calculs.

2.1.2 Le séparateur linéaire

Nous noterons le produit scalaire de deux vecteurs $\langle \vec{x}.\vec{y} \rangle$. Cette notation posée, nous pouvons définir le séparateur linéaire $f_{\vec{w},b}$ par l'équation suivante :

$$f_{\vec{w},b}(\vec{x}) = \langle \vec{w}.\vec{x} \rangle + b$$

Ce séparateur ne fournit pas des valeurs valant exclusivement -1 ou 1, mais nous dirons que quand le résultat $f_{\vec{w},b}(\vec{x})$ est positif, le vecteur $\vec{x}$ appartient à la même classe que les exemples d'étiquette 1, et que quand ce résultat est négatif, le vecteur $\vec{x}$ appartient à la même classe que les exemples d'étiquette -1.

Notons avant de disctuter plus avant cette notion de séparateur linéaire que l'équation $f_{\vec{w},b}(\vec{x}) = 0$ définit la frontière de séparation entre les deux classes, et que cette frontière est un hyperplan affine dans le cas du séparateur linéaire.

2.2 Séparabilité

Reprenons notre base d'exemples S, et décomposons-la en deux sous-ensembles selon la valeur de l'étiquette y. Définissons donc :

$$\begin{aligned} S^+ &= \{\vec{x} : (\vec{x}, y) \in S \text{ et } y = 1\} \\ S^- &= \{\vec{x} : (\vec{x}, y) \in S \text{ et } y = -1\} \end{aligned}$$

Dire que S est linéairement séparable signifie qu'il existe $\vec{w}$ et b tel que :

$$\begin{aligned} f_{\vec{w},b}(\vec{x}) > 0 \quad \forall \vec{x} \in S^+ \\ f_{\vec{w},b}(\vec{x}) < 0 \quad \forall \vec{x} \in S^- \end{aligned}$$

Ce n'est pas toujours faisable, il peut y avoir des distributions d'étiquettes sur les vecteurs de S qui rendent S non linéairement séparable. Dans le cas d'exemples pris dans le plan, dire que la distribution d'exemples est linéairement séparable signifie qu'on peut tracer un trait (un hyperplan donc) tel que les exemples de la classe 1 et ceux de la classe -1 se retrouvent de part et d'autre de cette frontière.

2.3 Marge

Supposons que S soit linéairement séparable pour ce qui suit. C'est une hypothèse forte, que l'on réduira par la suite, mais qui va nous permettre de poser quelques notions. L'idée au cœur des SVM est que, certes, il convient de séparer les exemples de chaque classe, mais qu'il faut que l'hyperplan passe « bien au milieu ». C'est pour définir cette notion de « bien au milieu » (cf. figure 2.1) que l'on introduit la marge.

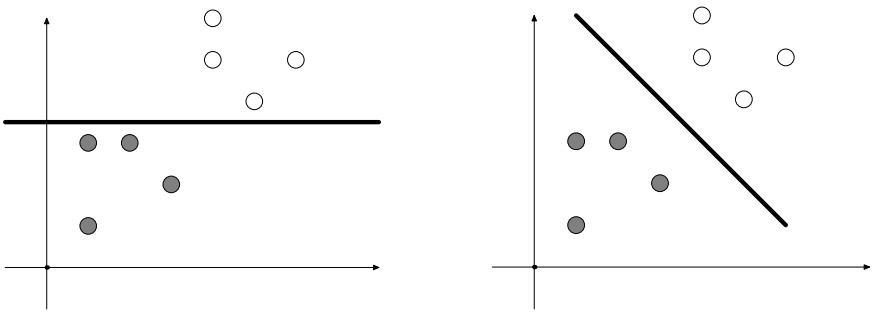

FIGURE 2.1 – Les mêmes exemples (la classe -1 ou 1 est représentée par la couleur) sont, à gauche comme à droite, séparés par une droite. La notion de marge sert à qualifier mathématiquement le fait que dans le cas de droite, les exemples sont « mieux » séparés par la droite.

Aidons nous de la figure 2.2 pour faire les observations suivantes. Notons déjà que les courbes d'équation $f_{\vec{w},b} = C$ sont des hyperplans parallèles, et que $\vec{w}$ est normal à ces hyperplans. Le paramètre b traduit un décalage de l'hyperplan séparateur, soit une translation des valeurs de $f_{\vec{w},b}$. La norme $\|\vec{w}\|$ de $\vec{w}$ influence les courbes de niveau $f_{\vec{w},b} = C$. Plus la $\|\vec{w}\|$ est élevée, plus les courbes de niveau sont serrées.

Si l'on souhaite une frontière séparatrice donnée, on se trouve confronté à une indétermination pour le choix de $\vec{w}$ et de b. N'importe quel vecteur $\vec{w}$ non nul perpendiculaire à l'hyperplan

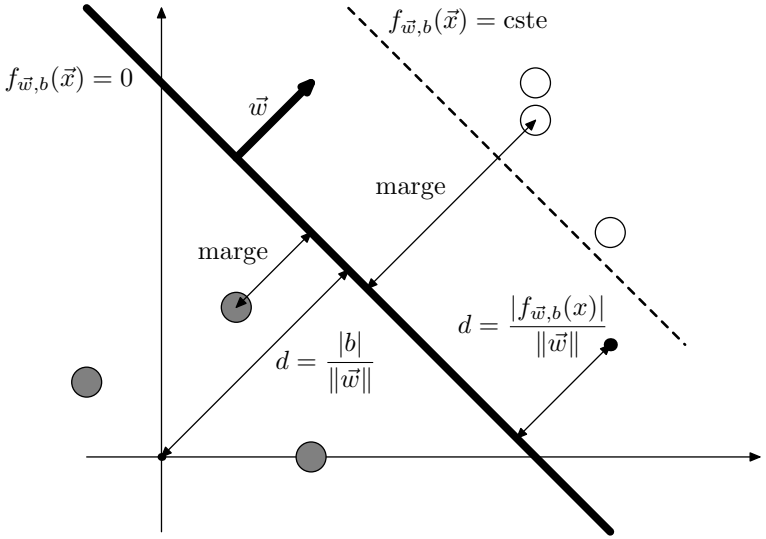

FIGURE 2.2 – Définition d'un séparateur $f_{\vec{w},b}$. Les grandeurs mentionnées sont les valeurs du séparateur au niveau des points, non les distances euclidiennes. Si la distance euclidienne d'un point à la frontière de séparation est d, $|f_{\vec{w},b}|$ en ce point vaut $d\,\|\vec{w}\|$.

convient. Une fois choisi, on détermine b tel que $b/\|\vec{w}\|$ soit la mesure orientée [1] de la distance de l'origine au plan de séparation.

La marge se définit comme une notion relative à un séparateur f et un ensemble d'exemples S donnés. On la notera γ_S^f cette marge. Elle se définit à partir d'une grandeur $\gamma_{(\vec{x},y)}^f$ calculée sur chaque exemple $(\vec{x}, y)$, appelée marge aussi, mais marge de l'exemple. Cette marge-là est la grandeur suivante :

$$\gamma_{(\vec{x},y)}^f = y \times f(\vec{x}) \tag{2.1}$$

Comme $y \in \{-1, 1\}$, et comme un séparateur met les exemples de classe 1 du côté positif de sa frontière, et ceux de classe -1 du côté négatif, la marge d'un exemple est, à la norme de $\vec{w}$ près, la distance de l'exemple à la frontière. La marge tout court, pour tout l'ensemble des exemples, est simplement le minimum des marges.

$$\gamma_S^f = \min_{(\vec{x},y)\in S} \gamma_{(\vec{x},y)}^f \tag{2.2}$$

On sent bien, pour revenir aux deux cas de la figure 2.1, que dans le cas de droite, le meilleur, la marge γ_S^f est plus grande, vu que la frontière passe plus loin des exemples. La maximisation de la marge est en effet l'activité majeure d'une SVM lors de la phase d'apprentissage.

1. La direction de $\vec{w}$ donne le sens positif.

Chapitre 3

Un problème d'optimisation

3.1 Le problème à résoudre par la SVM

3.1.1 Cas séparable

Continuons dans l'esprit du chapitre 2 en supposant toujours que l'on dispose d'une base d'exemples S effectivement séparable par un séparateur linéaire. Si le séparateur sépare effectivement S, avec du côté positif les exemples étiquetés 1, et du côté négatif ceux étiquetés -1, alors toutes les marges des exemples sont positives (cf. eq. 2.1). Si l'une des marges est négatives, c'est que le séparateur ne sépare pas correctement les deux classes, alors que c'est possible, vu qu'on suppose S linéairement séparable. Dans ce cas incorrect, γ_S^f est négatif (cf. eq. 2.2). Donc maximiser la marge, veut bien dire d'abord qu'on sépare (marge positive), puis qu'on sépare bien (marge maximale).

Le séparateur de marge maximal est tel que pour lui, l'exemple de plus petite marge a une marge plus grande que l'exemple de plus petite marge des autres séparateurs possibles.

Intéressons nous à cet exemple de plus petite marge. En fait, il peut y en avoir plusieurs (ex æquos), appartenant aussi bien à la classe 1 qu'à la classe -1. En fait, en réfléchissant un peu à l'aide de la figure 3.1, il y a forcément au moins un exemple de classe 1 et un exemple de clase -1 qui contraignent cette marge, et la frontière de séparation passe pile au milieu. On remarque aussi que seuls ces exemples-là contraignent l'hyperplan, et qu'on pourrait enlever tous les autres de la base sans que le séparateur de marge maximale change. C'est pourquoi ces exemples sont appelés *vecteurs support*.

Notons que l'hyperplan séparateur de marge maximale est définie à une constante près, et qu'on peut s'arranger pour que les vecteurs supports soient sur les courbes de niveau $+1$ et -1. La figure 3.2 illustre ce propos. Dans ce cas précis, la distance qui sépare les vecteurs supports au plan de séparation, la marge donc, est tout simplement $1/\|\vec{w}\|$.

Partant de ce constat, plaçons nous dans le cas où S est séparable. La figure 3.3 montre deux séparateurs, tels que tous les exemples se situent non seulement du bon côté ($\gamma_{(\vec{x},y)}^f > 0$), mais en dehors de la bande constituée par les courbes de niveau -1 et 1 ($\gamma_{(\vec{x},y)}^f > 1$).

La largeur de la bande constituée par les courbes de niveau -1 et 1 est $2/\|\vec{w}\|$. Pour trouver le séparateur de marge maximale, il suffit alors de chercher, parmi les séparateurs analogues à ceux de la figure 3.3, c'est-à-dire parmi ceux qui vérifient pour tous les exemples $\gamma_{(\vec{x},y)}^f > 1$, le séparateur pour lequel $\|\vec{w}\|$ est minimal. Minimiser $\|\vec{w}\|$ revient à élargir la bande $-1/1$ jusqu'à ce qu'elle « se coince » contre les vecteurs supports, ce qui nous ramène au cas de la figure 3.2.

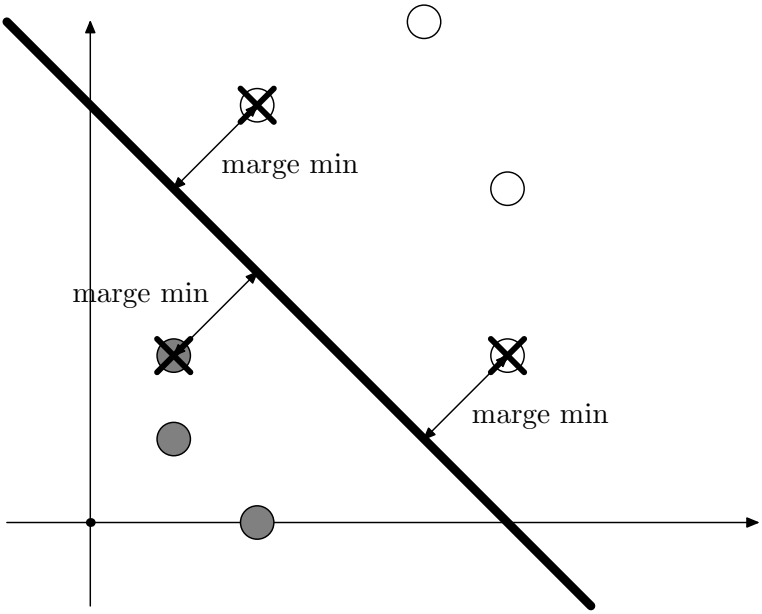

FIGURE 3.1 – Le séparateur de marge maximale a une frontière qui se définit par au moins un exemple de chaque classe. Les *vecteurs supports* sont marqués d'une croix.

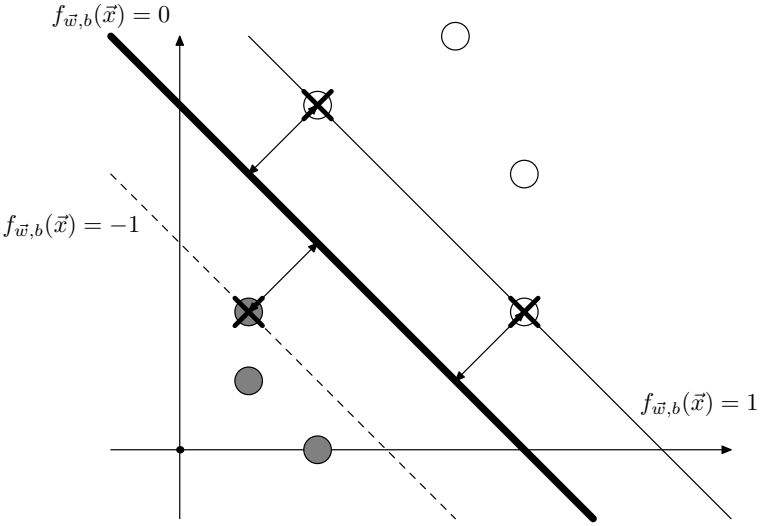

FIGURE 3.2 – Le séparateur de cette figure à la même frontière que celui de la figure 3.1, et sépare donc les exemples avec la même qualité. La différence est que les courbes de niveau $f_{\vec{w},b} = 1$ et $f_{\vec{w},b} = -1$ passent par les vecteurs supports. Il a fallu pour cela modifier la norme de $\vec{w}$, et réajuster b.

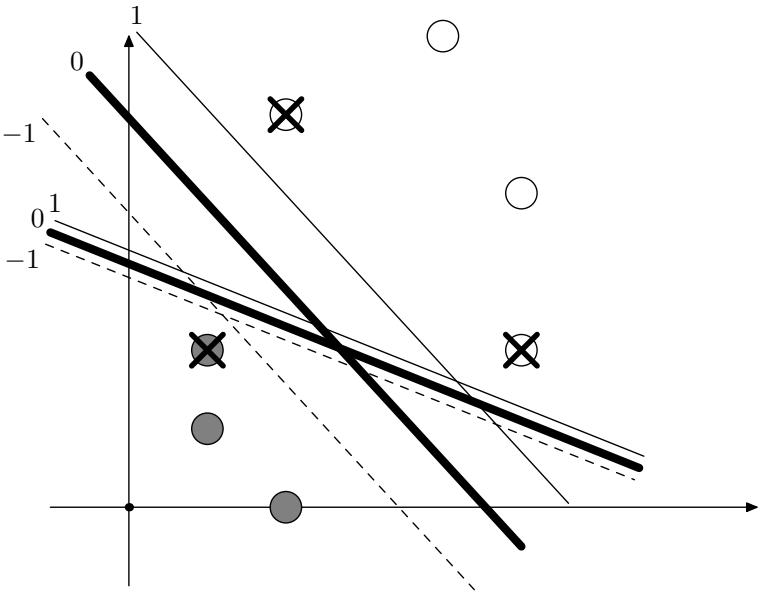

FIGURE 3.3 – Les deux séparateurs de cette figure vérifient que la marge de tous les exemples est supérieure à 1. Pour chacun d'eux, la largeur des bandes est $2/\|\vec{w}\|$, $\vec{w}$ étant le terme qui intervient dans $f_{\vec{w},b}(\vec{x}) = \langle \vec{w}.\vec{x} \rangle + b$. En conséquence, le séparateur « le plus vertical » sur cette figure a un vecteur $\vec{w}$ de norme plus faible que l'autre.

L'hyperplan de marge maximale, pour une base d'exemple $S = \{\vec{x}_l, y_l\}_l$, est donc la solution du problème d'optimisation suivant, dans lequel le coefficient $1/2$ est là juste pour simplifier les calculs de dérivée qui vont venir ultérieurement :

Jouer sur $\vec{w}$ et b pour minimiser	$\frac{1}{2} \langle \vec{w}.\vec{w} \rangle$
En respectant	$y_l \left(\langle \vec{w}.\vec{x}_l \rangle + b \right) \geq 1, \ \forall (\vec{x}_l, y_l) \in S$

3.1.2 Cas général

Pour le cas général où S n'est pas séparable, la solution consiste à autoriser certains exemples à avoir une marge plus petite que 1, voire négative. On va pour cela transformer la contrainte $y_l \left(\langle \vec{w}.\vec{x}_l \rangle + b \right) \geq 1$ par $y_l \left(\langle \vec{w}.\vec{x}_l \rangle + b \right) \geq 1 - \xi_l$, avec $\xi_l \geq 0$. Bien sûr, si on fait ça sans contrepartie, rien ne va plus car on peut minimiser $\langle \vec{w}.\vec{w} \rangle$ jusqu'à le rendre nul, en prenant des ξ_l suffisamment grands. L'idée est alors de rajouter les ξ_l dans l'expression à minimiser, de sorte à éviter qu'ils ne grandissent trop, ce qui limitera les exemples qui dérogeront à la séparabilité par le séparateur solution du problème d'optimisation. Ce problème est donc, pour résumer, le suivant, où C est un paramètre positif déterminant pour la tolérence de la SVM aux exemples mal séparés :

$$\begin{array}{rl} \text{Jouer sur } \vec{w},\, b \text{ et } \xi \text{ pour minimiser} & \frac{1}{2}\langle \vec{w}.\vec{w}\rangle + C\sum_l \xi_l \\ \text{En respectant} & \begin{cases} y_l\left(\langle \vec{w}.\vec{x}_l\rangle + b\right) \geq 1 - \xi_l,\ \forall (\vec{x}_l, y_l) \in S \\ \xi_l \geq 0,\ \forall l \end{cases} \end{array}$$

3.2 Résolution lagrangienne

3.2.1 Un problème convexe

Dans tout ce qui suit, on admettra que l'on parle d'un problème convexe, c'est-à-dire d'un problème d'optimisation qui n'admet pas d'optimum local, mais un seul optimum, global donc. Cette remarque, non justifiée dans ce document, est cruciale car la convexité du problème est une garantie de la convergence des SVM vers **la** solution optimale.

3.2.2 Problème direct

L'objet de ce document n'est pas d'introduire la théorie de l'optimisation, mais d'énoncer le strict minimum pour comprendre le lien avec les SVM. Dans la section 3.1.2, on definit un problème d'optimisation de la forme suivante :

$$\begin{array}{rl} \text{Jouer sur } \vec{k} \text{ pour minimiser} & f(\vec{k}) \\ \text{En respectant} & g_i(\vec{k}) \leq 0,\ 1 \leq i \leq n \end{array}$$

La résolution de ce problème passe par la définition de la fonction suivante, appelée le lagrangien du problème, qui fait intervenir les contraintes multipliées par des coefficients $\alpha_i \geq 0$, dits multiplicateurs de Lagrange. Les contraintes g_i sont affines.

$$L(\vec{k}, \vec{\alpha}) = f(\vec{k}) + \sum_{1 \leq i \leq n} \alpha_i g_i(\vec{k})$$

La théorie dit alors que le vecteur $\vec{k}^\star$ qui minimise $f(\vec{k})$ en respectant les contraintes doit vérifier que L est un point-selle en $(\vec{k}^\star, \vec{\alpha}^\star)$, minimum pour $\vec{k}$ et maximum pour $\vec{\alpha}$:

$$\forall \vec{k},\ \forall \vec{\alpha} \geq \vec{0},\ L(\vec{k}^\star, \vec{\alpha}) \leq L(\vec{k}^\star, \vec{\alpha}^\star) \leq L(\vec{k}, \vec{\alpha}^\star)$$

Il s'avère que l'on a alors à l'optimum

$$\frac{\partial L(\vec{k}^\star, \vec{\alpha}^\star)}{\partial \vec{k}} = \vec{0}$$

alors que $\partial L(\vec{k}^\star, \vec{\alpha}^\star)/\partial \vec{\alpha}$, qui devrait être nul également au point-selle, peut ne pas être défini (voir l'encadré de la figure 3.4)

Ces conditions sont suffisantes pour définir l'optimum si le lagrangien est une fonction convexe, ce qui sera notre cas pour les SVM. Voir (Cristanini and Shawe-Taylor, 2000) pour plus de justifications mathématiques.

Le problème est qu'écrire ces conditions ne conduit pas toujours facilement à la résolution du problème. Dans le cas des SVM, la démarche sera facilitée par la résolution du problème dual.

3.2.3 Problème dual

le problème dual s'obtient en injectant les contraintes d'optimalité données par les $\frac{\partial L}{\partial \vec{k}} = 0$ dans l'expression de L, ce qui ne laisse apparaître que les multiplieurs, dans une expression à maximiser cette fois-ci, sous d'autres contraintes.

En effet, le lagrangien est une fonction en scelle de cheval, et l'optimum est un minimum suivant $\vec{k}$, mais un maximum suivant $\vec{\alpha}$ (cf. figure 3.4). Réinjecter $\frac{\partial L}{\partial \vec{k}} = 0$ dans $L(\vec{k}, \vec{\alpha})$ revient à définir la fonction $\theta(\vec{\alpha})$ qui pour $\vec{\alpha}$ calcule la valeur minimale du lagrangien, c'est-à-dire la valeur minimale de L obtenue quand pour ce $\vec{\alpha}$-là, on minimise en jouant sur $\vec{k}$. Reste alors à maximiser $\theta(\vec{\alpha})$ en jouant sur $\vec{\alpha}$, ce qui définit le problème d'optimisation dual.

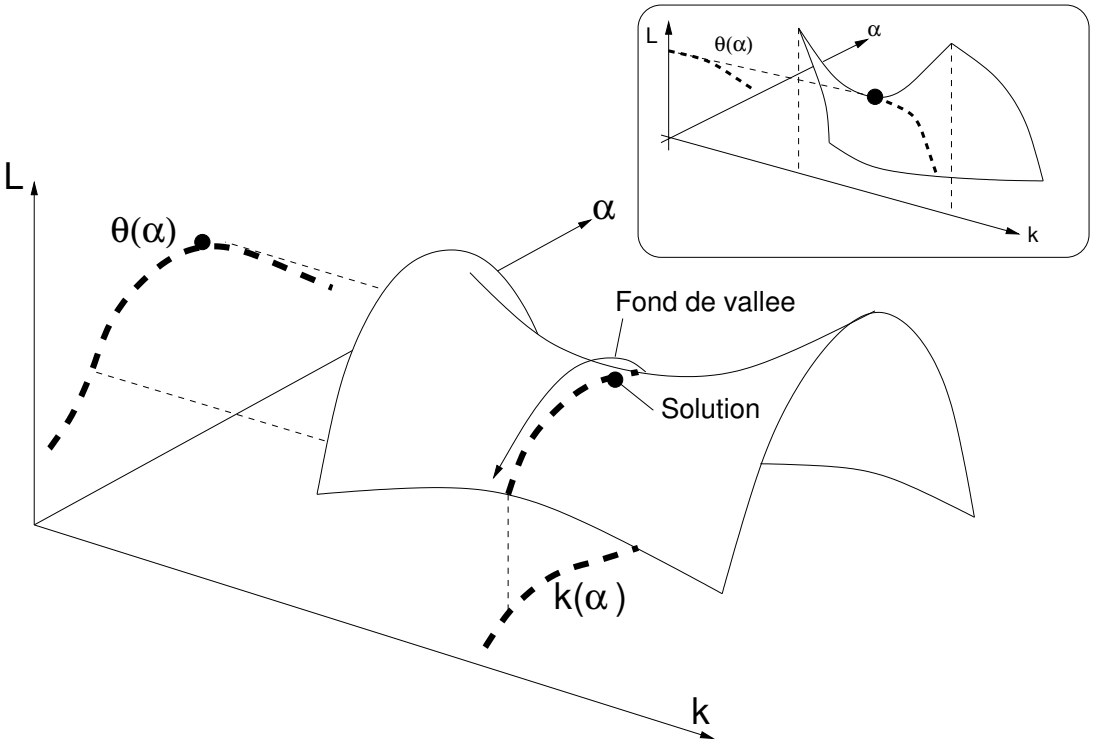

FIGURE 3.4 – Passage du problème primal au problème dual. Le lagrangien $L(\vec{k}, \vec{\alpha})$ a une forme de selle autour de la solution du problème. Le « fond de vallée », c'est-à-dire les minima suivant $\vec{k}$, est représenté sur la scelle par des pointillés. L'équation $\frac{\partial L}{\partial \vec{k}} = 0$ permet de lier $\vec{k}$ et $\vec{\alpha}$, de sorte à exprimer $\vec{k}$ en fonction de $\vec{\alpha}$. Cette liaison est la projection de la vallée sur le plan « horizontal ». L'injection de cette relation dans L donne une fonction $L(\vec{k}, \vec{k}(\vec{\alpha})) = \theta(\vec{\alpha})$. Cette fonction est la fonction objectif du problème dual, dont on cherche le maximum, comme l'illustre sa représentation sur la figure.

L'intérêt du problème dual dans le cadre des SVM apparaîtra plus clairement si on quitte les généralités de la théorie de l'optimisation pour revenir à nos moutons.

3.2.4 Vue intuitive de l'optimisation sous contrainte

Ce paragraphe vous donnera une vue intuitive des théorèmes d'optimisation sous contrainte. Vous pouvez toutefois admettre ces théorèmes et passer directement à la section suivante. Je remercie Arnaud Golinvaux [1] pour m'avoir expliqué ce qui suit.

Optimisation avec contraintes d'égalité

Partons du problème d'optimisation suivant, en nous aidant de la figure 3.5.

Jouer sur $\vec{k}$ pour minimiser	$f(\vec{k})$
En respectant	$\vec{g}(\vec{k}) = \vec{0}$ où $\vec{g}(\vec{k}) = \left(g_i(\vec{k})\right)_{1 \leq i \leq n}$

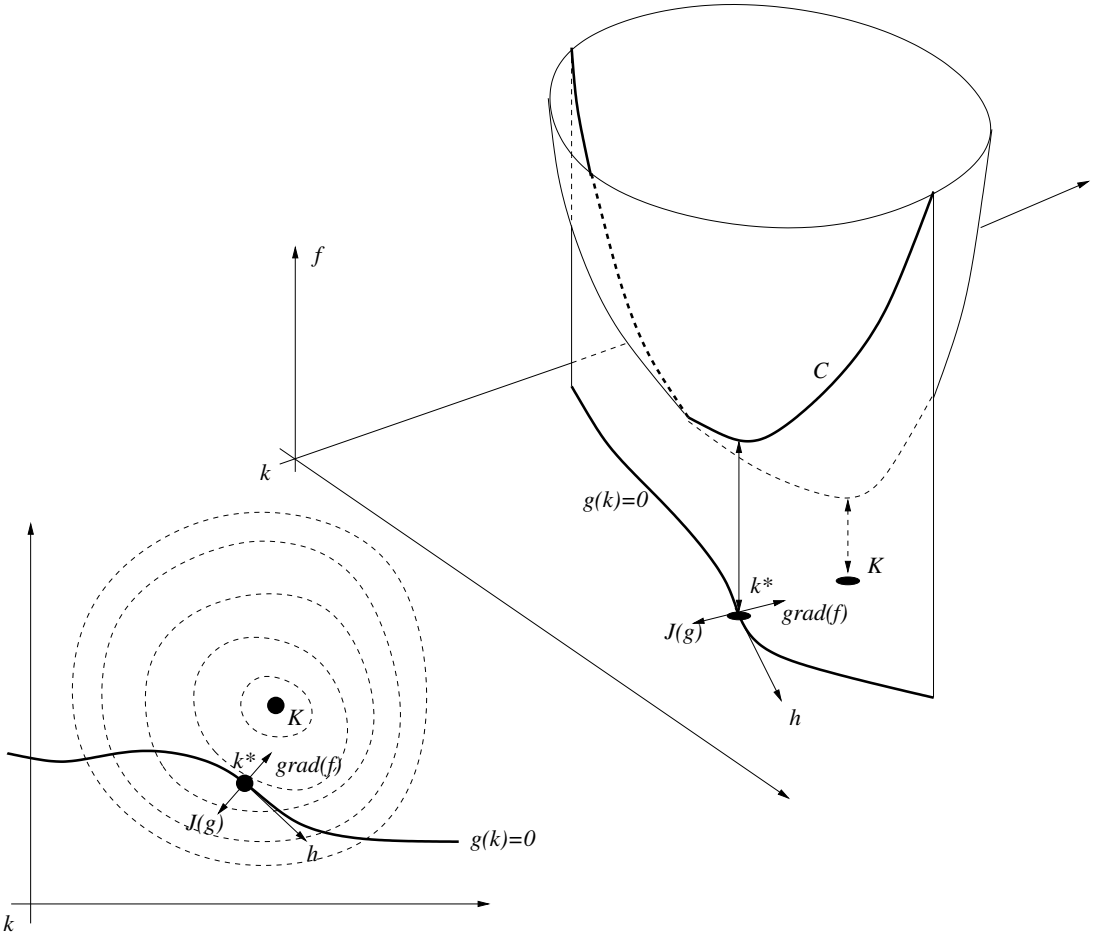

FIGURE 3.5 – Voir le texte.

1. Élève de troisième année à Supélec en 2012.

Plaçons nous à la solution $\vec{k}^\star$ de notre problème. Cette solution, du fait des contraintes, peut être différente du minimum K de la fonction objectif (voir la figure 3.5). Les contraintes $\vec{g}$ sont une fonction des paramètres $\vec{k}$. L'idée est que l'on ne cherche $\vec{k}^\star$ que dans le noyau de $\vec{g}$, i.e. parmi les $\vec{k}$ tels que $\vec{g}(\vec{k}) = \vec{0}$.

Rester dans ce noyau (i.e. sur la courbe en gras sur la figure) a la conséquence suivante autour de la solution $\vec{k}^\star$. Soit $\vec{h}$ un déplacement élémentaire autour de l'optimum, tel que $\vec{k}^\star + \vec{h}$ reste dans le noyau de $\vec{g}$, on a :

$$\begin{aligned} \vec{g}(\vec{k}^\star) &= 0 \\ \vec{g}(\vec{k}^\star + \vec{h}) &= 0 \\ \vec{g}(\vec{k}^\star + \vec{h}) &= \vec{g}(\vec{k}^\star) + \mathrm{J}\vec{g}|_{\vec{k}} \left(\vec{k}^\star\right).\vec{h} \end{aligned}$$

Avec $\mathrm{J}g|_k\,(k_0)$ la notation de la matrice jacobienne de g par rapport à k, prise en k_0. On en déduit immédiatement

$$\mathrm{J}\vec{g}|_{\vec{k}} \left(\vec{k}^\star\right).\vec{h} = 0$$

Un déplacement $\vec{h}$ autour de la solution, qui respecte les contraintes, est donc inclus dans l'espace vectoriel que constitue le noyau de la matrice jacobienne :

$$\vec{h} \in \mathrm{Ker}\Big(\mathrm{J}\vec{g}|_{\vec{k}} \left(\vec{k}^\star\right)\Big)$$

Regardons maintenant ce que donne ce déplacement vis-à-vis de notre fonction objectif f. On a, en linéarisant autour de $\vec{k}^\star$, et en utilisant le déplacement $\vec{h}$ ci-dessus qui respecte les contraintes

$$f(\vec{k}^\star + \vec{h}) = f(\vec{k}^\star) + \nabla f|_{\vec{k}} \left(\vec{k}^\star\right).\vec{h}$$

Pour rappel, le gradient est la jacobienne dégénérée quand la fonction est scalaire et non vectorielle. Le fait d'être autour de $\vec{k}^\star$ signifie que $f(\vec{k}^\star)$ est minimale, tant que nos déplacements respectent les contraintes (ce qu'on voit en gras sur figure 3.5, i.e. la courbe C). Donc $\nabla f|_{\vec{k}} \left(\vec{k}^\star\right).\vec{h} >= 0$. Or tout comme $\vec{h}$, on a aussi $-\vec{h}$ qui respecte les contraintes, car l'ensemble $\mathrm{Ker}\Big(\mathrm{J}\vec{g}|_{\vec{k}} \left(\vec{k}^\star\right)\Big)$ des $\vec{h}$ qui respectent les contraintes est le noyau d'une matrice, donc un espace vectoriel. Donc on a $-\nabla f|_{\vec{k}} \left(\vec{k}^\star\right).\vec{h} >= 0$ également. On en déduit que

$$\nabla f|_{\vec{k}} \left(\vec{k}^\star\right).\vec{h} = 0, \forall \vec{h} \in \mathrm{Ker}\Big(\mathrm{J}\vec{g}|_{\vec{k}} \left(\vec{k}^\star\right)\Big)$$

Dit autrement, $\nabla f|_{\vec{k}} \left(\vec{k}^\star\right)$ est dans le sous espace vectoriel E perpendiculaire à $\mathrm{Ker}\Big(\mathrm{J}\vec{g}|_{\vec{k}} \left(\vec{k}^\star\right)\Big)$. Or cet espace E est justement celui engendré par les vecteurs colonne de la matrice $\mathrm{J}\vec{g}|_{\vec{k}} \left(\vec{k}^\star\right)$. On peut donc affirmer que $\nabla f|_{\vec{k}} \left(\vec{k}^\star\right)$ est une combinaison linéaire de ces vecteurs colonne.

Or comme $\vec{g}$ est le vecteur des n contraintes scalaires g_i,

$$\mathrm{J}\vec{g}|_{\vec{k}} \left(\vec{k}^\star\right) = \left[\nabla g_1|_{\vec{k}} \left(\vec{k}^\star\right), \nabla g_2|_{\vec{k}} \left(\vec{k}^\star\right) \cdots \nabla g_n|_{\vec{k}} \left(\vec{k}^\star\right)\right]$$

donc ces vecteurs colonnes sont les gradients de chacune des contraintes, par rapport aux paramètres. Il résulte de ça que

$$\exists(\alpha_1, \cdots, \alpha_n) \in \mathbb{R} : \nabla f|_{\vec{k}} \left(\vec{k}^\star\right) + \sum_{i=1}^{n} \alpha_i \nabla g_i|_{\vec{k}} \left(\vec{k}^\star\right) = 0$$

D'où l'idée de poser comme lagrangien

$$L(\vec{k}, \vec{\alpha}) = f(\vec{k}) + \sum_{i=1}^{n} \alpha_i g_i(\vec{k})$$

dont le gradient par rapport à $\vec{k}$ doit être nul à l'optimum contraint.

Cas des contraintes d'inégalité

Passons maintenant au problème d'optimisation suivant.

Jouer sur $\vec{k}$ pour minimiser	$f(\vec{k})$
En respectant	$g_i(\vec{k}) \leq 0,\ 1 \leq i \leq n$

L'idée est d'associer à chaque contrainte $g_i(\vec{k}) <= 0$ un nouveau paramètre scalaire y_i. On regroupe les y_i des contraintes d'inégalité au sein d'un vecteur $\vec{y}$. On pose $g'_i(\vec{k}, \vec{y}) = g_i(\vec{k}) + y_i^2$. Un problème d'optimisation avec contraintes d'inégalité devient alors un problème avec contraintes d'égalités, incluant des paramètres en plus.

Jouer sur $\vec{k}$ et $\vec{y}$ pour minimiser	$f(\vec{k})$
En respectant	$\vec{g'}(\vec{k}, \vec{y}) = \vec{0}$

L'astuce, c'est que bien sûr les nouveaux paramètres $\vec{y}$ n'influencent pas l'objectif $f(\vec{k})$. Attention, ce qu'on a dit précédemment sur les paramètres $\vec{k}$ est à appliquer sur $\vec{k}$ et sur $\vec{y}$ dans ce nouveau problème. Selon ce qui précède, on peut poser le lagrangien suivant :

$$L(\vec{k}, \vec{y}, \vec{\alpha}) = f(\vec{k}) + \sum_{i=1}^{n} \alpha_i (g_i(\vec{k}) + y_i^2)$$

Le gradient du lagrangien par rapport aux paramètres ($\vec{k}$ et $\vec{y}$ maintenant) doit être nul. Il est donc nul si on le considère suivant $\vec{k}$ et suivant $\vec{y}$.

En dérivant suivant $\vec{k}$, on a toujours, comme précédemment

$$\nabla f|_{\vec{k}} \left(\vec{k}^\star\right) + \sum_{i=1}^{n} \alpha_i \ \nabla g_i|_{\vec{k}} \left(\vec{k}^\star\right) = 0$$

En dérivant suivant y_i, on a $2\alpha_i y_i = 0$. On a alors soit $\alpha_i = 0$, soit $y_i = 0$. Or $y_i = 0$ est équivalent, par définition des nouvelles contraintes g', à $g_i(\vec{k}) = 0$.... et le $\vec{k}$ dont on parle, c'est celui défini par l'autre dérivée du lagrangien, c'est donc $\vec{k}^\star$. C'est pourquoi à l'optimum, on a $\alpha_i g_i(k^\star) = 0$ (ce sont les condition KKT que l'on utilisera plus loin). Ces conditions sont importantes, puisque la valeur de α_i à l'optimum permet de savoir si la contrainte g_i est saturée ou non. Un α_i non nul correspond à une contrainte saturée.

La dérivée seconde du lagrangien selon y_i est justement α_i. Intuitivement [2], si l'on reprend la courbe C de la figure 3.5, et si l'on garde en mémoire que les paramètres y_i font partie des

2. Pour faire les choses correctement, reportez-vous à un livre de mathématiques.

paramètres $\vec{k}$ de cette figure, on voit bien que la courbure de C est positive (vers le haut), et que donc la dérivée seconde est positive. D'où $\alpha_i >= 0$.

En pratique, nous avons fini par supprimer toute mention aux y_i dans l'expression des conditions qui nous intéressaient. On peut donc éviter de faire apparaître ces y_i dans le lagrangien, et garder pour notre problème, alors qu'il contient des contraintes sous forme d'inégalités :

$$
\begin{aligned}
& L(\vec{k}, \vec{\alpha}) = f(\vec{k}) + \sum_{i=1}^{n} \alpha_i g_i(\vec{k}) \\
& \nabla L|_{\vec{k}} \left(\vec{k}^\star, \vec{\alpha}\right) = 0 \\
& \forall i,\ \alpha_i \geq 0 \\
& \forall i,\ \alpha_i g_i(k^\star) = 0
\end{aligned}
$$

3.2.5 Retour au cas particulier des SVM

Revenons au problème défini à la section 3.1.2, et notons α_l et μ_l les coefficients relatifs aux deux types de contraintes. Après avoir réécrit ces contraintes pour qu'elles fassent apparaître des $\leq$, de sorte à coller à la forme vue en section 3.2.2, on peut définir le lagrangien de notre problème par :

$$
\begin{aligned}
L(\vec{w}, b, \vec{\xi}, \vec{\alpha}, \vec{\mu}) &= \frac{1}{2} \langle \vec{w}.\vec{w} \rangle + C \sum_l \xi_l - \sum_l \alpha_l \left(y_l \left(\langle \vec{w}.\vec{x}_l \rangle + b \right) + \xi_l - 1 \right) - \sum_l \mu_l \xi_l \\
&= \frac{1}{2} \langle \vec{w}.\vec{w} \rangle + \sum_l \xi_l \left(C - \alpha_l - \mu_l \right) + \sum_l \alpha_l - \sum_l \alpha_l y_l \left(\langle \vec{w}.\vec{x}_l \rangle + b \right) \\
& \forall l,\ \alpha_l \geq 0 \\
& \forall l,\ \mu_l \geq 0
\end{aligned}
$$

Les deux types de contraintes de notre problème sont des inégalités (cf. section 3.1.2). La théorie dit alors que si la contrainte est saturée, c'est-à-dire si c'est une égalité en fait, alors son multiplieur est non nul. Dans le cas où c'est une inégalité stricte, son multiplieur est nul. Donc pour une contrainte $g_i(...) \leq 0$ dont le multiplieur associé serait k_i, on a soit $k_i = 0$ et $g_i(...) < 0$, soit $k_i > 0$ et $g_i(...) = 0$. Ces deux cas se résument en une seule expression, $k_i g_i(...) = 0$. Cette expression est appelée condition supplémentaire de KKT [3]. Dans notre problème, on peut alors exprimer 6 conditions KKT, à savoir les contraintes, la positivité des multiplieurs, et les conditions KKT supplémentaires.

$$
\begin{aligned}
& \forall l,\ \alpha_l \geq 0 && \text{(KKT1)} \\
& \forall l,\ \mu_l \geq 0 && \text{(KKT2)} \\
& \forall l,\ \xi_l \geq 0 && \text{(KKT3)} \\
& \forall l,\ y_l \left(\langle \vec{w}.\vec{x}_l \rangle + b \right) \geq 1 - \xi_l && \text{(KKT4)} \\
& \forall l,\ \mu_l \xi_l = 0 && \text{(KKT5)} \\
& \forall l,\ \alpha_l \left(y_l \left(\langle \vec{w}.\vec{x}_l \rangle + b \right) + \xi_l - 1 \right) = 0 && \text{(KKT6)}
\end{aligned}
$$

Ceci posé, annulons les dérivées partielles du lagrangien suivant les termes qui ne sont pas des multiplieurs de lagrange.

$$
\frac{\partial L}{\partial \vec{w}} = \vec{0} \Rightarrow \vec{w} = \sum_l \alpha_l y_l \vec{x}_l \qquad \text{(L1)}
$$

3. Karush-Kuhn-Tucker

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_l \alpha_l y_l = 0 \qquad \text{(L2)}$$

$$\frac{\partial L}{\partial \xi_l} = \vec{0} \Rightarrow \forall l,\ C - \alpha_l - \mu_l = 0 \qquad \text{(L3)}$$

L'équation L1, réinjectée dans l'expression du lagrangien, permet de faire disparaître le terme $\vec{w}$. L'équation L3 permet de faire disparaître les μ_l. L2 permet également d'éliminer b, qui se retrouve dans L facteur d'un terme nul. Il nous reste après ces substitutions une expression du lagrangien qui ne dépend que des α_l, qu'il nous faut maximiser en jouant sur ces α_l, sachant que l'injection de L1, L2, et L3 nous place déjà sur un minimum vis-à-vis de $\vec{w}$, $\vec{\xi}$ et b. C'est ça le problème dual. Les contraintes de ce problème s'infèrent des contraintes sur les α_l données par les équations KKTi. En utilisant L3, KKT2, KKT3, on peut montrer ce qui suit, en traîtant les deux cas induits par KKT5 :

- soit $\xi_l = 0, \mu_l > 0$, et alors $0 \leq \alpha_l \leq C$,
- soit $\xi_l > 0, \mu_l = 0$, et alors $\alpha_l = C$ selon L3.

Les contraintes sur les α_l sont donc $0 \leq \alpha_l \leq C$ et L2. On doit donc résoudre le problème d'optimisation suivant, dual de notre problème de départ, pour trouver les multiplieurs de Lagrange α_l.

$$\text{Jouer sur } \vec{\alpha} \text{ pour maximiser} \quad \sum_l \alpha_l - \frac{1}{2} \sum_k \sum_l \alpha_k \alpha_l y_k y_l \langle \vec{x}_k . \vec{x}_l \rangle$$

$$\text{En respectant} \quad \begin{cases} \forall l,\ \sum_l \alpha_l y_l = 0 \\ \forall l,\ 0 \leq \alpha_l \leq C \end{cases}$$

Un remarque en passant, des deux cas précédemment étudiées, on peut aussi déduire que $\xi_l(\alpha_l - C) = 0$. Cela signifie que tolérer un exemple x_l mal séparé ($\xi_l \neq 0$) revient à utiliser son α_l avec la valeur maximale C.

La formulation de ce problème dual a cela d'intéressant qu'elle ne fait intervenir que les exemples x_l, voire plus précisément leurs produits scalaires uniquement. Nous y reviendrons. De plus, le vecteur de l'hyperplan de séparation étant défini par L1, il est constitué de la contribution de tous les exemples x_l, à concurrence d'une valeur α_l. Or celle-ci, après optimisation, peut s'avérer nulle pour bien des exemples, qui n'entrent ainsi pas en compte pour définir le séparateur. Ceux qui restent, c'est-à-dire ceux pour lesquels α_l est non nul, sont appelés vecteurs supports, car ce sont eux qui déterminent l'hyperplan séparateur.

Résoudre le problème dual n'est pas trivial, nous n'avons fait ici que le poser. En particulier, b a disparu du problème dual, et il faut pas mal ruser [4] pour le retrouver une fois ce problème résolu. Nous y reviendrons au chapitre 5. Terminons ce chapitre par un exemple de séparation linéaire, où le séparateur est solution du problème d'optimisation que nous avons posé. Les exemples qui rentrent effectivement en compte dans la formule L1 avec un coefficient α_l non nuls, à savoir les vecteurs supports, sont marqués d'une croix (cf. figure 3.6).

4. Voir paragraphe 5.1.2 page 32.

La séparation est donc définie par l'équation suivante :

$$f(\vec{x}) = \left\langle \left(\sum_l \alpha_l y_l \vec{x}_l \right) . \vec{x} \right\rangle + b$$

que l'on préférera écrire comme suit, pour ne faire intervenir les vecteurs qu'au travers de produits scalaires.

$$f(\vec{x}) = \sum_l \alpha_l y_l \langle \vec{x}_l . \vec{x} \rangle + b \tag{3.1}$$

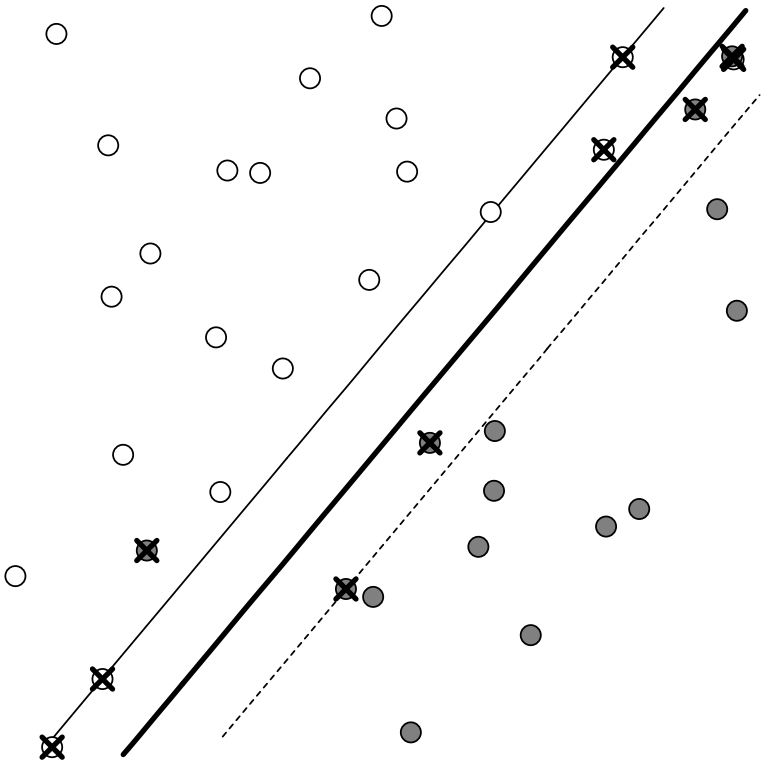

FIGURE 3.6 – Hyperplan résultat de la résolution du problème d'optimisation de la section 3.1.2. La frontière de séparation $f(\vec{x}) = 0$ est la ligne épaisse, les courbes $f(\vec{x}) = 1$, ligne continue fine, et $f(\vec{x}) = -1$, ligne en pointillés, sont représentées également. Les *vecteurs supports* sont marqués d'une croix.

Chapitre 4

Noyaux

Le gros intérêt des noyaux [1] est que tout ce qu'on vient de voir sur la séparation linéaire s'applique en fait très facilement à des séparations non linéaires, sous réserve de bien faire les choses.

4.1 L'espace des caractéristiques

Imaginons un ensemble d'exemples x_l étiquetés par -1 ou 1 suivant la classe à laquelle ils appartiennent, qui ne soit pas du tout séparable linéairement. La méthode vue au chapitre précédent fonctionne, mais la séparation est bien entendu de piètre qualité, et bon nombre de vecteurs sont des supports (cf. figure 4.1).

Une solution pour mieux séparer les exemples est de les projeter dans un espace différent [2], et de réaliser une séparation linéaire dans cet espace-là, où cette fois-ci elle devrait-être plus adaptée.

Soit Φ cette projection, on a :

$$\Phi(\vec{x}) = \begin{pmatrix} \phi_1(\vec{x}) \\ \phi_2(\vec{x}) \\ \phi_3(\vec{x}) \\ \vdots \\ \phi_n(\vec{x}) \end{pmatrix}$$

et bien sûr, les fonctions ϕ_i ne sont pas nécessairement linéaires, et on peut même avoir $n = \infty$! Si l'on reprend les méthodes du chapitre 3, et que l'on travaille dans l'espace des caractéristiques [3], c'est-à-dire si on travaille avec le corpus

$$\bar{S} = \{(\phi(\vec{x}_l), y_l)\}_{1 \leq l \leq p} \text{ avec } \forall l,\ y_l \in \{-1, 1\}$$

au lieu de

$$S = \{(\vec{x}_l, y_l)\}_{1 \leq l \leq p} \text{ avec } \forall l,\ y_l \in \{-1, 1\}$$

on se retrouve à faire de la séparation linéaire sur le corpus $\bar{S}$. On obtient un séparateur, donné par la formule L1 et le b. Pour décider de la classe d'un vecteur $\vec{x}$, on pourrait calculer alors

1. kernel en anglais.
2. Souvent de dimension très élevée.
3. Traduction personnelle de l'anglais *feature space*.

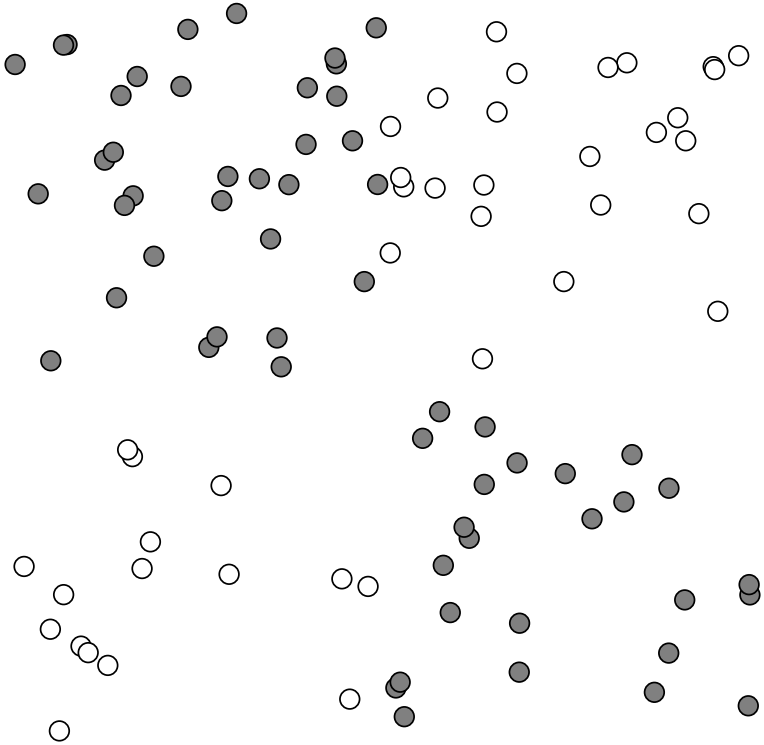

FIGURE 4.1 – Exemples non séparables linéairement.

$\Phi(\vec{x})$, que l'on passerait à son tour comme argument du séparateur pour en connaître la classe, $+1$ où -1.

En fait, on ne procédera pas comme ça, et on préférera éviter le calcul explicite de $\Phi(\vec{x})$ en remarquant que le problème d'optimisation posé au chapitre 3 ne fait intervenir les vecteurs que via des produits scalaires entre eux.

Notons $\mathbf{k}(\vec{x},\vec{z})$ le produit $\langle \Phi(\vec{x}).\Phi(\vec{z})\rangle$. Travailler sur le corpus $\bar{S}$ revient à travailler sur le corpus S avec les algorithmes du chapitre 3, mais en remplaçant toutes les occurrences de $\langle \bullet.\bullet\rangle$ par $\mathbf{k}(\bullet,\bullet)$.

Pour l'instant, on ne voit pas trop l'intérêt, vu que pour calculer $\mathbf{k}(\vec{x},\vec{z})$, il faut appliquer la définition, à savoir projeter $\vec{x}$ et $\vec{z}$ dans l'espace des caractéristiques et calculer le produit scalaire, dans cet espace, des deux vecteurs obtenus.

La ruse en fait est qu'on ne fera pas cette projection, car on calculera $\mathbf{k}(\vec{x},\vec{z})$ autrement. En fait, $\mathbf{k}(\vec{x},\vec{z})$ est une fonction que l'on va se donner, en s'assurant qu'il existe bien en théorie une projection Φ dans un espace qu'on ne cherchera pas à décrire. Ainsi, on calculera directement $\mathbf{k}(\vec{x},\vec{z})$, à chaque fois que l'algorithme du chapitre 3 requiert un produit scalaire, et c'est tout ! La projection dans le gros espace de caractéristiques sera implicite.

Prenons un exemple. Posons

$$\mathbf{k}(\vec{x},\vec{z}) = \exp\left(-\frac{\|\vec{x}-\vec{z}\|^2}{2\sigma}\right)$$

Il est connu que cette fonction correspond au produit scalaire des projetés de $\vec{x}$ et $\vec{z}$ dans un espace de dimension infinie. L'algorithme d'optimisation, qui utilisera cette fonction [4], réalisera une séparation linéaire, en maximisant la marge, dans cet espace, sans qu'un boucle infinie ne soit invoquée pour calculer les produits scalaires en multipliant deux à deux les composantes des vecteurs projetés !

La fonction de séparation est alors directement inspirée de l'équation 3.1 page 19, une fois les α_l optimaux trouvés et b calculé,

$$f(\vec{x}) = \sum_l \alpha_l y_l \mathbf{k}(\vec{x}_l,\vec{x}) + b$$

sachant que nombre de termes de la somme sont nuls si le problème est bien séparable. Dans notre cas, ce séparateur peut se réécrire

$$f(\vec{x}) = \sum_l \alpha_l y_l \exp\left(-\frac{\|\vec{x}_l-\vec{x}\|^2}{2\sigma}\right) + b$$

Les valeurs $f(\vec{x}) = 0$ définissent la frontière de séparation entre les classes, et les valeurs $f(\vec{x}) = 1$ et $f(\vec{x}) = -1$ permettent de représenter la marge. La figure 4.2 donne le résultat de l'algorithme du chapitre 3 avec notre fonction noyau.

4.2 Quelles fonctions sont des noyaux ?

Il serait bien sûr trop beau que n'importe quelle fonction de deux vecteurs puisse être un noyau, il faut en effet pour que ça marche qu'il existe une projection dans un espace de caractéristiques dont la fonction donne le même résultat que le produit scalaire des projetés.

4. fonction que l'on appelle noyau.

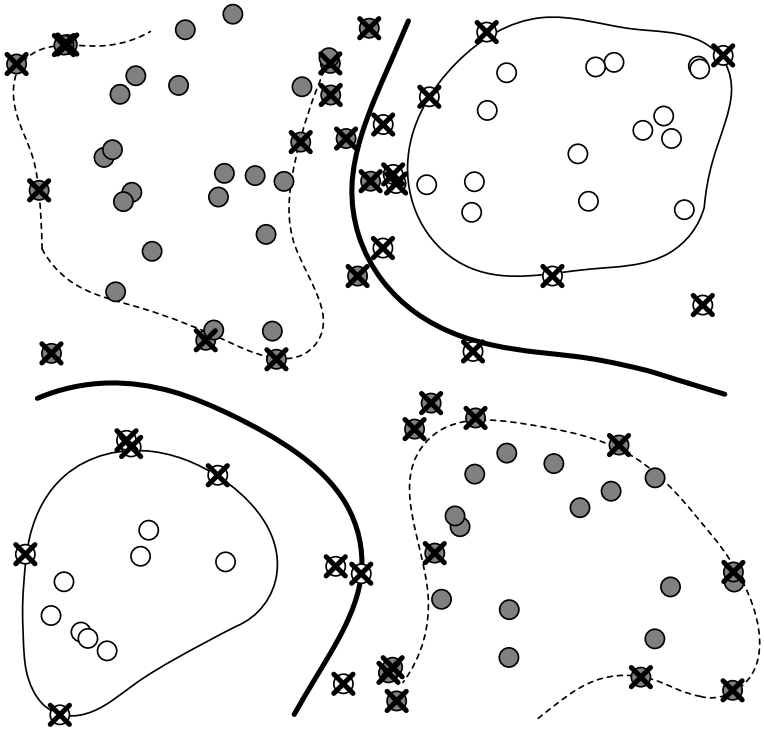

FIGURE 4.2 – Résolution du problème d'optimisation de la section 3.1.2 sur le corpus de la figure 4.1, mais avec un noyaux gaussien. La frontière de séparation $f(\vec{x}) = 0$ est la ligne épaisse, les courbes $f(\vec{x}) = 1$ et $f(\vec{x}) = -1$ sont représentées également. Les *vecteurs supports* sont marqués d'une croix.

Bien qu'on ne soit pas obligé de réaliser cette projection, comme nous l'avons vu, il faut s'assurer de son existence. Il en va par exemple de la convergence de l'algorithme des SVM, car cette convergence n'est assurée que si le problème est convexe, comme évoqué au paragraphe 3.2.1, ce qui requiert que le noyau ne soit pas n'importe quoi.

4.2.1 Exemple simple

On s'intéresse à la fonction suivante pour comparer deux vecteurs :

$$\mathbf{k}\left(\vec{x},\vec{z}\right) = \left(\langle\vec{x}.\vec{z}\rangle + c\right)^2$$

Est-ce un noyau ? Si oui, quelle est la projection qui lui correspond ? Une façon de le montrer, c'est de faire apparaître le produit scalaire des projetés.

$$\begin{aligned}\left(\langle\vec{x}.\vec{z}\rangle + c\right)^2 &= \left(\sum_i x_i z_i + c\right)^2 \\ &= \sum_{i,j} x_i z_i x_j z_j + 2c\sum_i x_i z_i + c^2 \\ &= \sum_{i,j}(x_i x_j)(z_i z_j) + \sum_i(\sqrt{2c}x_i)(\sqrt{2c}z_i) + (c)(c) \qquad (4.1)\end{aligned}$$

On en conclut que la projection, dont notre noyau correspond au produit scalaire, combine 2 à 2 les composantes du vecteur $\vec{x}$:

$$\Phi\left(\vec{x}\right) = \begin{pmatrix} x_1^2 \\ x_1 x_2 \\ x_1 x_3 \\ \cdots \\ x_n x_{n-1} \\ x_n^2 \\ \sqrt{2c}.x_1 \\ \sqrt{2c}.x_2 \\ \cdots \\ \sqrt{2c}.x_n \\ c \end{pmatrix}$$

4.2.2 Conditions pour avoir un noyau

Il y a des conditions mathématiques, appelées théorème de Mercer, qui permettent de dire si une fonction est un noyau ou non, sans construire la projection dans l'espace des caractéristiques. En fait, il faut assurer que pour tout ensemble d'exemples de longueur l, la matrice $\left(\mathbf{k}\left(\vec{x}_i,\vec{x}_j\right)\right)_{1\leq i,j\leq l}$ soit définie positive. Nous ne nous attardons pas d'avantage sur ce point, préférant construire des noyaux à partir de noyaux existants.

4.2.3 Noyaux classiques

Nous mentionnons ici deux noyaux classiquement utilisés. Le premier d'entre eux est le noyau polynômial suivant :

$$\mathbf{k}_d\left(\vec{x},\vec{z}\right) = \left(\langle\vec{x}.\vec{z}\rangle + c\right)^d$$

et correspond à une projection $\Phi(\vec{x})$ dans un espace de caractéristiques où chaque composantes $\phi_i(\vec{x})$ est un monôme de degré inférieur à d de certaines des composantes de $\vec{x}$. Le séparateur calculé avec ce noyau est un polynôme de degré d dont les monômes sont les composantes de $\vec{x}$. La constante c, quand elle est élevée, donne plus d'importance aux monômes de degré élevé. Avec $c = 1$ et $d = 3$, la figure 4.3 montre le résultat de la séparation.

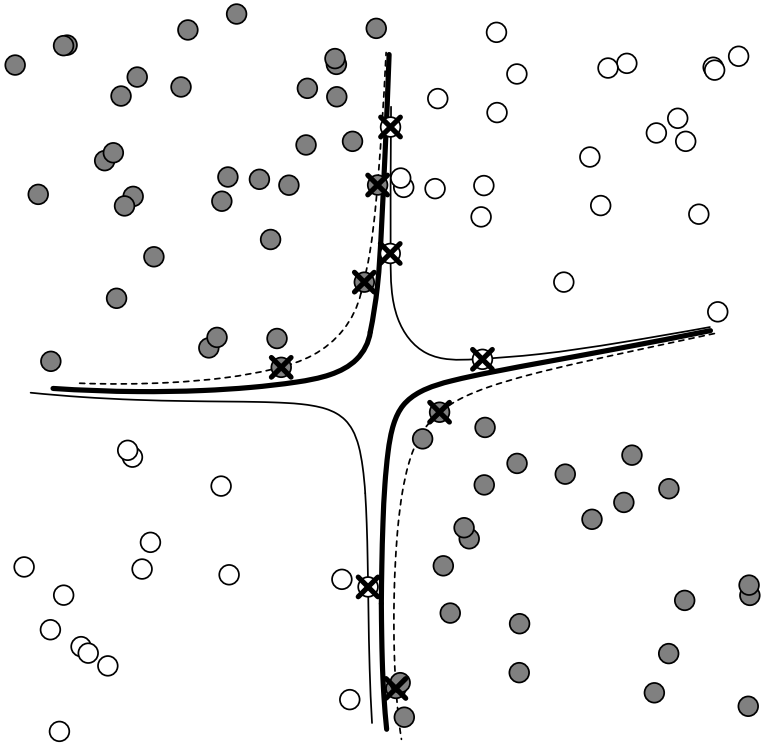

FIGURE 4.3 – Résolution du problème d'optimisation de la section 3.1.2 sur le corpus de la figure 4.1, mais avec un noyaux polynomial de degré 3. La frontière de séparation $f(\vec{x}) = 0$ est la ligne épaisse, les courbes $f(\vec{x}) = 1$ et $f(\vec{x}) = -1$ sont représentées également. Les *vecteurs supports* sont marqués d'une croix.

Le deuxième noyau que nous présenterons ici est le noyau gaussien, dit RBF [5], que nous avons déjà vu.

$$\mathbf{k}_{\text{rbf}}(\vec{x}, \vec{z}) = \exp\left(-\frac{\|\vec{x} - \vec{z}\|^2}{2\sigma}\right)$$

Ce noyau correspond à une projection dans un espace de dimension infinie. Toutefois, dans cet espace *tous les points sont projetés sur l'hypersphère de rayon 1.* En effet, $\|\phi(\vec{x})\|^2 = \mathbf{k}(\vec{x}, \vec{x}) = \exp(0) = 1$.

5. Radial Basis Function, ce terme vient des réseaux de neurones RBF que les SVM utilisant ce noyau généralisent.

4.2.4 Contruction de noyaux

Disposant de noyaux, il y a des résultats permettant d'en élaborer d'autres. Les principaux sont les suivants.
Soient :

- $\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{k}_3$ des fonctions noyau.
- f une fonction à valeurs dans $\mathbb{R}$.
- Φ une fonction qui projette les vecteurs dans un autre espace vectoriel.
- B une matrice semi-définie positive.
- p un polynome à coefficients positifs.
- α un réel positif.

alors les fonctions $\mathbf{k}$ suivantes sont des noyaux :

$$\mathbf{k}(\vec{x},\vec{z}) = \mathbf{k}_1(\vec{x},\vec{z}) + \mathbf{k}_2(\vec{x},\vec{z})$$
$$\mathbf{k}(\vec{x},\vec{z}) = \alpha\mathbf{k}_1(\vec{x},\vec{z})$$
$$\mathbf{k}(\vec{x},\vec{z}) = \mathbf{k}_1(\vec{x},\vec{z})\,\mathbf{k}_2(\vec{x},\vec{z})$$
$$\mathbf{k}(\vec{x},\vec{z}) = f(\vec{x})\,f(\vec{z})$$
$$\mathbf{k}(\vec{x},\vec{z}) = \mathbf{k}_3(\Phi(\vec{x}),\Phi(\vec{z}))$$
$$\mathbf{k}(\vec{x},\vec{z}) = \vec{x}^{\mathrm{T}}B\vec{z}$$
$$\mathbf{k}(\vec{x},\vec{z}) = p(\mathbf{k}_1(\vec{x},\vec{z}))$$
$$\mathbf{k}(\vec{x},\vec{z}) = \exp(\mathbf{k}_1(\vec{x},\vec{z}))$$

4.3 L'idée fondatrice des SVM

Ayant maintenant la ruse des noyaux à notre disposition, nous pouvons travailler, sans s'y projeter, dans des espaces de très grande dimension. Or une séparation linéaire, ainsi qu'une régression linéaire, est facilitée par la projection des données dans un espace de haute dimension... La contre partie étant que la séparation qu'on y fait, facilement, ne signifie rien. Dit autrement, on a vite fait d'apprendre par cœur, c'est-à-dire d'apprendre une fonction qui ne se généralisera pas à de nouveaux exemples. Cette contre-partie est ce qu'on appelle la *malédiction de la dimentionalité*[6]. En maximisant la marge, les SVM s'en sortent malgré tout, et ne se font pas piéger par cette malédiction. En projetant dans l'espace des caractéristiques pour employer un algorithme de maximisation de la marge, on arrive à obtenir une séparabilité en conservant de bonnes capacités de généralisation, et c'est l'idée centrale des SVM.

4.4 Quelques « trucs » relatifs aux noyaux

Pouvoir calculer un produit scalaire entre des vecteurs est suffisant pour réaliser plus d'opérations qu'il n'y paraît, opérations dont on profite sans projeter explicitement les vecteurs dans l'espace des caractéristiques. En voici quelques exemples.

4.4.1 Normalisation des données

La norme d'un vecteur caractéristique est donnée par :

$$\|\Phi(\vec{x})\| = \sqrt{\langle\Phi(\vec{x}).\Phi(\vec{x})\rangle} = \sqrt{\mathbf{k}(\vec{x},\vec{x})}$$

6. *Curse of dimentionality* en anglais.

Il en résulte que l'on peut très facilement travailler sur des données normalisées... dans l'espace des caractéristiques ! En effet, leur produit scalaire vaut :

$$\left\langle \frac{\Phi(\vec{x})}{\|\Phi(\vec{x})\|}.\frac{\Phi(\vec{z})}{\|\Phi(\vec{z})\|} \right\rangle = \frac{\langle \Phi(\vec{x}).\Phi(\vec{z}) \rangle}{\|\Phi(\vec{x})\| \, \|\Phi(\vec{z})\|} = \frac{\mathbf{k}\left(\vec{x},\vec{z}\right)}{\sqrt{\mathbf{k}\left(\vec{x},\vec{x}\right)\mathbf{k}\left(\vec{z},\vec{z}\right)}}$$

Donc il suffit d'utiliser le membre droit de l'expression ci-dessus comme nouveau noyau, construit à partir d'un noyau $\mathbf{k}$, pour travailler sur des vecteurs normalisés de l'espace des caractéristiques correspondant à $\mathbf{k}$. Si on note $\bar{\mathbf{k}}$ le noyau normalisé, on a tout simplement :

$$\bar{\mathbf{k}}(\vec{x},\vec{z}) = \frac{\mathbf{k}\left(\vec{x},\vec{z}\right)}{\sqrt{\mathbf{k}\left(\vec{x},\vec{x}\right)\mathbf{k}\left(\vec{z},\vec{z}\right)}}$$

On peut de même très facilement calculer la distance, dans l'espace des caractéristiques, des projections de deux vecteurs.

$$\|\Phi(\vec{x}) - \Phi(\vec{z})\| = \sqrt{\mathbf{k}\left(\vec{x},\vec{x}\right) - 2\mathbf{k}\left(\vec{x},\vec{z}\right) + \mathbf{k}\left(\vec{z},\vec{z}\right)}$$

4.4.2 Centrer et réduire

Pour certains algorithmes [7], il est préférable de centrer (soustraire leur centre de gravité) et de réduire les donnés (diviser par la variance). C'est quelque chose que l'on peut faire également dans l'espace des caractéristiques. Rappelons dans ce qui suit que l est, dans nos notations, le nombre d'exemples de la base.

Commençons par calculer la variance des données, qui nous permettra de les réduire si on le souhaite.

$$\text{var} = \frac{1}{l}\sum_{i=1}^{l}\left\|\Phi(\vec{x}_i) - \frac{1}{l}\sum_{j=1}^{l}\Phi(\vec{x}_j)\right\|^2 = \cdots = \frac{1}{l}\sum_{i=1}^{l}\mathbf{k}\left(\vec{x}_i,\vec{x}_i\right) - \frac{1}{l^2}\sum_{i,j=1}^{l}\mathbf{k}\left(\vec{x}_i,\vec{x}_j\right)$$

Pour travailler sur des données centrées reduites, on utilise alors le noyau $\hat{\mathbf{k}}$ définit comme suit :

$$\begin{aligned}\hat{\mathbf{k}}(\vec{x},\vec{z}) &= \left\langle \frac{\Phi(\vec{x}) - \frac{1}{l}\sum_{j=1}^{l}\Phi(\vec{x}_j)}{\sqrt{\text{var}}}.\frac{\Phi(\vec{z}) - \frac{1}{l}\sum_{j=1}^{l}\Phi(\vec{x}_j)}{\sqrt{\text{var}}} \right\rangle \\ &= \frac{1}{\text{var}}\left(\mathbf{k}\left(\vec{x},\vec{z}\right) - \frac{1}{l}\sum_{i=1}^{l}\mathbf{k}\left(\vec{x},\vec{x}_i\right) - \frac{1}{l}\sum_{i=1}^{l}\mathbf{k}\left(\vec{z},\vec{x}_i\right) + \frac{1}{l^2}\sum_{i,j=1}^{l}\mathbf{k}\left(\vec{x}_i,\vec{x}_j\right)\right)\end{aligned}$$

Attention toutefois, ces noyaux sont très gourmands en calcul, ce qui est déjà le cas pour les SVM avec des noyaux simples. C'est le genre de situation où il vaut mieux, à l'avance, stocker une bonne fois les valeurs du noyau pour les tous les couples de vecteurs de la base.

7. Il n'y a pas que les SVM qui profitent de la ruse des noyaux.

4.5 Noyaux pour l'analyse de documents et données structurées

En fait, dès qu'on a des données vectorielles, leur produit scalaire est une fonction noyau, et une façon d'utiliser les SVM est de rendre « vectorielles » les données. Dans les cas que nous allons considérer, ces vecteurs sont de grande dimension... il n'est donc pas forcément opportun, mais néanmoins possible, de les projeter dans un espace de caractéristiques. En revanche, la capacité des SVM à garder un pouvoir de généralisation dans les espaces de forte dimension, où les données sont « clairsemées » est essentielle ici.

L'ensemble des méthodes esquissées ici sont extraites de (Shawe-Taylor and Cristanini, 2004), où bien d'autres méthodes sont décrites.

4.5.1 Analyse de documents

Une façon de traiter les documents par des méthodes statistiques est de les considérer comme des « sacs de mots » [8]. On se donne pour cela un dictionnaire $m_1, \cdots, m_N$ de N mots, ainsi que la fonction Φ suivante associant un vecteur $\Phi(d)$ à un document d :

$$\begin{array}{rccl} \Phi & : & \text{Documents} & \to & \mathbb{N}^N \\ & & d & \mapsto & \Phi(d) = (f(m_1, d), \cdots, f(m_N, d)) \end{array}$$

Où $f(m, d)$ est le nombre de fois que le mot m apparaît dans le document d. Pour un ensemble $\{d_l\}_l$ de l documents, la matrice D [9] dont la ligne i est formé par le vecteur $\Phi(d_i)$ permet de définir un produit scalaire (donc un noyau) sur les documents. En effet, $\mathbf{k}(d_i, d_j)$ est donné par le coefficient (i, j) de DD^{T}. Les documents n'ayant pas la même taille on peut utiliser un noyau normalisé pour ne pas y être sensible.

De plus, on peut moduler ce noyau en injectant des connaissance sur la sémantique. Par exemple, on peut définir une matrice diagonale R dont chaque valeur sur la diagonale correspond à l'importance d'un mot. On peut définir aussi une matrice P de proximité sémantique, dont le coefficient $p_{i,j}$ traduit la proximité sémantique des mots m_i et m_j. La matrice sémantique $S = RP$ permet de créer un noyau tenant compte de ces connaissances :

$$\mathbf{k}(d_i, d_j) = \Phi(d_i) S S^{\mathrm{T}} \Phi(d_j)$$

4.5.2 Chaînes

Les chaînes de caractères sont très étudiées en informatique, et de nombreuses méthodes permettent de quantifier la similitude entre deux chaînes. D'ailleurs, la bio-informatique [10] est un des domaines d'application des SVM, et tire parti dans ce contexte de ces fonctions de similitude. Nous ne donnons ici qu'un exemple.

Considérons le cas du *p-spectrum kernel*. Il s'agit de comparer deux chaînes, qui peuvent être de longueur variable, en fonction des sous-chaînes [11] communes de longueur p qui les constituent. Soit un alphabet Σ, on note Σ^p l'ensemble des chaînes de longueur p, ainsi que $s_1 s_2$ la concaténation de s_1 et s_2, et $|A|$ le cardinal d'un ensemble A. On définit alors l'expression suivante, pour $u \in \Sigma^p$:

$$\Phi_u^p(s) = |\{(v_1, v_2) \ : s = v_1 u v_2\}|$$

8. *Bag of words* en anglais.
9. *Document term matrix* en anglais
10. Application des techniques d'apprentissage au traitement de séquences d'ADN.
11. Cha ne vaut pas l'hydromel...

Pour une chaîne s, on a donc un $\Phi_u^p(s)$ par sous-chaîne u possible. $\Phi_u^p(s)$ vaut zéro pour la plupart des $u \in \Sigma^p$. On projette donc une chaîne s sur un espace vectoriel à $|\Sigma|^p$ dimensions, le vecteur $\Phi^p(s)$ ayant les $\Phi_u^p(s)$ pour composantes. On peut alors définir un noyau par un simple produit scalaire :

$$\mathbf{k}(s,t) = \langle \Phi^p(s).\Phi^p(t) \rangle = \sum_{u \in \Sigma^p} \Phi_u^p(s)\Phi_u^p(t)$$

Illustrons ce propos avec $p = 3$ et les chaînes `bateau`, `rateau`, `oiseau`, `croise ciseaux`. Les éléments de Σ^3 qui conduiront à des composantes non nulles sont `ate`, `aux`, `bat`, `cis`, `cro`, `eau`, `ise`, `ois`, `rat`, `roi`, `sea`, `tea`. Les lignes du tableau 4.1 sont les composantes non nulles des $\Phi^p(s)$.

	ate	aux	bat	cis	cro	eau	ise	ois	rat	roi	sea	tea
bateau	1		1			1						1
rateau	1					1			1			1
oiseau						1	1	1			1	
croise					1		1	1		1		
ciseaux		1		1		1	1				1	

TABLE 4.1 – *3-spectrum* des mots `bateau`, `rateau`, `oiseau`, `croise ciseaux`.

On peut alors représenter comme sur le tableau 4.2, sous forme d'une matrice, les valeurs du noyau pour chaque couple de mots.

k	bateau	rateau	oiseau	croise	ciseaux
bateau	4	3	1	0	1
rateau	3	4	1	0	1
oiseau	1	1	4	2	3
croise	0	0	2	4	1
ciseaux	1	1	3	1	5

TABLE 4.2 – *3-spectrum* des mots `bateau`, `rateau`, `oiseau`, `croise ciseaux`.

4.5.3 Autres exemples

On peut aussi construire des noyaux dont le calcul est récursif, qui permettent de « faire le produit scalaire » de deux objets structurés [12], comme des graphes ou des arbres. Toute la subtilité et la difficulté pour les concepteurs de ces noyaux réside dans le fait qu'il faut produire un nombre à partir de deux objets structurés, et que la fonction qui produit ce nombre ait bien la propriété de Mercer.

12. Objets symboliques.

Chapitre 5

Résolution

5.1 Problème d'optimisation quadratique via SMO

5.1.1 Principe général

Il y a plusieurs méthodes de résolution du problème d'optimisation exprimé au paragraphe 3.1.2, l'une d'elle étant l'algorithme SMO [1]. Même pour cette méthode, on trouve plusieurs raffinements dans la littérature.

L'idée de base est de partir des KKT conditions du problème dual énoncé page 18. Il s'agit de se promener dans l'espace des α_l pour faire croître la fonction objectif. SMO consiste à s'y promener en bougeant les α_l deux par deux. En effet, l'équation L2 présentée en 3.2.5 montre que si l'on fixe tous les α_l sauf deux, ceux-ci sont liés linéairement : on peut exprimer un des deux α_l restant en fonction de l'autre, ce qui permet de réécrire la fonction objectif en fonction d'un seul α_l variable uniquement, et d'en prendre le gradient.

L'arrêt de l'algorithme se fait quand certaines conditions, qui caractétrisent l'optimalité, sont remplies.

En pratique, cette méthode est truffée de ruses algorithmiques pour bien choisir les couples de α_l afin d'accélérer la convergence, et est relativement difficile à implémenter. Nous présentons ici la méthode proposée dans (Keerthi et al., 1999), qui est une amélioration de l'algorithme SMO proposé initialement proposé dans (Platt, 1998).

5.1.2 Détection de l'optimalité

L'idée est de repartir du problème dual énoncé page 18. Il s'agit d'un problème de minimisation (on multiplie par -1 la fonction objectif pour cela), sous 3 contraintes (il y a deux inégalités dans la deuxième). Bien que ce problème soit déjà issu d'une résolution lagrangienne, on peut à nouveau définir sa solution à l'aide d'un lagrangien... mais on ne tourne pas en rond, car ce lagrangien-là ne nous servira qu'à exprimer des conditions d'optimalité qui seront les conditions d'arrêt d'un algorithme qui lui, résout directement le problème dual énoncé page 18. Ce lagrangien est donc [2] :

$$L(\alpha, \delta, \mu, \beta) = \frac{1}{2} \sum_k \sum_l \alpha_k \alpha_l y_k y_l \langle \vec{x}_k . \vec{x}_l \rangle - \sum_l \alpha_l - \sum_l \delta_l \alpha_l + \sum_l \mu_l (\alpha_l - C) - \beta \sum_l \alpha_l y_l$$

1. Sequential Minimal Optimization

2. Attention, les α_l sont maintenant les paramètres primaux, ce sont les δ_l, μ_l et le paramètre β qui sont les multiplicateurs de Lagrange.

Pour simplifier les notations dans la suite, on définit :

$$F_i = \sum_l \alpha_l y_l \langle \vec{x}_i . \vec{x}_l \rangle - y_i$$

On peut alors exprimer par ce qui suit les conditions KKT, c'est-à-dire l'annulation des dérivées partielles du lagrangien, ainsi que les conditions supplémentaires de KKT, c'est-à-dire le fait que lorsque'un multiplicateur est nul, c'est que la contrainte n'est pas saturée, et que lorsqu'il est non nul, c'est qu'elle est saturée. Le produit de la contrainte et de son multiplcateur est donc nul à l'optimum, sans que les deux facteurs soient nuls en même temps (voir page 17 pour un rappel). Les multiplicateurs sont également tous positifs.

$$\begin{aligned} \forall l \\ \frac{\partial L}{\partial \alpha_l} = (F_l - \beta) y_l - \delta_l + \mu_l &= 0 \\ \delta_l \alpha_l &= 0 \\ \mu_l (\alpha_l - C) &= 0 \end{aligned}$$

Ces conditions se simplifient si on les écrit de la façon suivante, en distinguant 3 cas selon la valeur de α_l.

Cas $\alpha_l = 0$ **:** On a donc $\delta_l > 0$ et $\mu_l = 0$, donc

$$(F_l - \beta) y_l \geq 0$$

Cas $0 < \alpha_l < C$ **:** On a donc $\delta_l = 0$ et $\mu_l = 0$, donc

$$(F_l - \beta) y_l = 0$$

Cas $\alpha_l = C$ **:** On a donc $\delta_l = 0$ et $\mu_l > 0$, donc

$$(F_l - \beta) y_l \leq 0$$

Comme $y_l \in \{-1, 1\}$, on peut séparer les l selon le signe de $F_l - \beta$. On définit ainsi les ensembles d'indices suivants :

$$\begin{aligned} I_0 &= \{l : 0 < \alpha_l < C\} \\ I_1 &= \{l : y_l = 1, \alpha_l = 0\} \\ I_2 &= \{l : y_l = -1, \alpha_l = C\} \\ I_3 &= \{l : y_l = 1, \alpha_l = C\} \\ I_4 &= \{l : y_l = -1, \alpha_l = 0\} \\ I_{\text{sup}} &= I_0 \cup I_1 \cup I_2 \\ I_{\text{inf}} &= I_0 \cup I_3 \cup I_4 \end{aligned}$$

On a alors :

$$\begin{aligned} l \in I_{\text{sup}} \Rightarrow \beta \leq F_l \\ l \in I_{\text{inf}} \Rightarrow \beta \geq F_l \end{aligned}$$

On peut alors définir les bornes suivantes de ces ensembles :

$$\begin{aligned} b_{\text{sup}} = \min_{l \in I_{\text{sup}}} F_l \\ b_{\text{inf}} = \max_{l \in I_{\text{inf}}} F_l \end{aligned}$$

En fait, quand on itèrera l'algorithme, on aura $b_{\text{sup}} \leq b_{\text{inf}}$ tant qu'on n'atteindra pas l'optimum, mais à l'optimum :

$$b_{\text{inf}} \leq b_{\text{sup}}$$

On peut prendre le critère à l'envers, et dire qu'il n'y a pas optimalité si on trouve deux indices, un dans I_{sup} et l'autre dans I_{inf}, qui contredisent la relation que $b_{\text{inf}} \leq b_{\text{sup}}$. Une telle paire d'indice définit une *violation* des conditions d'optimalités :

$$(i,j) \text{ tq } i \in I_{\text{sup}} \text{ et } j \in I_{\text{inf}} \text{ est une violation si } F_i < F_j \tag{5.1}$$

L'équation 5.1 est théorique, car en effet, on n'aura jamais, numériquement $b_{\text{inf}} \leq b_{\text{sup}}$ à l'optimum. On se contentera donc de définir cette condition à un poil près, disons $\tau > 0$. Dit autrement, l'optimalité approximative est :

$$b_{\text{inf}} \leq b_{\text{sup}} + \tau \tag{5.2}$$

L'équation 5.1 qui définit qu'une paire d'indice viole les conditions d'optimalité est alors modifiée en conséquence :

$$(i,j) \text{ tq } i \in I_{\text{sup}} \text{ et } j \in I_{\text{inf}} \text{ est une violation si } F_i < F_j - \tau \tag{5.3}$$

Le critère 5.3 sera testé pour savoir si on continue l'algorithme d'optimisation, ou si on peut considérer que l'optimum est atteint.

Avant de conclure sur ce paragraphe, destiné finalement à présenter la condition d'arrêt de l'algorithme décrit ci-après, signalons qu'à l'optimum, $b_{\text{sup}} \approx b_{\text{inf}} \approx \beta$... et que cette valeur est aussi le b de notre séparateur ! Une formule pour b n'était pas disponible par la résolution lagrangienne, comme nous l'avions déploré en 3.2.5 page 18.

5.1.3 Algorithme d'optimisation

Le principe d'optimisation SMO consiste à partir du constat que l'équation L2 (cf. 3.2.5) lie les α_l, leur somme, pondérée par les y_l, étant constante. Donc si l'on autorise deux de ces coefficients α_{l_1} et α_{l_2} seulement à bouger (dans $[0,C]^2$), fixant les autres, on a $L2 \Rightarrow \alpha_{l_1} y_{l_1} + \alpha_{l_2} y_{l_2} + C = 0$. Ayant $\alpha_{l_1} = (-C - \alpha_{l_2} y_{l_2})/y_{l_1}$, on peut écrire la fonction objectif du problème dual (cf. page 18) comme une fonction de α_{l_1} seulement. On peut calculer le point $\alpha^\star_{l_1}$ pour lequel cette fonction est maximale [3]. On met à jour $\alpha_{l_1} \leftarrow \alpha^\star_{l_1}$ et on en déduit α_{l_2}. On prend toutefois la précaution de confiner ces variations pour que $(\alpha_{l_1}, \alpha_{l_2})$ reste dans $[0,C]^2$. Ensuite, on choisit une autre paire de coefficients à mettre à jour, etc.

Tout le problème est de choisir les paires (l_1, l_2). C'est un problème épineux, dont dépend la rapidité avec laquelle on atteint le maximum. La solution proposée dans (Keerthi et al., 1999) consiste à parcourrir tous les l_1, à déterminer si $l_1 \in I_{\text{sup}}$ ou si $l_1 \in I_{\text{inf}}$. Pour le l_1 considéré, l_2 est l'indice appartenant à l'autre ensemble, pour lequel la borne b_{sup} ou b_{inf}, selon l'ensemble considéré, est atteinte. On travaille ainsi sur la paire qui viole le plus les conditions d'optimalité. Il y a encore quelques raffinements, détaillés dans le pseudo-code de (Keerthi et al., 1999).

Numériquement, la mise à jour d'une paire peut conduire à une modification négligeable des deux $(\alpha_{l_1}, \alpha_{l_2})$, disons plus petite en valeur absolue qu'un seuil θ. Si toutes les modifications occasionnées par une passe sur les exemples sont négligeables, on arrête l'algorithme (cas d'arrêt #1). De plus, si l'on constate via l'équation 5.2 que l'optimalité est atteinte, on arrête également l'optimisation (cas d'arrêt #2).

3. C'est une fonction quadratique dont il suffit d'annuler la dérivée.

5.1.4 Problèmes numériques

Le cas d'arrêt #1 dépend de la valeur de θ, alors que le cas #2 dépend de τ. Or on peut très bien s'arrêter à cause du cas #1, sans que l'optimalité soit atteinte... Cela arrive par exemple quand l'optimisation se joue sur le 15ième chiffre après la virgule.

Il faut donc bien choisir les paramêtres τ et θ. Une bonne chose à vérifier quand l'algorithme s'arrête, c'est le taux de paires de coefficients qui violent les critères d'optimalité. Si ce taux est fort, il faut réduire θ. Si en revanche la solution vous paraît manifestement mauvaise, alors que plus aucune paire ne viole les critères d'optimalité, c'est que τ est trop élevé.

5.2 Choix des paramètres

Le choix des paramètres qui correspondent le mieux au problème n'est pas intuitif. En effet, dans le cas de la classification que nous avons vu précédemment, le paramètre C est à déterminer. De même, si par exemple on utilise un noyau gaussien, le paramètre σ est aussi à ajuster.

Pour ce faire, une méthode brutale consiste à essayer plusieurs valeurs de paramètres [4], et à mesurer l'erreur de généralisation de la SVM par une validation croisée. Ainsi, on prendra les paramètres pour lesquels l'erreur de généralisation est la plus faible.

4. *Grid search method* en anglais.

Chapitre 6

Régression

Nous avons jusqu'ici étudié le problème de la séparation d'un corpus d'exemple en deux classes, suivant leurs étiquettes $+1$ ou -1. La régression consiste à considérer des étiquettes ayant n'importe quelle valeur réelle, et à essayer d'inférer la fonction qui au vecteur associe son étiquette, à partir des exemples du corpus.

6.1 Position du problème d'optimisation

Là encore, nous allons étudier le cas d'une régression linéaire, qui se généralisera aux autres régressions par l'utilisation de noyaux à la place des produits scalaires.

La régression présentée ici part du principe qu'un séparateur linéaire $f_{\vec{w},b}$ est bon si les exemples sont bien représentés par ce séparateur, à une erreur $\epsilon > 0$ près, c'est-à-dire :

$$\forall l,\ |(\langle \vec{w}.\vec{x}_l \rangle + b) - y_l| \leq \epsilon$$

Bien sûr, cette contrainte est trop forte, et en pratique on exprimera une fonction à optimiser qui autorise certains exemples à déroger à cette contrainte. La figure 6.1 illustre ce propos.

Par une démarche analogue à celle du séparateur, on en vient à poser le problème d'optimisation suivant pour la régression :

Jouer sur $\vec{w}$, b, ξ et ξ' pour minimiser $\frac{1}{2}\langle \vec{w}.\vec{w} \rangle + C \sum_l (\xi_l + \xi'_l)$

En respectant
$$\begin{cases} y_l - \langle \vec{w}.\vec{x}_l \rangle - b \leq \epsilon + \xi_l,\ \forall (\vec{x}_l, y_l) \in S \\ \langle \vec{w}.\vec{x}_l \rangle + b - y_l \leq \epsilon + \xi'_l,\ \forall (\vec{x}_l, y_l) \in S \\ \xi_l, \xi'_l \geq 0,\ \forall l \end{cases}$$

6.2 Résolution

La résolution de ce problème d'optimisation est plus aisée en passant au problème dual, comme pour la séparation vue au chapitre 3. Soient α_l et α'_l les multiplieurs relatifs aux deux premières contraintes du problème d'optimisation précédent, le vecteur $\vec{w}$ du séparateur est donné par :

$$\vec{w}_{\alpha,\alpha'} = \sum_l (\alpha_l - \alpha'_l)\vec{x}_l$$

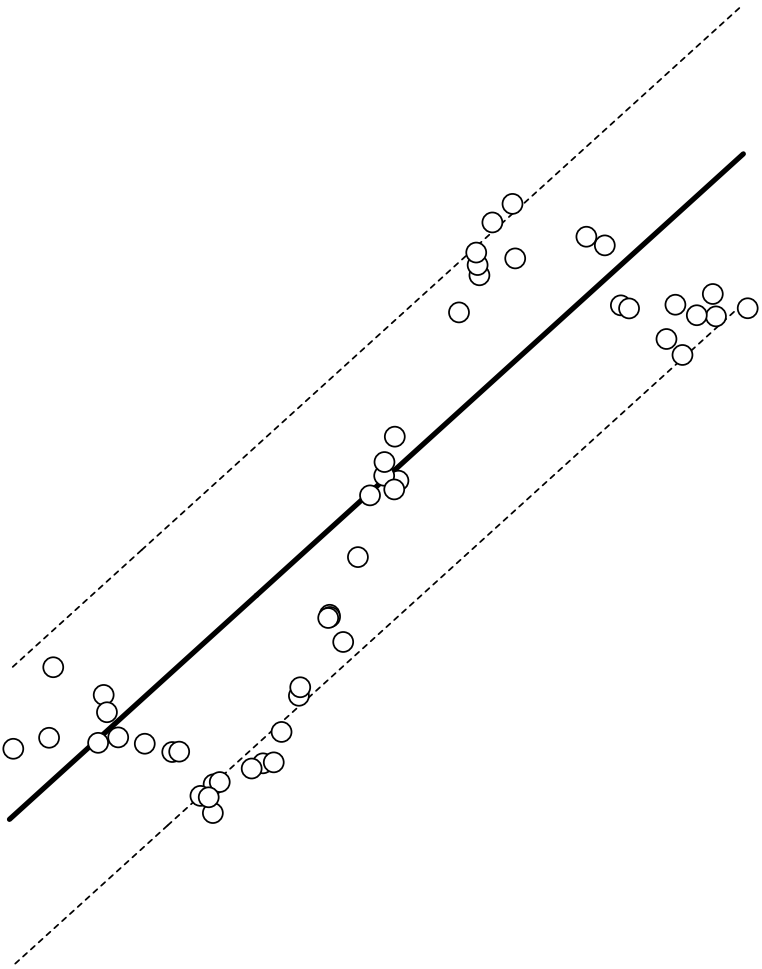

FIGURE 6.1 – Régression linéaire. Les points blancs ont pour abscisse x_l, vecteur de dimension 1, et pour ordonnée y_l. La bande en pointillés correspond à la zone acceptable de distance au séparateur, $|w.x + b - y| \leq \epsilon$, et peu d'exemples sortent de cette zone.

avec les α_l et α'_l solution du problème d'optimisation dual suivant :

$$\text{Jouer sur } \vec{\alpha} \text{ et } \vec{\alpha}' \text{ pour maximiser } \sum_l y_l(\alpha_l - \alpha'_l) - \epsilon \sum_l (\alpha_l + \alpha'_l) - \frac{1}{2} \langle \vec{w}_{\alpha,\alpha'} . \vec{w}_{\alpha,\alpha'} \rangle$$

$$\text{En respectant } \begin{cases} \sum_l (\alpha_l - \alpha'_l) = 0 \\ \alpha_l, \alpha'_l \in [0, C], \ \forall l \end{cases}$$

Là aussi, il reste à appliquer un algorithme de recherche du maximum de ce problème dual, et des méthodes type SMO existent, et aboutissent à des algorithmes assez difficiles à implémenter.

6.3 Exemples

La figure 6.2 montre l'utilisation de ce type de SVM pour la régression dans le cas de vecteurs 1D, avec différents noyaux, et la figure 6.3 montre la régression dans le cas de vecteurs 2D.

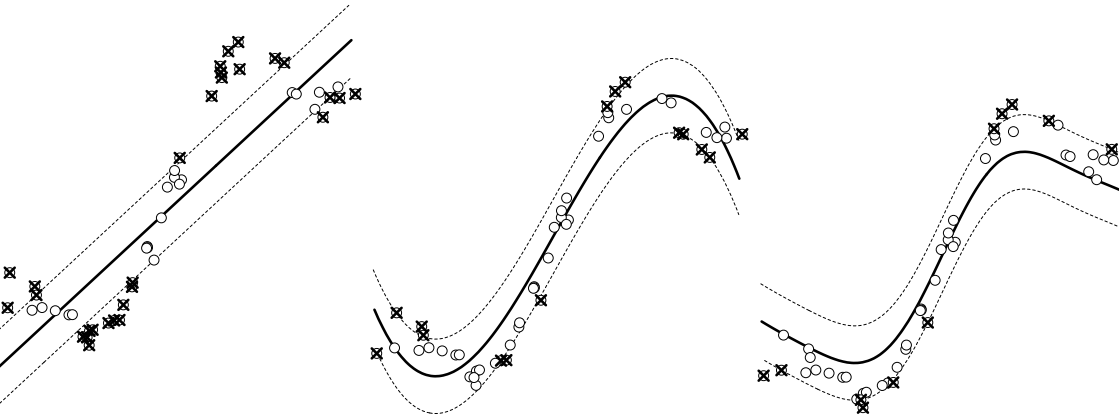

FIGURE 6.2 – Régression sur des vecteurs 1D, comme pour la figure 6.1. À gauche, utilisation du produit scalaire normal, au milieu d'un noyau polynomial de degré 3, et à droite, d'un noyau gaussien. Les vecteurs supports sont marqués d'une croix. Il s'agit des vecteurs x_l pour lesquels un des deux α_l, α'_l est non nul. Ce sont eux qui conditionnent la position de la courbe de régression. La tolérance $+\epsilon, -\epsilon$ est représentée en pointillés.

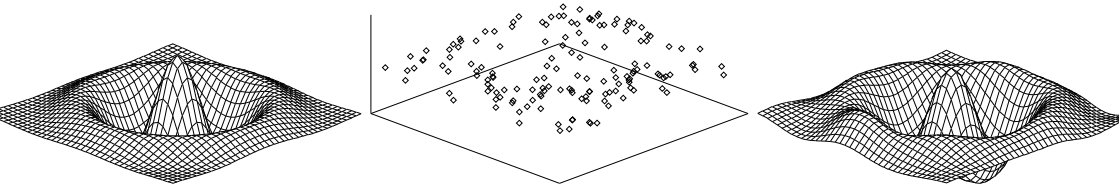

FIGURE 6.3 – À gauche, la fonction $z = f(x, y) = exp(-2.5(x^2 + y^2) * cos(8 * \sqrt{x^2 + y^2})$ que l'on utilise pour générer les exemples. Au milieu, on montre le tirage de 150 exemples, obtenus en tirant aléatoirement x et y, et en attribuant au vecteur $\vec{x}_l = (x, y)$ la valeur $f(x, y) + \nu$, avec ν une valeur aléatoire dans $[-.1, .1]$. À droite, on montre les valeurs en tous points (x, y) de la fonction résultat de la régression sur les exemples, avec un noyau gaussien de variance $\sigma = .25$ et une tolérence $\epsilon = .05$.

Chapitre 7

Florilège de SVM

Finalement, le principe des méthodes que nous avons vues jusqu'ici est toujours le même. Il s'agit de poser un problème d'optimisation quadratique, dont la résolution n'implique que des produits scalaires deux à deux.

7.1 Classification

Ces méthodes sont appelées SVC [1].

7.1.1 C-SVC

Cette méthode est celle que nous avons vue, le problème d'optimisation est simplement rappelé ici :

$$\text{Jouer sur } \vec{w},\ b \text{ et } \xi \text{ pour minimiser} \quad \frac{1}{2}\langle \vec{w}.\vec{w}\rangle + C\sum_l \xi_l$$

$$\text{En respectant} \quad \begin{cases} y_l\left(\langle \vec{w}.\vec{x}_l\rangle + b\right) \geq 1 - \xi_l,\ \forall (\vec{x}_l, y_l) \in S \\ \xi_l \geq 0,\ \forall l \end{cases}$$

7.1.2 ν-SVC

Le problème avec une C-SVM, c'est que C, qui préside au recours aux *slack variables* ξ_l, est indépendant du nombre d'exemples. On peut en effet vouloir contrôler le nombre de supports, relativement au nombre d'exemples, et non dans l'absolu. Le paramètre $\nu \in]0, 1]$ est lié au pourcentage d'exemples que l'on autorise à servir de supports [2]. Dans une C-SVM, on force les exemples à se situer en dehors de la bande $[-1, 1]$. Ici, on choisit une bande $[-\rho, \rho]$, que l'on ajuste pour obtenir le taux de vecteurs supports souhaités.

1. *Support Vector Classification.*
2. Ce pourcentage tend vers ν si on a beaucoup d'exemples.

$$\text{Jouer sur } \vec{w},\ b,\ \xi \text{ et } \rho \text{ pour minimiser } \quad \frac{1}{2}\langle \vec{w}.\vec{w}\rangle - \nu\rho + \frac{1}{l}\sum_l \xi_l$$

$$\text{En respectant } \quad \begin{cases} y_l\left(\langle \vec{w}.\vec{x}_l\rangle + b\right) \geq \rho - \xi_l,\ \forall (\vec{x}_l, y_l) \in S \\ \xi_l \geq 0,\ \forall l \\ \rho \geq 0 \end{cases}$$

La formulation de la fonction objectif ci-dessus n'est toutefois pas si simple, car le fait que ν soit le taux d'exemples de la base qui servent de support ne saute pas aux yeux. Ceci se justifie en analysant les conditions KKT de ce problème d'optimisation (Schölkopf et al., 2000).

7.2 Régression

Ces méthodes sont appelées SVR [3].

7.2.1 ϵ-SVR

Il s'agit de la méthode que l'on a vue. On rappelle simplement ici le problème d'optimisation.

$$\text{Jouer sur } \vec{w},\ b,\ \xi \text{ et } \xi' \text{ pour minimiser } \quad \frac{1}{2}\langle \vec{w}.\vec{w}\rangle + C\sum_l (\xi_l + \xi'_l)$$

$$\text{En respectant } \quad \begin{cases} \langle \vec{w}.\vec{x}_l\rangle + b - y_l \geq \epsilon - \xi_l,\ \forall (\vec{x}_l, y_l) \in S \\ \langle \vec{w}.\vec{x}_l\rangle + b - y_l \leq \epsilon + \xi'_l,\ \forall (\vec{x}_l, y_l) \in S \\ \xi_l, \xi'_l \geq 0,\ \forall l \end{cases}$$

7.2.2 ν-SVR

À l'instar des ν-SVC, il s'agit ici de moduler la largeur ϵ de la $\epsilon - SVR$, selon un paramètre $\nu \in]0, 1]$. L'idée est que le nombre d'exemple en dehors du tube de rayon ϵ autour de la fonction soit une fraction ν du nombre total d'exemples [4].

$$\text{Jouer sur } \vec{w},\ b,\ \xi,\ \xi' \text{ et } \epsilon \text{ pour minimiser } \quad \frac{1}{2}\langle \vec{w}.\vec{w}\rangle + C(\nu\epsilon + \frac{1}{l}\sum_l (\xi_l + \xi'_l))$$

$$\text{En respectant } \quad \begin{cases} \langle \vec{w}.\vec{x}_l\rangle + b - y_l \geq \epsilon - \xi_l,\ \forall (\vec{x}_l, y_l) \in S \\ \langle \vec{w}.\vec{x}_l\rangle + b - y_l \leq \epsilon + \xi'_l,\ \forall (\vec{x}_l, y_l) \in S \\ \xi_l, \xi'_l \geq 0,\ \forall l \\ \epsilon \geq 0 \end{cases}$$

Comme pour le cas des ν-SVC, on trouvera dans (Schölkopf et al., 2000) une justification de cette formulation du problème d'optimisation, qui s'obtient d'après les conditions KKT que ce problème induit.

3. *Support Vector Regression.*

4. Du moins, en pratique, le pourcentage d'exemples en dehors du tube de rayon ϵ autour de la fonction tend vers ν quand le nombre d'exemples est élevé.

7.3 Apprentissage non supervisé

On peut faire de l'apprentissage non supervisé (analyse de distribution) avec les SVM. En fait, il s'agit de construire un détecteur de nouveauté. On demande à la SVM de décrire les exemples comme un agglomérat [5] de points. Une fois cet agglomérat identifié, tout exemple qui serait en dehors sera considéré comme ne relevant pas de la distribution des autres, il leur est différent, on a affaire à un nouveau phénomène. C'est pratique quand on n'a que des exemples et peu de contre-exemples. Dans l'analyse d'un signal EEG pour prédire une crise d'épilepsie par exemple, les données fournies décrivent principalement des états non annonciateurs de crises, car les crises sont bien moins fréquentes que l'état normal. Une façon de détecter la crise est de voir que le signal sort du *périmètre* des signaux normaux.

7.3.1 Plus petite sphère englobante

L'idée derrière ces techniques est d'englober les exemples $x_1, \cdots, x_l$ dans une sphère de rayon minimal. Bien sûr, si on prend cette contrainte stricte, le moindre exemple bruité qui est loin des autres empêchera la sphère de coller au reste des données. En fait, on pourrait vouloir trouver la sphère la plus petite qui contient $\alpha\%$ des données. Or ce problème est NP-complet...

On peut utiliser la ruse des *slack variables* qui donneront une solution approchée du problème NP-complet.

$$\text{Jouer sur } \vec{\omega}, r \text{ et } \xi \text{ pour minimiser} \quad r^2 + C\sum_l \xi_l$$
$$\text{En respectant} \quad \begin{cases} \|\vec{x}_l - \vec{\omega}\|^2 \leq r^2 + \xi_l, \ \forall \vec{x}_l \in S \\ \xi_l \geq 0, \ \forall l \end{cases}$$

On trouve après résolution lagrangienne une formule qui donne r et une fonction qui vaut 1 quand on sort de la sphère, ces deux résultats ne s'exprimant bien heureusement que par des produits scalaires (cf. figure 7.1).

On peut comme précédemment trouver une ν-version de ce problème, pour contrôler le pourcentage de vecteurs supports, c'est-à-dire le pourcentage d'exemples en dehors de la sphère (Shawe-Taylor and Cristanini, 2004). Le problème d'optimisation est le même que le précédent, en posant $C = 1/\nu l$, puis en multipliant par ν la quantité à minimiser [6]. La raison pour laquelle ce C-là conduit à ce que ν soit effectivement lié au pourcentage d'exemples hors de la sphère s'obtient là-aussi par une analyse des conditions KKT.

$$\text{Jouer sur } \vec{\omega}, r \text{ et } \xi \text{ pour minimiser} \quad \nu r^2 + \frac{1}{l}\sum_l \xi_l$$
$$\text{En respectant} \quad \begin{cases} \|\vec{x}_l - \vec{\omega}\|^2 \leq r^2 + \xi_l, \ \forall \vec{x}_l \in S \\ \xi_l \geq 0, \ \forall l \end{cases}$$

5. *Cluster* en anglais.
6. Ce qui ne change rien au résultat.

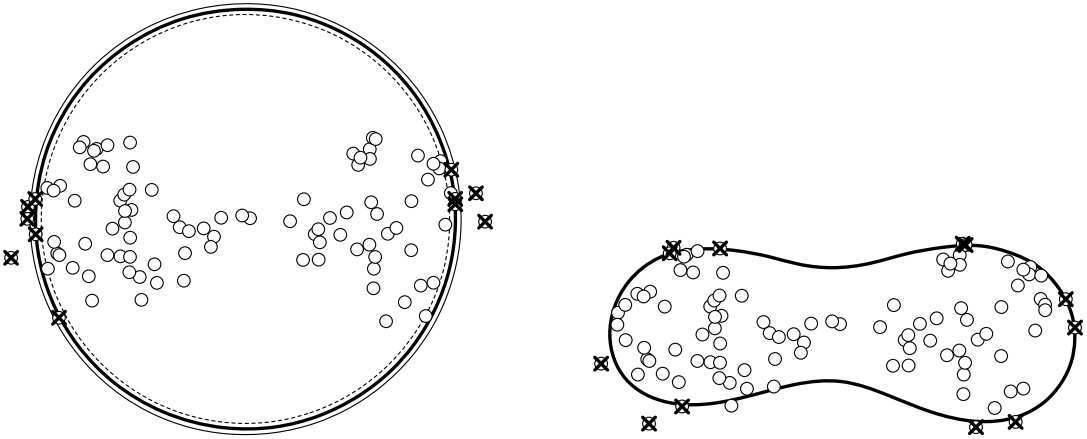

FIGURE 7.1 – Plus petite sphère englobante. À gauche on utilise le produit scalaire, à droite un noyau gaussien de paramètre $\sigma = 3$. Dans les deux cas, $C = 0.1$ et les exemples tirés sont inclus dans un carré de 10×10.

7.3.2 One-class SVM

Ce cas est un peu particulier, quoi que très utilisé. Il s'agit de trouver un hyperplan qui vérifie deux conditions. La première est que l'origine soit du côté négatif de l'hyperplan, et que les exemples soient tous du côté positif, aux *slack* variables près, comme d'habitude. La seconde condition est que cet hyperplan soit le plus éloigné possible de l'origine. De façon générale, l'intérêt ne saute pas aux yeux, car les exemples pourraient très bien se situer tout autour de l'origine. En fait, cette SVM est pertinente dans le cas de noyaux radiaux, comme les noyaux gaussiens. Ces derniers en effet projettent les vecteurs sur une hypersphère centrée à l'origine (cf. remarque du paragraphe 4.2.3), l'origine est donc « naturellement » loin des données pour ce type de noyaux, et les exemples sont agglutinés sur une portion de l'hypersphère. On isole cette portion en repoussant au maximum l'hyperplan (cf. figure 7.2).

$$\text{Jouer sur } \vec{w},\, b,\, \xi \text{ et } \rho \text{ pour minimiser} \quad \frac{1}{2}\langle \vec{w}.\vec{w}\rangle - \rho + \frac{1}{\nu l}\sum_l \xi_l$$
$$\text{En respectant} \quad \begin{cases} \langle \vec{w}.\vec{x}_l\rangle \geq \rho - \xi_l,\ \forall \vec{x}_l \in S \\ \xi_l \geq 0,\ \forall l \end{cases}$$

Le fait que, là-aussi, ν est lié à la fraction des exemples qui sont du côté négatif de l'hyperplan est justifié dans (Schölkopf et al., 2001) par des arguments sur les conditions KKT du problème.

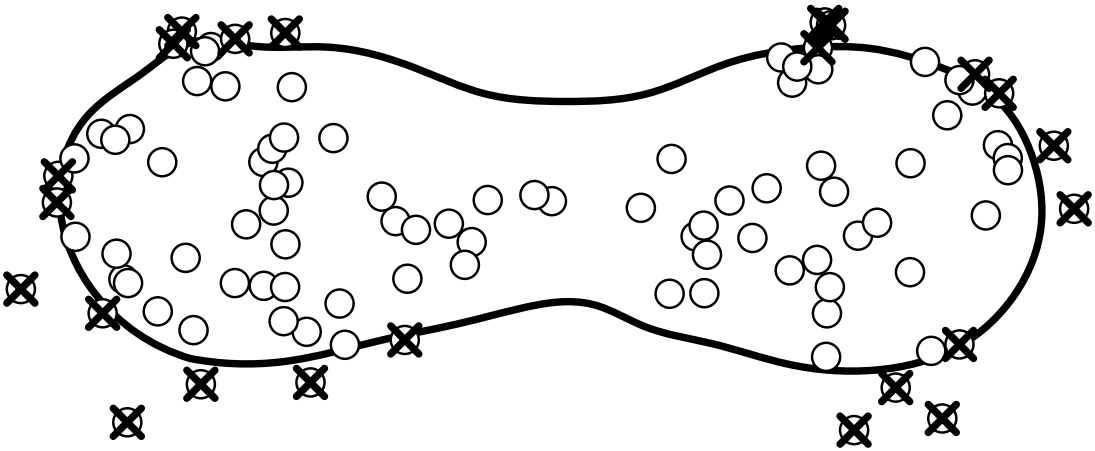

FIGURE 7.2 – One class SVM. Les exemples sont ceux de la figure 7.1. On utilise un noyau gaussien de paramètre $\sigma = 3$, et $\nu = 0.2$, ce qui signifie que 20% des exemples environ sont hors de la région.

Bibliographie

Cristanini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., and Murthy, K. R. K. (1999). Improvements to platt's smo algorithm for svm classifier design. Technical Report CD-99-14, National University of Singapore.

Lin, C.-J. (2010). libsvm. http://www.csie.ntu.edu.tw/~cjlin.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods : Support Vector Machines*. MIT Press, Cambridge, MA.

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13 :1443–1471.

Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, 12 :1207–1245.

Shawe-Taylor, J. and Cristanini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Shevade, S. K., Keerthi, S. S., Bhattacharyya, C., and Murthy, K. R. K. (1999). Improvements to smo algorithm for svm regression. Technical Report CD-99-16, National University of Singapore.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer.