CAMUS
(Cahiers Mathématiques de l'Université de Sherbrooke)

Titre : Machines à vecteurs de support : une introduction

Auteur(s) : Dominik Francoeur

Revue : CaMUS (Cahiers Mathématiques de l'Université de Sherbrooke)

Volume : 1

Année : 2010

Pages : 7-25

Éditeur : Université de Sherbrooke. Département de Mathématiques

URI : Repéré à : http://camus.math.usherbrooke.ca/revue.html

Page vide laissée intentionnellement

CaMUS **1** (2010), 7–25

Machines à vecteurs de support—Une introduction

Dominik Francoeur

Résumé. Les machines à vecteurs de support, ou SVM (Support Vector Machines), sont une méthode relativement récente de résolution de problèmes de classification (trier des individus en fonction de leurs caractéristiques) qui suscite beaucoup d'intérêt, à la fois pour son élégance et ses bonnes performances. Cet article explique leur fonctionnement, puis préente une importante variante, indispensable à l'application de la méthode en pratique.

La classification consiste à classer des individus en fonction de certaines de leurs caractéristiques. Il existe différents types de classification, mais un des plus intuitifs et des plus utilisés est la classification supervisée. L'idée de la classification supervisée est d'apprendre une règle de classement à partir d'un ensemble de données dont le classement est déjà connu. Une fois la règle apprise, il est possible de l'appliquer pour catégoriser de nouvelles données, dont le classement est inconnu. Les *machines à vecteurs de support*, ou *SVM* (*Support Vector Machines*), sont une technique relativement récente (elles ont été introduites en 1992 par Vladimir Vapnik, Bernhard Boser et Isabelle Guyon) de classification supervisée qui suscite beaucoup d'intérêt pour ses bonnes performances dans un large éventail d'applications pratiques.

Cet article constitue une introduction aux machines à vecteurs de support. Après la présentation de certains éléments essentiels de la théorie de l'optimisation, leur fonctionnement sera expliqué, d'abord dans le cas simple où les données sont linéairement séparables, puis dans le cas général où elles ne le sont pas nécessairement. Enfin, le concept de marge souple, très important pour l'utilisation des SVM dans la pratique, sera introduit.

J'aimerais remercier le CRSNG pour leur support financier, ainsi que Jessica Lévesque pour son support et ses nombreuses révisions de mon article.

1. Éléments de la théorie de l'optimisation

1.1. Introduction à l'optimisation

Soit f une fonction dérivable définie sur un domaine ouvert D dont on cherche à connaître le minimum. Supposons que ce minimum est atteint au point $\mathbf{x}^*$, c'est-à-dire que $f(\mathbf{x}^*) \leq f(\mathbf{x})$ pour tout $\mathbf{x} \in D$. Alors, nécessairement, $f'(\mathbf{x}^*) = 0$. Par conséquent, afin de déterminer le minimum de la fonction f, il suffit de considérer les points où la dérivée de f s'annule.

Cependant, supposons maintenant que le problème ne soit plus simplement de déterminer le minimum de la fonction f, mais plutôt de trouver le point $\mathbf{x}$ qui minimise la valeur de $f(\mathbf{x})$ tout en respectant aussi la contrainte $g(\mathbf{x}) = 0$, où g est aussi une fonction définie sur le domaine D. Dans ce cas, ce point n'annulera pas nécessairement la dérivée de f, puisque l'ajout de la contrainte restreint f à un ensemble $S = \{\mathbf{x} \in D : g(\mathbf{x}) = 0\}$ et que le point qui minimise f dans S ne la minimise pas nécessairement dans D.

Le but de l'optimisation est de trouver une solution à un tel problème. De manière générale, un problème d'optimisation est un problème dans lequel on cherche un point qui minimise ou qui maximise une certaine fonction et qui est sujet à certaines contraintes. Il s'agit donc d'un problème s'écrivant ainsi :

$$\text{Minimiser } f(\mathbf{x}) \text{ pour } \mathbf{x} \in D \text{ sujet à } \begin{cases} g_i(\mathbf{x}) \leq 0 & i = 1, \ldots, k, \\ h_j(\mathbf{x}) = 0 & j = 1, \ldots, m. \end{cases}$$

La fonction f est appelée la fonction objectif, les fonctions g_i $(i = 1, \ldots, k)$ sont appelées les contraintes d'inégalité et les fonctions h_j $(j = 1, \ldots, m)$ sont appelées les contraintes d'égalité. Puisque maximiser $f(\mathbf{x})$ est équivalent à minimiser $-f(\mathbf{x})$ et que toutes les contraintes peuvent être réécrites soit sous la forme $g(\mathbf{x}) \leq 0$ ou sous la forme $h(\mathbf{x}) = 0$, tous les problèmes d'optimisation peuvent s'écrire sous la forme ci-dessus.

1.2. La convexité

Introduisons maintenant quelques définitions. Un ensemble $D \subseteq \mathbb{R}^n$ est dit *convexe* si pour tout $\mathbf{x}_1, \mathbf{x}_2 \in D$ et pour tout $\theta \in]0, 1[$,

$$(\theta \mathbf{x}_1 + (1 - \theta)\mathbf{x}_2) \in D.$$

Une fonction $f : D \to \mathbb{R}$ est dite *convexe* si pour tout $\mathbf{x}_1, \mathbf{x}_2 \in D$ (où D est un ensemble convexe) et pour tout $\theta \in]0, 1[$,

$$f(\theta \mathbf{x}_1 + (1 - \theta)\mathbf{x}_2) \leq \theta f(\mathbf{x}_1) + (1 - \theta) f(\mathbf{x}_2).$$

On dit que la fonction f est *strictement convexe* si l'inégalité précédente est stricte.

Enfin, un problème d'optimisation est dit *convexe* si le domaine D, la fonction objectif et toutes les contraintes sont convexes.

La convexité possède une propriété très intéressante qui sera essentielle plus loin : si f est une fonction convexe et $\mathbf{x}^*$ un point tel que

$$\frac{\partial f(\mathbf{x}^*)}{\partial \mathbf{x}} = 0,$$

alors $\mathbf{x}^*$ est un minimum global de f. De plus, si f est strictement convexe, alors $\mathbf{x}^*$ est l'unique point où f atteint ce minimum global (voir [**1**] pour la

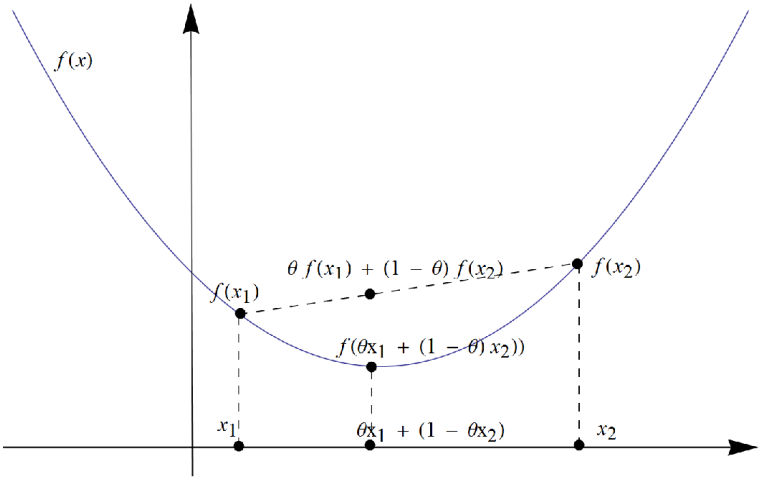

FIGURE 1. Une fonction convexe

démonstration). Notons qu'ici et pour le reste de ce texte, si $\mathbf{x} = (x_1, \ldots, x_n)$, alors

$$\frac{\partial f(\mathbf{x}_a)}{\partial \mathbf{x}} = \left(\frac{\partial f(\mathbf{x}_a)}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x}_a)}{\partial x_n} \right).$$

1.3. La dualité et le théorème de Kuhn-Tucker

Considérons le problème d'optimisation suivant :

$$\text{minimiser } f(\mathbf{x}) \text{ pour } \mathbf{x} \in D \text{ sujet à } \begin{cases} g_i(\mathbf{x}) \leq 0 & i = 1, \ldots, k, \\ h_j(\mathbf{x}) = 0 & j = 1, \ldots, m \end{cases}$$

où $D \subseteq \mathbb{R}^n$, $f : D \to \mathbb{R}$ est la fonction objectif, $g_i : D \to \mathbb{R}$ $(i = 1, \ldots, k)$ sont les contraintes d'inégalité et $h_j : D \to \mathbb{R}$ $(j = 1, \ldots, m)$ sont les contraintes d'égalité. À partir de ce problème, il est possible de définir la fonction suivante, appelée le Lagrangien :

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{x}) + \sum_{j=1}^{m} \beta_j h_j(\mathbf{x}) = f(\mathbf{x}) + \boldsymbol{\alpha}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\beta}^T \mathbf{h}(\mathbf{x})$$

où $\alpha_i, \beta_j \in \mathbb{R}$ $(i = 1, \ldots, k,\ j = 1, \ldots, m)$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$, $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_k(\mathbf{x}))$ et $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_m(\mathbf{x}))$. Les constantes α_i $(i = 1, \ldots, k)$ et β_j $(j = 1, \ldots, m)$ sont appelées les *multiplicateurs de Lagrange*.

Il est possible d'utiliser le Lagrangien pour construire un nouveau problème d'optimisation, appelé le problème dual, qui possède des liens très intéressants avec le problème original, que nous appellerons dorénavant le problème primal.

Considérons la fonction

$$\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\mathbf{x} \in D} L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}).$$

Il s'agit d'une fonction qui prend la valeur minimale du Lagrangien pour un $\boldsymbol{\alpha}$ et un $\boldsymbol{\beta}$ donnés. Soumettons maintenant cette fonction à la contrainte $\boldsymbol{\alpha} \geq 0$. Alors, si $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ respecte cette contrainte,

$$\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq f(\mathbf{x})$$

pour tout $\mathbf{x} \in D$ tel que $g_i(\mathbf{x}) \leq 0$ $(i = 1, \ldots, k)$ et $h_j(\mathbf{x}) = 0$ $(j = 1, \ldots, m)$.

$$\begin{aligned} \text{En effet,} \quad \theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \inf_{\mathbf{u} \in D} L(\mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &\leq L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= f(\mathbf{x}) + \boldsymbol{\alpha}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\beta}^T \mathbf{h}(\mathbf{x}) \\ &\leq f(\mathbf{x}) \end{aligned}$$

puisque $\mathbf{h}(\mathbf{x}) = 0$ et $\mathbf{g}(\mathbf{x}) \leq 0$, et $\boldsymbol{\alpha} \geq 0$.

Une conséquence immédiate de ce fait est que la valeur maximale que peut prendre $\theta(\boldsymbol{\alpha}, \boldsymbol{\beta})$ en respectant la contrainte $\boldsymbol{\alpha} \geq 0$ est bornée supérieurement par la valeur minimale du problème primal, c'est-à-dire que

$$\sup \{\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) : \boldsymbol{\alpha} \geq 0\} \leq \inf \{f(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{h}(\mathbf{x}) = 0\}.$$

Par conséquent, s'il existe une solution réalisable $\mathbf{x}^*$ du problème primal telle que $f(\mathbf{x}^*) = \theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$, avec $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \in \mathbb{R}^{l+m}$ et $\boldsymbol{\alpha}^* \geq 0$, alors $\mathbf{x}^*$ correspond au minimum du problème primal et $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ correspond au maximum de $\theta(\boldsymbol{\alpha}, \boldsymbol{\beta})$ sujette à la contrainte $\boldsymbol{\alpha} \geq 0$.

Ceci amène à définir ainsi un problème d'optimisation dual :

$$\text{maximiser } \theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) \text{ sujet à } \boldsymbol{\alpha} \geq 0.$$

Si la valeur maximale du problème dual correspond à la valeur minimale du problème primal, alors il est possible de résoudre le problème dual pour découvrir la solution du problème primal. Cependant, en général, il n'est pas toujours certain que les solutions du problème primal et du problème dual coïncident. Il peut en effet exister un *écart de dualité* (*duality gap*) entre les deux solutions. Cependant, le théorème de la dualité forte assure que dans le cas où la fonction f est convexe et où les contraintes sont des fonctions affines, c'est-à-dire qu'elles sont de la forme $A\mathbf{x} + \mathbf{b}$, avec A une matrice et $\mathbf{b}$ un vecteur, l'écart de dualité est nul, et donc la valeur maximale du problème dual correspond à la valeur minimale du problème primal.

Ceci permet d'établir des conditions nécessaires et suffisantes pour qu'une solution du problème primal soit optimale. Ces conditions sont présentées dans le théorème de Karush-Kuhn-Tucker.

Théorème de Karush-Kuhn-Tucker : Soit un problème d'optimisation convexe de domaine $D \subseteq \mathbb{R}^n$:

$$\text{minimiser } f(\mathbf{x}) \text{ pour } \mathbf{x} \in D \text{ sujet à } \begin{cases} g_i(\mathbf{x}) \leq 0 & i = 1, \ldots, k, \\ h_j(\mathbf{x}) = 0 & j = 1, \ldots, m \end{cases}$$

avec $f \in C^1$ et toutes les contraintes affines. Alors, un point $\mathbf{x}^*$ est un optimum si et seulement si il existe des vecteurs $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}^*$ tels que

$$\begin{aligned}
\frac{\partial L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \mathbf{x}} &= 0, \\
\frac{\partial L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} &= 0, \\
\alpha_i^* g_i(\mathbf{x}^*) &= 0, \qquad i = 1, \ldots, k, \\
g_i(\mathbf{x}^*) &\leq 0, \qquad i = 1, \ldots, k, \\
\alpha_i^* &\geq 0, \qquad i = 1, \ldots, k. \quad \square
\end{aligned}$$

Ces conditions correspondent à imposer l'existence d'une solution réalisable du dual ayant la même valeur. En effet, imposer

$$\frac{\partial L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \mathbf{x}} = 0$$

revient à s'assurer que $\theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \inf_{\mathbf{x} \in D} L(\mathbf{x}, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$, puisque le minimum du Lagrangien sans contrainte est atteint lorsque sa dérivée est nulle. En effet, puisque la fonction objectif est une fonction convexe et que les contraintes sont des fonctions affines, il est simple de vérifier que le Lagrangien est une fonction convexe. Or, on a vu que tout point qui annule la dérivée d'une fonction convexe est un minimum global de cette fonction. Par conséquent, le minimum du Lagrangien sans contraintes est bien atteint lorsque sa dérivée est nulle.

Les conditions $\frac{\partial L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} = 0$ et $g_i(\mathbf{x}^*) \leq 0, i = 1, \ldots, k$ sont simplement les contraintes du problème primal (remarquons que dériver le Lagrangien par rapport à β_j donne h_j), et visent seulement à assurer que la solution est réalisable pour le problème primal. Ensuite, la condition $\boldsymbol{\alpha}^* \geq 0$ assure que $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ est une solution réalisable du dual. Enfin, la condition $\alpha_i^* g_i(\mathbf{x}^*) = 0$ pour tout $i = 1, \ldots, k$, appelée la condition complémentaire de Karush-Kuhn-Tucker, assure que $\theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = f(\mathbf{x}^*)$. En effet, on a alors $\beta_j h_j(\mathbf{x}^*) = 0$ pour tout $j = 1, \ldots, m$ et $\alpha_i^* g_i(\mathbf{x}^*) = 0$ pour tout $i = 1, \ldots, k$. Par conséquent,

$$f(\mathbf{x}^*) = f(\mathbf{x}^*) + \boldsymbol{\alpha}^{*T} \mathbf{g}(\mathbf{x}^*) + \boldsymbol{\beta}^{*T} \mathbf{h}(\mathbf{x}^*) = L(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*).$$

Comme $f(\mathbf{x}^*) = \theta(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$, $\mathbf{x}^*$ est une solution optimale pour le problème primal et $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ est une solution optimale pour le dual. Les conditions de Karush-Kuhn-Tucker peuvent être très utiles pour vérifier si une solution est optimale.

2. Machines à vecteurs de support pour données linéairement séparables

2.1. Introduction aux machines à vecteurs de support

Maintenant que certains éléments importants de la théorie de l'optimisation ont été présentés, nous sommes en mesure d'introduire les machines à vecteurs de support. Les machines à vecteurs de support (SVM) sont un algorithme dont le but est de résoudre les problèmes de discrimination à deux classes. On appelle

problème de discrimination à deux classes un problème dans lequel on tente de déterminer la classe à laquelle appartient un individu (individu est ici employé au sens de constituant d'un ensemble) parmi deux choix possibles. En réalité, plusieurs méthodes ont été suggérées pour étendre l'application des SVM aux problèmes de discrimination à plus de deux classes (voir par exemple [**3**] et [**4**]), et il existe aussi une modification qui permet de les utiliser pour la régression (voir notamment [**2**] et [**6**]), mais nous nous concentrerons ici sur les problèmes de discrimination à deux classes.

Pour ce faire, on utilise les caractéristiques connues de cet individu. Ces n caractéristiques sont représentées par un vecteur $\mathbf{x} \in \mathbb{R}^n$. La classe à laquelle appartient l'individu est représentée par $y \in \{-1, 1\}$, où une des classes possible est représentée par -1 et l'autre par 1. Par conséquent, avec cette notation, le problème est de déterminer la valeur de y en se servant de $\mathbf{x}$.

Pour y parvenir, les machines à vecteurs de support utilisent un ensemble de données pour lesquelles le classement est déjà connu et s'en servent pour construire une règle qui permet d'effectuer une bonne classification. Cet ensemble de données est appelé l'ensemble d'apprentissage. La règle trouvée avec l'ensemble d'apprentissage doit être la plus générale possible, puisqu'il faut aussi qu'elle soit bonne pour de nouvelles données qui n'étaient pas dans l'ensemble d'apprentissage. Nous présentons ici comment les SVM font pour trouver cette règle dans le cas le plus simple possible, c'est-à-dire le cas où les données sont linéairement séparables. Les autres cas seront traités dans les sections 3 et 4.

2.2. Hyperplan séparateur

Supposons que nous disposons d'un ensemble d'apprentissage de l données de la forme $(\mathbf{x_i}, y_i) \in \mathbb{R}^n \times \{-1, 1\}$ $(i = 1, \ldots, l)$, dont nous voulons nous servir pour déterminer une règle permettant de classer les données. Supposons aussi que ces données sont linéairement séparables, c'est-à-dire qu'il existe un hyperplan dans $\mathbb{R}^n$ tel que toutes les données appartenant à la classe 1 se retrouvent d'un côté de l'hyperplan alors que celles de la classe -1 se situent de l'autre côté.

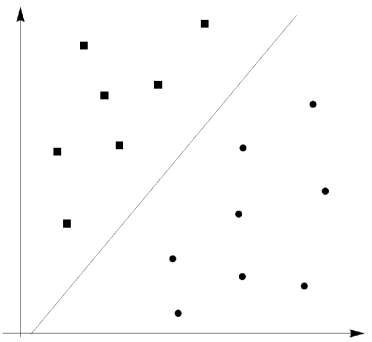

FIGURE 2. Des données linéairement séparables

Plus formellement, les données sont dites linéairement séparables s'il existe un hyperplan

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

tel que $\mathbf{w} \cdot \mathbf{x} + b > 0$ pour tout $\mathbf{x}$ appartenant à la classe 1, et $\mathbf{w} \cdot \mathbf{x} + b < 0$ pour tout $\mathbf{x}$ appartenant à la classe -1, avec $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{R}^n$ le vecteur des coefficients de l'hyperplan et $b \in \mathbb{R}$ un scalaire appelé le biais (remarquons que tout hyperplan peut s'écrire sous cette forme). Nous dirons d'un tel hyperplan qu'il sépare les données.

Sous l'hypothèse que les données sont linéairement séparables, trouver une règle pour les classer est très simple. En effet, il suffit de prendre un hyperplan qui sépare les classes, puis de classer les données selon le côté de l'hyperplan où elles se trouvent. Plus formellement, soit

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

un hyperplan qui sépare les données. Alors, il suffit d'utiliser la fonction suivante (parfois appelée la *fonction indicatrice*) pour effectuer la classification :

$$\text{Classe}\,(\mathbf{x}) = \text{signe}\,(\mathbf{w} \cdot \mathbf{x} + b),$$

où

$$\text{signe}\,(\mathbf{w} \cdot \mathbf{x} + b) = \begin{cases} -1 & \text{si } \mathbf{w} \cdot \mathbf{x} + b < 0 \\ 0 & \text{si } \mathbf{w} \cdot \mathbf{x} + b = 0 \\ 1 & \text{si } \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

Cette fonction classe les données par rapport au côté de l'hyperplan où elles se trouvent. On remarque que si un ensemble de données est séparé par un hyperplan, il sera parfaitement classé par cette fonction. Notons que si une donnée est directement sur l'hyperplan (ce qui peut arriver en considérant des données qui ne sont pas dans l'ensemble d'apprentissage), elle sera assignée à la classe 0, ce qui signifie qu'elle ne peut être classée par le modèle actuel. Dans ce cas, il est possible de la laisser inclassée, d'utiliser une autre règle ou de l'assigner aléatoirement à l'une des deux classes.

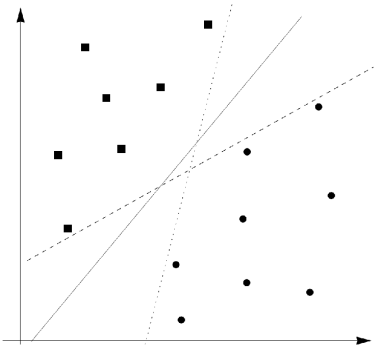

FIGURE 3. Il existe une infinité d'hyperplans pouvant séparer les données

Grâce à la fonction indicatrice, on constate qu'il suffit de trouver un hyperplan qui sépare les données pour déterminer une règle permettant de les classer. Cependant, si les données sont linéairement séparables, il existe une infinité d'hyperplans qui peuvent servir de séparateurs. L'idée des machines à vecteurs de support est de choisir le meilleur hyperplan, c'est-à-dire celui qui donnera la règle

qui se généralisera le mieux à d'autres données que celles de l'ensemble d'apprentissage. Afin de déterminer ce qui caractérise le meilleur hyperplan, introduisons le concept de marge.

2.3. Marge et hyperplan canonique

Définissons la marge d'un hyperplan comme étant la distance entre l'hyperplan et la donnée la plus proche. Plus formellement, si $\text{dist}(\mathbf{x}, \mathbf{w}, b)$ représente la distance euclidienne entre le point $\mathbf{x}$ et l'hyperplan $\mathbf{w} \cdot \mathbf{x} + b = 0$, alors la marge M est définie ainsi :

$$M = \min \{\text{dist}(\mathbf{x}_i, \mathbf{w}, b) : i = 1, \ldots, l\}$$

où les $\mathbf{x}_i$ sont les données de l'ensemble d'apprentissage. Par abus de langage, nous dirons dans ce texte qu'un point se trouve *sur la marge* si sa distance avec l'hyperplan correspond exactement à la marge.

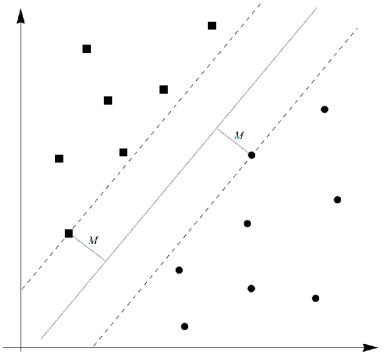

FIGURE 4. La marge

D'après un résultat de la théorie de l'apprentissage statistique, l'hyperplan qui aura la meilleure généralisation est celui qui possède la plus grande marge (le lecteur intéressé pourra consulter [**2**] et [**5**] pour plus de détails). Ce concept est à la base des machines à vecteurs de support. Dans le cas le plus simple, c'est-à-dire celui où les données sont linéairement séparables, les SVM trouvent l'hyperplan qui sépare les données avec la plus vaste marge possible, puis utilisent cet hyperplan pour classer de nouvelles données à l'aide de la fonction indicatrice donnée plus haut.

Toutefois, le problème de trouver l'hyperplan avec la marge maximale est mal posé, puisqu'il existe en réalité une infinité de manières différentes d'écrire le même hyperplan. En effet, supposons que l'hyperplan

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

soit un hyperplan dont la marge est maximale, et soit $\lambda \in \mathbb{R}^+ \setminus \{0\}$. Alors, l'hyperplan

$$\lambda \mathbf{w} \cdot \mathbf{x} + \lambda b = 0$$

est en réalité le même hyperplan et sépare les données, puisque λ est positif. Par conséquent, $\lambda \mathbf{w} \cdot \mathbf{x} + \lambda b = 0$ correspond aussi à l'hyperplan dont la marge est maximale, mais possède un vecteur des coefficients et un biais différents (si $\lambda \neq 1$).

Le nombre infini de manières d'écrire la solution du problème de l'hyperplan avec la plus vaste marge complique sa résolution. Afin de rendre le problème bien posé, introduisons le concept d'*hyperplan canonique*. Un hyperplan $\mathbf{w} \cdot \mathbf{x} + b = 0$ est dit canonique si

$$\min \{|\mathbf{w} \cdot \mathbf{x}_i + b| : i = 1, \ldots, l\} = 1,$$

où les $\mathbf{x}_i$ sont les données d'apprentissage.

On peut montrer que ce minimum correspond aux données qui sont directement sur la marge.

On peut aussi montrer que tout hyperplan qui sépare les données peut s'écrire sous forme canonique et qu'il n'existe qu'une seule façon d'écrire un hyperplan pour qu'il soit canonique. Ainsi, en ne considérant que les hyperplans canoniques, chaque hyperplan s'écrit de manière unique. De plus, il n'existe qu'un seul hyperplan pour lequel la marge est maximale. Ceci deviendra évident un peu plus loin, puisque le vecteur des coefficients de l'hyperplan sera exprimé comme étant le point qui minimise une fonction strictement convexe (rappelons que les fonctions strictement convexes n'ont qu'un unique minimum global). Par conséquent, en ne considérant que les hyperplans canoniques, le problème de trouver l'hyperplan avec la plus grande marge est bien posé.

2.4. Trouver l'hyperplan

On peut montrer que pour un hyperplan canonique $\mathbf{w} \cdot \mathbf{x} + b = 0$, la marge M est donnée par l'expression

$$M = \frac{1}{\|\mathbf{w}\|},$$

où $\|\mathbf{w}\| = \sqrt{w_1^2 + \cdots + w_n^2}$. On voit donc que plus $\|\mathbf{w}\|$ est petite, plus la marge de l'hyperplan canonique correspondant est grande. Ainsi, afin de trouver l'hyperplan qui sépare le mieux les données, il faut trouver celui qui respecte les conditions d'un hyperplan canonique et pour lequel $\|\mathbf{w}\|$ est minimale.

La recherche du meilleur hyperplan peut donc s'écrire sous la forme du problème d'optimisation suivant :

$$\text{minimiser } \|\mathbf{w}\| \text{ sujet à } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, l.$$

Les contraintes $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$, $i = 1, \ldots, l$ assurent d'une part que l'hyperplan sépare les données correctement, et d'autre part qu'il est canonique. En effet, $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 0$ si et seulement si $\text{signe}\,(\mathbf{w} \cdot \mathbf{x}_i + b) = y_i$, donc si et seulement si $\mathbf{x}_i$ est du bon côté de l'hyperplan. Ainsi, l'hyperplan doit correctement séparer les données. Ensuite, on peut montrer qu'imposer $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ assure que pour toutes les données qui ne sont pas sur la marge, $|\mathbf{w} \cdot \mathbf{x}_i + b| > 1$ et que $|\mathbf{w} \cdot \mathbf{x}_i + b| = 1$ pour les données sur la marge, donc que l'hyperplan est canonique.

Nous avons ainsi formulé un problème d'optimisation dont la solution optimale est l'hyperplan canonique séparant les données avec la plus vaste marge possible. Cependant, il est possible de formuler un problème équivalent, mais avec une fonction objectif plus simple. En effet, comme

$$\|\mathbf{w}\| = \sqrt{\mathbf{w} \cdot \mathbf{w}},$$

minimiser $\|\mathbf{w}\|$ est équivalent à minimiser $\mathbf{w}\cdot\mathbf{w}$. Évidemment, minimiser $\mathbf{w}\cdot\mathbf{w}$ est équivalent à minimiser $\frac{1}{2}\mathbf{w}\cdot\mathbf{w}$ (cette petite modification permet d'éviter d'avoir une constante dans la représentation duale du problème, comme nous le verrons un peu plus loin). Par conséquent, afin de trouver l'hyperplan canonique qui sépare les données avec la plus grande marge possible, il suffit de résoudre le problème d'optimisation suivant :

$$\text{minimiser } \frac{1}{2}\mathbf{w}\cdot\mathbf{w} \text{ sujet à } y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1, \quad i=1,\ldots,l.$$

Une propriété très intéressante de ce problème est que

$$f(\mathbf{w})=\mathbf{w}\cdot\mathbf{w}=w_1^2+\cdots+w_n^2$$

est une fonction strictement convexe. Ceci assure qu'il n'y a pas de minimum relatif et qu'il n'existe qu'une unique solution optimale.

2.5. Représentation duale

Il serait possible de résoudre le problème d'optimisation ci-dessus directement. Toutefois, sa représenta-tion duale possède des propriétés très intéressantes qui auront des répercussions majeures lorsque nous considérerons les machines à vecteurs de supports pour le cas où les données ne sont pas linéairement séparables.

Commençons tout d'abord par écrire le Lagrangien. Pour ce faire, il est nécessaire de réécrire les contraintes ainsi :

$$-(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1)\leq 0, \qquad i=1,\ldots,l.$$

Le Lagrangien est

$$L(\mathbf{w},b,\boldsymbol{\alpha})=\frac{1}{2}\mathbf{w}\cdot\mathbf{w}-\sum_{i=1}^{l}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1).$$

Il faut maintenant calculer la fonction objectif du problème dual. Rappelons que cette fonction correspond à la valeur minimale du Lagrangien pour un $\boldsymbol{\alpha}$ donné. Or, ce minimum correspond au point où la dérivée du Lagrangien par rapport aux variables du primal est nulle. On a donc ainsi

$$\begin{aligned}\frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial \mathbf{w}} &= \mathbf{w}-\sum_{i=1}^{l}\alpha_i y_i\mathbf{x}_i=0,\\ \frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial b} &= -\sum_{i=1}^{l}\alpha_i y_i=0,\end{aligned}$$

ce qu'il est possible de réécrire de cette manière :

$$\mathbf{w}=\sum_{i=1}^{l}\alpha_i y_i\mathbf{x}_i \text{ et } \sum_{i=1}^{l}\alpha_i y_i=0.$$

Utilisons ces équations pour réécrire le Lagrangien minimal uniquement en fonction des variables duales :

$$\begin{aligned}\min_{\mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{l}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1) \\ &= \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i\alpha_j y_i y_j \mathbf{x_i}\cdot\mathbf{x_j}).\end{aligned}$$

Ainsi, nous avons le problème dual suivant :

$$\text{maximiser } \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j)$$

$$\text{sujet à } \begin{cases}\sum_{i=1}^{l}\alpha_i y_i = 0, & \\ \alpha_i \geq 0 & i = 1, \ldots, l.\end{cases}$$

On remarque qu'il est nécessaire d'ajouter la contrainte $\sum_{i=1}^{l}\alpha_i y_i = 0$ pour s'assurer que $\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j)$ correspond bien au minimum du Lagrangien. En effet, alors que la contrain-te $\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i$ est automatiquement satisfaite par la construction de la fonction, rien n'assure que $\sum_{i=1}^{l}\alpha_i y_i = 0$ est elle aussi respectée.

La solution de ce problème d'optimisation sera bien sûr un vecteur $\boldsymbol{\alpha}^* = (\alpha_1^*, \ldots, \alpha_l^*)$, alors que c'est l'équation d'un hyperplan qu'il faut pour classer les données à l'aide de la fonction indicatrice

$$\text{Classe}\,(\mathbf{x}) = \text{signe}\,(\mathbf{w}\cdot\mathbf{x} + b).$$

Il est toutefois possible de réécrire la fonction indicatrice ainsi, puisque $\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i$:

$$\text{Classe}\,(\mathbf{x}) = \text{signe}\left(\sum_{i=1}^{l}(\alpha_i y_i \mathbf{x}_i\cdot\mathbf{x}) + b\right).$$

Tout ce qu'il manque pour pouvoir utiliser cette fonction est la valeur de b. Or, comme l'hyperplan est canonique, et d'après les contraintes du problème primal, si une donnée $\mathbf{x}_m$ se trouve sur la marge, alors on sait que

$$y_m(\mathbf{w}\cdot\mathbf{x}_m + b) = 1.$$

Donc,

$$b = \frac{1}{y_m} - \mathbf{w}\cdot\mathbf{x}_m = y_m - \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i\cdot\mathbf{x}_m \text{ puisque } y_m \in \{-1, 1\}.$$

Ainsi, la résolution du problème dual permet de construire l'hyperplan canonique séparant les données avec la plus grande marge et de l'utiliser pour classer des données, tout comme la résolution du problème primal.

2.6. Vecteurs de support

Comme la fonction $\mathbf{w} \cdot \mathbf{w}$ est une fonction convexe continue et dérivable, que les contraintes $-(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1) \leq 0$ sont des fonctions affines et que le domaine du problème est $\mathbb{R}^n$, la solution optimale trouvée respecte nécessairement les conditions de Karush-Kuhn-Tucker. En particulier, elle respecte la condition complémentaire de Karush-Kuhn-Tucker, c'est-à-dire que

$$\alpha_i^*(y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1) = 0, \qquad i = 1, \ldots, l,$$

où $\boldsymbol{\alpha}^*$ représente la solution optimale du problème dual et $(\mathbf{w}^*, b^*)$ représente celle du problème primal.

Cette condition implique que si $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1 \neq 0$, alors $\alpha_i = 0$. Par conséquent, les seuls cas où α_i peut ne pas être nul sont ceux où $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1 = 0$, c'est-à-dire ceux où

$$y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) = 1.$$

Or, les seuls points où $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) = 1$ sont ceux qui sont sur la marge. Par conséquent, seuls les points sur la marge peuvent avoir des α_i non nuls. Ces points sont appelés les *vecteurs de support*.

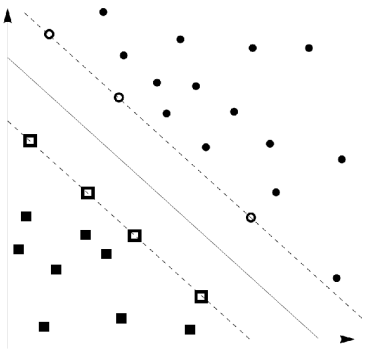

FIGURE 5. Les vecteurs de support

La raison de ce nom est que ce sont les seuls points utiles pour déterminer l'hyperplan. En effet, rappelons que le vecteur des coefficients de l'hyperplan est donné par

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i.$$

Ainsi, tout point qui n'est pas sur la marge n'apporte aucune contribution, puisque α_i est alors nul. Si tous les points sauf les vecteurs de support étaient retirés de l'ensemble d'apprentissage, on retrouverait le même hyperplan.

Les vecteurs de support peuvent donc être vus comme les points contenant toute l'information essentielle du problème.

3. Machines à vecteurs de support pour données non linéairement séparables

3.1. Transformations

Jusqu'à présent, les machines à vecteurs de support permettent de trouver une règle pour classer les données lorsque celles-ci sont linéairement séparables. Cependant, il existe bien des cas pour lesquels il est impossible de séparer entièrement les données avec un hyperplan. Telles qu'elles ont été présentées jusqu'à présent, les SVM sont incapables de traiter un tel problème, puisqu'il est alors impossible que les contraintes

$$y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) \geq 1$$

soient toutes respectées.

Afin de régler ce problème, il est possible d'appliquer une transformation aux données de sorte qu'une fois transformées, elles soient linéairement séparables. L'espace où se trouvent les données avant d'être transformées est appelé *l'espace d'entrée* (*input space*), alors qu'après avoir appliqué la transformation, les données se trouvent dans ce qu'on appelle *l'espace de redescription* (*feature space*). Il suffit alors de trouver l'hyperplan dans l'espace de redescription qui sépare le mieux ces données transformées. De retour dans l'espace d'entrée, le séparateur n'est pas linéaire.

$$\begin{aligned} \text{Soit } \Phi : \mathbb{R}^n &\to \mathbb{R}^r \\ \mathbf{x} &\to \Phi(\mathbf{x}) \end{aligned}$$

la transformation appliquée aux données pour les rendre linéairement séparables, avec r la dimension de l'espace de redescription. Très souvent, $r > n$, ce qui signifie que la transformation amène les données dans un espace de dimension supérieure afin de mieux pouvoir les séparer.

Pour trouver le séparateur, on procède de la même manière que précédemment, mais en substituant $\Phi(\mathbf{x}_i)$ à $\mathbf{x}_i$ $(i = 1, \ldots, l)$. Il s'agit donc de résoudre le problème suivant :

$$\text{minimiser } \frac{1}{2}\mathbf{w} \cdot \mathbf{w} \text{ sujet à } y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, \ldots, l.$$

Le dual de ce problème est

$$\text{maximiser } \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$$

$$\text{sujet à } \begin{cases} \sum_{i=1}^{l} \alpha_i y_i = 0, \\ \alpha_i \geq 0 \qquad\qquad i = 1, \ldots, l. \end{cases}$$

La fonction indicatrice associée à ce problème dual est par conséquent

$$\text{Classe}\,(\mathbf{x}) = \text{signe}\left(\sum_{i=1}^{l} (\alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) + b \right),$$

$$\text{où} \quad b = y_m - \sum_{i=1}^{l} \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_m)$$

avec $\Phi(\mathbf{x}_m)$ une donnée sur la marge de l'hyperplan dans l'espace de redescription.

Si la transformation utilisée est appropriée, la résolution d'un de ces problèmes (le primal ou le dual) permet de trouver un séparateur non linéaire avec la marge la plus grande possible, permettant ainsi d'utiliser les machines à vecteurs de support dans le cas où les données ne peuvent pas être séparées linéairement.

3.2. Les noyaux

Toutefois, l'utilisation des transformations pose certains problèmes. En effet, outre le fait qu'il faille choisir une bonne transformation, il faut l'appliquer à toutes les données, puis effectuer les calculs avec ces données transformées, c'est-à-dire dans l'espace de redescription. Or, comme la dimension de cet espace est bien souvent beaucoup plus grande que celle de l'espace d'entrée, les calculs requis peuvent devenir extrêmement longs à effectuer.

C'est ici que la formulation duale du problème d'optimisation prend toute son importance. En effet, on remarque que lorsque le problème est sous sa forme duale, les données de l'ensemble d'apprentissage n'apparaissent que dans un produit scalaire avec d'autres données du même ensemble. Il en est de même dans la fonction indicatrice duale. Ceci amène à définir comme suit une fonction appelée *noyau* (*kernel*) :

$$\begin{aligned} K : \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R} \\ (\mathbf{x}_i, \mathbf{x}_j) &\to \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \end{aligned}$$

Cette fonction prend en entrée deux points dans l'espace d'entrée et calcule leur produit scalaire dans l'espace de redescription. L'avantage d'une telle fonction est qu'il n'est pas nécessaire d'appliquer une transformation aux données afin de calculer leur produit scalaire dans l'espace de redescription. Ce calcul peut se faire directement à partir des données de l'espace d'entrée.

Grâce au concept de noyau, il est possible de réécrire le problème dual de cette manière :

$$\text{maximiser} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j))$$

$$\text{sujet à} \quad \begin{cases} \sum_{i=1}^{l} \alpha_i y_i = 0, \\ \alpha_i \geq 0, \qquad\qquad i = 1, \ldots, l. \end{cases}$$

La fonction indicatrice peut elle aussi être réécrite :

$$\text{Classe}\,(\mathbf{x}) = \text{signe}\left(\sum_{i=1}^{l} (\alpha_i y_i K(\mathbf{x}_i, \mathbf{x})) + b \right),$$

$$\text{où} \quad b = y_m - \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_m)$$

avec $\mathbf{x}_m$ un vecteur de support.

On remarque que de cette manière, lorsque la fonction noyau est connue, la transformation $\Phi(\mathbf{x})$ n'apparaît nulle part, ni dans le problème, ni dans l'application de la solution. Par conséquent, grâce à la fonction noyau, il n'est pas nécessaire d'effectuer la transformation sur les données. Cette fonction permet

donc de faire tous les calculs nécessaires sans avoir à se préoccuper de la dimension de l'espace de redescription.

3.3. Noyaux : Construction et exemples

Il est bien de savoir qu'un noyau est tout ce qui est nécessaire pour utiliser les SVM dans le cas non linéaire, mais cette information est inutile sans la connaissance des noyaux qu'il est possible d'utiliser. Nous présentons maintenant les manières de construire des noyaux, ainsi que les noyaux les plus fréquemment utilisés pour les machines à vecteurs de support.

La première méthode pour construire un noyau est de choisir une transformation, de calculer le produit scalaire de deux éléments quelconques transformés, puis d'en faire une fonction. Cette méthode permet de déterminer le noyau d'une transformation bien spécifique, mais peut s'avérer difficile à utiliser, surtout lorsque le nombre de dimensions augmente. Une autre méthode consiste à utiliser le théorème de Mercer. D'après ce théorème, une fonction est un noyau si et seulement si elle est symétrique et semi-définie positive (voir [**2**] pour plus de détails). Ainsi, au lieu de choisir une transformation puis de calculer son noyau, on choisit plutôt une fonction symétrique et semi-définie positive, ce qui nous assure qu'elle correspond au produit scalaire d'une quelconque transformation. La transformation elle-même est généralement inconnue pour les noyaux construits de cette façon.

Le théorème de Mercer permet aussi de construire des noyaux à partir de noyaux déjà connus. En effet, si K_1 et K_2 sont des noyaux, alors les fonctions suivantes sont aussi des noyaux :

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= K_1(\mathbf{x}_i, \mathbf{x}_j) + K_2(\mathbf{x}_i, \mathbf{x}_j); \\ K(\mathbf{x}_i, \mathbf{x}_j) &= aK_1(\mathbf{x}_i, \mathbf{x}_j) \qquad a \in \mathbb{R}; \\ K(\mathbf{x}_i, \mathbf{x}_j) &= K_1(\mathbf{x}_i, \mathbf{x}_j)K_2(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

Les noyaux présentés dans la figure 6 sont les plus fréquemment utilisés. Le choix du noyau a un impact majeur sur la performance des SVM. Quelques méthodes ont été suggérées pour sélectionner un bon noyau, mais il s'agit encore d'un sujet de recherche actif. En général, le noyau gaussien est souvent préféré, puisqu'il donne de bonnes performances dans toutes sortes de contextes.

Nom	**Noyau**
Linéaire	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)$
Polynomial de degré d	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$
Gaussien	$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{(\mathbf{x}_i - \mathbf{x}_j)\cdot(\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2}}$
Multiquadratique inverse	$K(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\sqrt{(\mathbf{x}_i - \mathbf{x}_j)\cdot(\mathbf{x}_i - \mathbf{x}_j) + \beta}}$

FIGURE 6. Les noyaux les plus fréquemment utilisés

4. Marges souples

4.1. Machines à vecteurs de support et bruit

En pratique, les données sont rarement parfaites. Il y a souvent du « bruit », c'est-à-dire des données qui sont mal classées par un modèle qui est toutefois excellent en général. Il s'agit donc d'erreurs qui sont inévitables, même pour les meilleurs modèles. Toutefois, les machines à vecteurs de support ne permettent pas de tenir compte de ce phénomène, puisque dans les contraintes, toutes les données doivent être correctement classées. Supposons par exemple qu'un ensemble de données serait très bien séparé par un hyperplan, mais qu'il n'est pas linéairement séparable dû à la présence d'un certain bruit dans les données. Dans un tel cas, il serait impossible de construire une SVM linéaire, car il est impossible que toutes les contraintes soient respectées.

Afin de contourner ce problème, il peut être tentant d'utiliser un noyau quelconque afin de rendre les données linéairement séparables. Ceci est en effet toujours possible en utilisant un noyau polynomial avec un degré suffisamment élevé. Toutefois, bien que les données de l'ensemble d'apprentissage seront parfaitement séparées, la règle trouvée risque de très mal se généraliser, puisqu'elle va tenir compte de toutes les petites variations et ainsi généraliser des phénomènes qui sont en réalité bien spécifiques à l'ensemble de données actuel.

4.2. Marge souple

Un meilleur moyen serait de permettre à quelques données d'être à l'intérieur de la marge ou du mauvais côté de l'hyperplan. Il s'agit du concept de *marge souple* (*soft margin*). Une première idée serait de tenter de maximiser la marge tout en minimisant le nombre de données mal classées. Toutefois, le nombre de données mal classées peut être trompeur, puisqu'il ne permet pas de déterminer si une donnée était presque correctement classée ou si elle était en réalité très loin de l'hyperplan.

Une meilleure idée est d'attribuer à chaque donnée $\mathbf{x}_i$ une valeur ξ_i qui représente à quel point la donnée est éloignée d'un bon classement, puis de tenter de minimiser la somme des ξ_i. Plus formellement, au lieu d'imposer

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, l$$

ce qui oblige les données à être bien classées, les contraintes seront plutôt

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l \quad \text{avec } \xi_i \geq 0.$$

Par conséquent, il est possible pour une donnée d'être du mauvais côté de la marge, si ξ_i est non nul. On dira d'une donnée qu'elle est du mauvais côté de la marge si elle est mal classée ou si sa distance par rapport à l'hyperplan séparateur est plus petite que la marge (remarquons que les points pour lesquels $\xi_i \neq 0$ ne sont pas considérés dans le calcul de la marge). L'objectif est ainsi de maximiser la marge tout en minimisant la somme des ξ_i. Le problème d'optimisation devient alors

$$\text{minimiser } \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{l} \xi_i \quad \text{sujet à} \quad \begin{cases} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i & i = 1, \ldots, l, \\ \xi_i \geq 0 \end{cases}$$

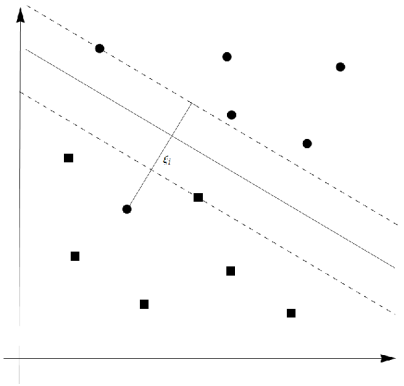

FIGURE 7. Marge souple

où $C > 0$ est une constante qui représente la pénalité d'avoir des données mal classées. Lorsque C est très élevée, il y aura très peu de données mal classées, alors qu'il y en aura plus pour une valeur plus faible de cette constante. Le choix de C a une grande influence sur le modèle. En pratique, plusieurs modèles sont souvent construits, avec différentes valeurs de C, puis le meilleur est choisi.

4.3. Représentation duale

Il est possible de construire le dual de ce problème de la même manière que précédemment. Le Lagrangien est

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r}) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C\sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{l} r_i \xi_i.$$

Afin de trouver le Lagrangien minimal pour un $(\boldsymbol{\alpha}, \mathbf{r})$ donné, il faut le dériver par rapport aux variables primales. On obtient alors

$$\begin{aligned}
\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = 0; \\
\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r})}{\partial b} &= -\sum_{i=1}^{l} y_i \alpha_i = 0; \\
\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, r)}{\partial \xi_i} &= C - \alpha_i - r_i = 0.
\end{aligned}$$

De ceci, on obtient

$$\mathbf{w} = \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i, \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \quad \text{et } C = \alpha_i + r_i \quad \text{pour tout } i \in \{1, \ldots l\}.$$

Utilisons ces expressions pour réécrire le Lagrangien uniquement en fonction des variables duales :

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r}) &= \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{l} r_i\xi_i \\ &= \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j). \end{aligned}$$

Il s'agit de la fonction objectif du problème dual pour la marge souple. On remarque que cette fonction est exactement la même que celle obtenue précédemment. La différence se situe au niveau des contraintes.

En effet, rappelons que dans le problème dual, les multiplicateurs de Lagrange qui sont associés à des contraintes d'inégalités doivent être supérieurs ou égaux à zéro (voir la sous-section 1.3). Par conséquent, $\alpha_i \geq 0$ et $r_i \geq 0$. Toutefois, r_i n'apparaît pas dans le problème dual, mais on sait que $C - \alpha_i - r_i = 0$. La contrainte $r_i \geq 0$ s'écrit donc aussi $C - \alpha_i \geq 0$. Ceci implique que $\alpha_i \leq C$.

Le problème dual est ainsi

$$\text{maximiser } \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j)$$

$$\text{sujet à } \begin{cases} \sum_{i=1}^{l}\alpha_i y_i = 0, \\ 0 \leq \alpha_i \leq C \qquad i = 1, \ldots, l. \end{cases}$$

4.4. Remarques

Les conditions de Karush-Kuhn-Tucker tiennent toujours dans le cas de la marge souple. Par consé-quent, d'après la condition complémentaire, pour la solution optimale, les égalités suivantes sont vérifiées :

$$\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i) = 0 \quad \text{pour } i = 1, \ldots, l;$$

$$r_i\xi_i = (C - \alpha_i)\xi_i = 0 \quad \text{pour } i = 1, \ldots, l.$$

Ceci implique que si $\xi_i \neq 0$, alors $C - \alpha_i = 0$ (puisque $(C - \alpha_i)\xi_i = 0$), et donc $\alpha_i = C$. De plus, si un point est tel que $\xi_i \neq 0$, alors il est du mauvais côté de la marge, ce qui découle directement du rôle de ξ_i dans le problème d'optimisation. À l'opposé, tous les points pour lesquels $\xi_i = 0$ sont du bon côté de la marge, et ainsi nécessairement bien classés.

D'autre part, si, pour une certaine donnée, on a $0 < \alpha_i < C$, alors celle-ci est exactement sur la marge. En effet, on a alors $0 < \alpha_i < C$, $\alpha_i \neq C$, et donc il faut que $\xi_i = 0$ pour que $(C - \alpha_i)\xi_i = 0$. De plus, $\alpha_i \neq 0$, ce qui implique que $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i = 0$ afin de respecter l'égalité $\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i) = 0$. Comme $\xi_i = 0$, il s'ensuit que

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1$$

et donc que $\mathbf{x}_i$ est directement sur la marge.

Enfin, les points pour lesquels $\xi_i = 0$ et $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \neq 1$ ont un α_i nul, afin de respecter l'égalité $\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i) = 0$.

Les points directement sur la marge sont appelés *vecteurs de support libres* (*free support vectors*), ou encore *vecteurs de support non-bornés* (*unbounded support vectors*). Les points pour lesquels $\alpha_i = C$ sont quant à eux appelés *vecteurs*

de support bornés (*bounded support vectors*). Ici encore, les vecteurs de support sont les seuls point qui sont vraiment important pour déterminer l'hyperplan optimal, puisque ce sont les seuls points pour lesquels $\alpha_i \neq 0$.

Enfin, remarquons que bien que la marge souple ait été présentée pour le cas linéaire, il est possible de l'utiliser aussi dans le cas non linéaire exactement de la même manière, puisque la fonction objectif du dual est parfaitement identique à celle de la marge non souple. Il suffit donc encore de remplacer tous les produits scalaires par une fonction noyau.

Références

[1] Berliaire, M. : *Introduction à l'optimisation différentiable*, Presses polytechniques et universitaires romandes, 2006, 532 p.

[2] Christianini, N. et Shawe-Taylor, J. : *An Introduction to Support Vector Machines and other kernel based learning methods*, Cambridge University Press, Cambridge, 2000, 189 p.

[3] Orchard, B., Yang, C. et Ali, M. : *Innovation in Applied Artificial Intelligence : 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Springer, New York, 2004, 1272 p.

[4] Perner, P. : *Machine Learning and Data Mining in Pattern Recognition : 5th International Conference*, Springer, New York, 2007, 913 p.

[5] Vapnik, V.N. : *The Nature of Statistical Learning Theory*, Springer, New York, 1995, 188 p.

[6] Wang, L. : *Support Vector Machines : Theory and Applications*, Springer, New York, 2005, 431 p.

Dominik Francoeur, Département de mathématiques, Université de Sherbrooke
Courriel: dominik.francoeur@usherbrooke.ca