Présentation des Machines à Vecteurs de Support

Sommaire

3.1 Classification supervisée

Les Machines à Vecteurs de Support (SVM) font partie d'une vaste famille d'algorithmes originellement regroupés dans le domaine de la reconnaissance de formes (*pattern recognition*). Les données sont généralement modélisées sous forme d'un vecteur aléatoire réelle $\boldsymbol{x} \in \mathbb{R}^d$ dont la génération, gouvernée par une densité de probabilité $p(\boldsymbol{x}, y)$, est dépendante de $y \in \{1, \ldots, C\}$, la classe d'appartenance de $\boldsymbol{x}$. Dans le cas particulier de la discrimination (problème à deux classes), on suppose $y \in \{+1; -1\}$, où $y_i = +1$ est associé à la classe 1, et $y_i = -1$ à la classe 2, pour alléger les notations. La densité de probabilité de $p(\boldsymbol{x}, y)$ étant généralement inconnue, on se base sur un ensemble de n réalisations $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1\ldots n}$ pour caractériser cette dernière. On distinguera par la suite les ensembles $\mathcal{S}_1 = \{(\boldsymbol{x}_i, y_i), y_i = +1\}$ et $\mathcal{S}_2 = \{(\boldsymbol{x}_i, y_i), y_i = -1\}$, avec $\mathrm{Card}(\mathcal{S}_1) = n_1$ et $\mathrm{Card}(\mathcal{S}_2) = n_2$.

Contrairement à d'autres méthodes de classification, comme les modèles à mélanges de gaussiennes (GMM), les SVM ne se basent pas sur l'estimation de la densité de probabilité, mais sur l'estimation d'une fonction de discrimination entre les exemples des deux classes. On cherche donc une fonction de décision $f : \mathbb{R}^d \to \mathbb{R}$ telle que

$$\mathrm{sign}(f(\boldsymbol{x})) = y_i.$$

Les Machines à Vecteurs de Support se distinguent en premier lieu parmi les méthodes discriminatives par le critère d'optimalité guidant le choix d'une telle fonction, mais leur émergence fait écho à de nombreuses techniques antérieures de classification supervisée.

3.2 Prélude

Fisher [76] propose en 1936 l'un des premiers algorithmes de reconnaissance de formes, qui deviendra par la suite l'*Analyse Discriminante de Fisher*. Cette dernière consiste en la détermination d'un hyperplan de séparation linéaire optimal entre les exemples des deux classes. Ainsi si l'on caractérise cet hyperplan par son vecteur normal $\boldsymbol{w}$, la fonction de décision prend la forme suivante :

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b, \tag{3.1}$$

où b est appelé le *biais*, ou *poids de seuil*. Ce problème est résolu dans le cadre de l'Analyse Discriminante de Fisher en maximisant le critère de séparation S, défini comme le rapport de la variance inter-classes sur la variance intra-classes :

$$S = \frac{\left(\boldsymbol{w}^T \boldsymbol{\mu}_1 - \boldsymbol{w}^T \boldsymbol{\mu}_2\right)^2}{\boldsymbol{w}^T \left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right) \boldsymbol{w}}, \tag{3.2}$$

où $\boldsymbol{\mu}_i$ et $\boldsymbol{\Sigma}_i$ sont respectivement le centre et la matrice de covariance des exemples de la classe i. On montre que, sous l'hypothèse de gaussianité des densités de probabilités, l'hyperplan optimal au sens du critère de séparation est caractérisé par le vecteur suivant :

$$\boldsymbol{w} = \left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right)^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right). \tag{3.3}$$

Néanmoins l'hypothèse de gaussianité est très forte et rarement rencontrée dans des situations réelles.

Les Machines à Vecteurs de Support résolvent ce handicap en basant la résolution sur un critère d'optimalité différent qui offre en outre des bases théoriques sur les propriétés de généralisation de la fonction de décision. Elles constituent au début des années 90 un carrefour entre plusieurs domaines jusque-là indépendants : la reconnaissance de formes, les réseaux de neurones, les techniques de programmation mathématique et la théorie mathématique des Noyaux Reproduisants (RKHS, *Reproducing Kernel Hilbert Spaces*) introduite par Aronszajn en 1950 [15].

3.3 Machines à Vecteurs de Support linéaires

En 1964, Vapnik et Lerner introduisent le principe de maximisation de la marge dans un algorithme (*Generalized Portrait Algorithm*) qui constituera le principe fondamental des futures Machines à Vecteurs de Support [235].

Dans le cas de données séparables, il existe une infinité d'hyperplans permettant de séparer les deux classes, comme l'illustre la figure 3.1 (dont on ne retiendra pour l'instant que les hyperplans en trait plein). Néanmoins, les données étant considérées comme des réalisations d'une variable aléatoire, le choix de l'hyperplan doit être guidé par les propriétés de généralisation de la fonction de décision.

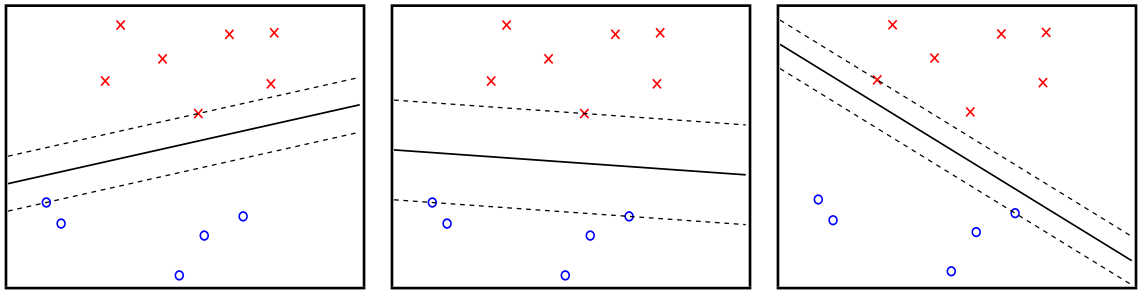

FIGURE 3.1 – Exemples d'hyperplans de séparation (en trait plein) et d'hyperplans de marge (en pointillés), entre deux classes dont les exemples sont respectivement représentés par des cercles et des croix.

La séparabilité des données implique que la contrainte $y_i f(\boldsymbol{x}_i) > 0$ est remplie pour chaque exemple. Il existe donc une valeur M, appelée *marge*, minimisant l'ensemble des distances $\frac{y_i f(\boldsymbol{x}_i)}{||\boldsymbol{w}||}$

entre les exemples et l'hyperplan :

$$\frac{y_i f(\boldsymbol{x}_i)}{\|\boldsymbol{w}\|} \geq M. \tag{3.4}$$

Le principe de l'algorithme consiste à déterminer le vecteur $\boldsymbol{w}$ maximisant la marge M. Il s'agit donc d'une optimisation de type *minimax* (ou plutôt maximin). Il est cependant nécessaire de fixer une contrainte additionnelle sur la norme de $||\boldsymbol{w}||$ afin de restreindre le champ infini de solutions ne différant que par un facteur d'échelle :

$$M \|\boldsymbol{w}\| = 1. \tag{3.5}$$

Ainsi, maximiser la marge M revient à minimiser la norme du vecteur $||\boldsymbol{w}||$. L'inéquation 3.4 devient donc la contrainte $y_i f(\boldsymbol{x}_i) \geq 1$. On appelle ainsi *hyperplans de marge* les deux hyperplans satisfaisant la condition $f(\boldsymbol{x}) = \pm 1$. La figure 3.1 montre trois exemples d'hyperplans de séparation (en trait plein) définis par des vecteurs $\boldsymbol{w}$ différents, ainsi que les hyperplans de marge correspondants (en pointillés). On peut voir que la marge, distance entre les hyperplans de marge et de séparation, diffère entre les trois cas, la figure centrale représentant la situation de marge maximale.

L'utilisation de la norme quadratique de $\boldsymbol{w}$ facilitera par la suite la résolution du problème et permettra en outre l'introduction du noyau, présentée plus bas. La recherche de l'hyperplan maximisant la marge se fait donc par la résolution du problème quadratique suivant :

$$\begin{array}{rl} \text{minimiser} & \frac{||\boldsymbol{w}||^2}{2} \\ \text{sous les contraintes} & y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) - 1 \geq 0, \quad i = 1, \ldots, n. \end{array} \tag{3.6}$$

On exprime alors le Lagrangien par l'introduction des *multiplicateurs de Lagrange* α_i (également appelés *coefficients de Kühn-Tucker*) sur les contraintes :

$$\begin{array}{l} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) - 1\right], \\ \text{avec } \alpha_i \geq 0, \quad i = 1, \ldots, n. \end{array} \tag{3.7}$$

À l'optimum, les multiplicateurs de Lagrange sont en outre soumis aux conditions suivantes :

$$\alpha_i \left(y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) - 1\right) = 0. \tag{3.8}$$

Le problème d'optimisation 3.6 revient à minimiser le Lagrangien L par rapport à $\boldsymbol{w}$ et b et à le maximiser par rapport aux variables α_i. Ils satisfait donc, au point optimal, les conditions nécessaires suivantes :

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = 0 \tag{3.9}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.10}$$

$$\text{soit :} \qquad \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i \tag{3.11}$$

Les exemples satisfaisant l'égalité $y_i f(\boldsymbol{x}_i) = 1$ sont appelés les *vecteurs de support*. Ils sont situés sur les hyperplans de marge et sont les exemples les plus proches de l'hyperplan de séparation. L'inégalité 3.8 montre que les α_i sont tous nuls à l'exception de ceux associés aux vecteurs de support. Ainsi si l'on définit $\mathcal{S}_{SV} = \{i, \alpha_i > 0\}$ l'ensemble des indices de ces vecteurs de support, on remarque que $\boldsymbol{w}$ s'exprime exclusivement en fonction de ces derniers :

$$\boldsymbol{w} = \sum_{i \in \mathcal{S}_{SV}} \alpha_i y_i \boldsymbol{x}_i. \tag{3.12}$$

C'est là un des avantages fondamentaux des Machines à Vecteurs de Support ; la contrainte de maximisation de la marge réduit le problème à un nombre restreint d'exemples, induisant ainsi

une robustesse aux exemples marginaux et de bonnes propriétés de généralisation, comme nous le verrons par la suite.

La fonction de décision prend donc la forme suivante (en développant l'équation 3.1) :

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \, y_i \, \boldsymbol{x}_i^T \boldsymbol{x} + b = \sum_{i \in \mathcal{S}_{SV}} \alpha_i \, y_i \, \boldsymbol{x}_i^T \boldsymbol{x} + b. \tag{3.13}$$

Toutefois, le problème précédent (équation 3.6), dit *primal*, est généralement compliqué à résoudre. Si l'on développe l'expression du Lagrangien (éq. 3.7) :

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \boldsymbol{w}^T \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i - b \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i \,, \tag{3.14}$$

l'injection des équations 3.10 et 3.11 fait disparaître les variables $\boldsymbol{w}$ et b et permet d'obtenir la forme *duale* du problème d'optimisation, où le Lagrangien L_D ne dépend que des multiplicateurs de Lagrange :

$$\begin{aligned} \text{maximiser} \quad & L_D(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha} \\ \text{sous les contraintes} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, n \\ \text{et} \quad & \boldsymbol{\alpha}^T\boldsymbol{y} = 0 \end{aligned} \tag{3.15}$$

où on a préféré l'écriture matricielle, plus concise, avec $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^T$, $\boldsymbol{y} = [y_1, \ldots, y_n]^T$, $\mathbf{1} = [1, \ldots, 1]^T$ et $[\boldsymbol{H}]_{ij} = y_i \, y_j \, \boldsymbol{x}_i^T \boldsymbol{x}_j$. Cette forme duale est généralement choisie pour opérer la résolution du problème d'optimisation.

3.4 Principe de Minimisation du Risque Structurel

Le principe de maximisation de la marge induit donc l'émergence d'un sous-ensemble d'exemples, appelés *vecteurs de support*, décrivant à eux seuls le comportement du classifieur. Ce principe, appliqué à la séparation linéaire dans la section précédente, peut être étendu à une multitude d'algorithmes [208], par le biais de fonctions noyaux (voir la section 3.5). L'engouement pour les Machines à Vecteurs de Support s'explique ainsi par le fait que diverses méthodes de classifications se sont trouvées fédérées dans un cadre théorique commun, mais surtout parce que leur principe est validé par la théorie de l'apprentissage statistique développée par Vapnik et Chervonenkis dans les années 70 [236].

On peut en effet reformuler le processus d'apprentissage comme la détermination d'une fonction de décision f_λ parmi un ensemble

$$\mathcal{H} = \{f_\lambda(\boldsymbol{x}), \lambda \in \Lambda\} \quad \text{avec} \quad f_\lambda : \mathbb{R}^d \to \mathbb{R}, \tag{3.16}$$

où λ est l'ensemble des paramètres ajustés durant l'apprentissage (par exemple les variables $\boldsymbol{w}$ et b dans le cas des SVM). On cherche donc une fonction f_{λ^*} qui minimise le *risque fonctionnel* suivant :

$$R(\lambda) = \int |f_\lambda(\boldsymbol{x}) - y| \, P(\boldsymbol{x}, y) d\boldsymbol{x} \, dy.$$

Néanmoins, la probabilité $P(\boldsymbol{x}, y)$ étant inconnue, il est impossible d'évaluer le risque $R(\lambda)$. N'ayant accès qu'à des réalisations de $P(\boldsymbol{x}, y)$, on calcule donc une approximation stochastique du risque, le *risque empirique* :

$$R_{emp}(\lambda) = \frac{1}{n} \sum_{i=1}^{n} |f_\lambda(\boldsymbol{x}_i) - y| \,.$$

où les $\boldsymbol{x}_i$ sont les exemples de l'ensemble d'apprentissage. La loi des grands nombres garantit la convergence du risque empirique vers le risque fonctionnel lorsque le nombre d'exemples est suffisamment conséquent. Cependant, seule une convergence uniforme peut garantir que le minimum

de R_{emp} converge vers le minimum R. Vapnik et Chervonenkis ont montré [232][233] que la condition nécessaire et suffisante pour garantir cette convergence est la finitude de la *dimension VC* (d'après les initiales des auteurs), notée h, de l'espace $\mathcal{H}$ (éq. 3.16). Celle-ci est un nombre entier naturel ou infini, défini comme le nombre maximum d'exemples pouvant être séparés de toutes les façons possibles par les fonctions de $\mathcal{H}$. Elle constitue une mesure de complexité de l'ensemble de classifieurs $\mathcal{H}$.

Un théorème de Vapnik et Chervonenkis [236] fournit une borne supérieure au risque fonctionnel, avec la probabilité $1-\eta$, décrivant l'influence de la dimension VC sur la convergence de R_{emp} vers R :

$$R(\lambda) \leq R_{emp}(\lambda) + \sqrt{\frac{1}{n}\left[h\left(\ln\frac{2n}{h}+1\right) - \ln\frac{\eta}{4}\right]}. \tag{3.17}$$

L'augmentation de la dimension VC h induit des fonctions de décision plus complexes, et par conséquent une réduction du risque empirique (du taux d'erreur). Mais ce gain, s'il n'est pas contrôlé, peut se faire au détriment de la capacité de généralisation du classifieur. Ce phénomène est connu sous le nom de *sur-apprentissage*. Il est en effet toujours possible d'obtenir un taux d'erreur nul sur tout ensemble d'apprentissage (il suffit pour cela d'en mémoriser tous les exemples), sans que le classifieur n'ait modélisé ou « *compris* » la structure des données en question. Le deuxième terme de la partie droite de l'inégalité 3.17, appelé *risque structurel*, apporte précisément l'information relative à la complexité de l'ensemble de recherche des classifieurs. Tandis que le risque empirique ($R_{emp}(\lambda)$) décroît avec l'augmentation de h, le risque structurel augmente. Il existe donc une valeur de h minimisant la borne ainsi formulée, qui traduit le compromis optimal entre le risque empirique et la complexité de la famille de classifieurs.

Le *principe de minimisation du risque structurel* est un résultat fondamental de la théorie de Vapnik et Chervonenkis mais la dimension VC reste généralement difficile à évaluer. Néanmoins, on peut montrer [236] que dans le cas où $\mathcal{H}$ décrit l'ensemble des hyperplans de séparation, si l'on impose la contrainte

$$||\boldsymbol{w}|| \leq A,$$

et si l'ensemble des exemples peuvent être contenus dans une sphère de rayon R, alors la dimension VC satisfait l'inégalité suivante

$$h \leq \min\left(\lceil R^2A^2\rceil, d\right) + 1, \tag{3.18}$$

où d est la dimension de l'espace d'entrée des fonctions de décision $f \in \mathcal{H}$. Ainsi, on utilise en général la relation suivante pour estimer la dimension VC :

$$h \approx R^2||\boldsymbol{w}||^2. \tag{3.19}$$

Le rayon R étant fixé par la répartition des exemples d'apprentissage, on voit donc que le principe de maximisation de la marge, présenté en section 3.3, qui équivaut à minimiser $||\boldsymbol{w}||$, remplit le principe de minimisation du risque structurel puisqu'il applique conjointement la minimisation du risque empirique et de la dimension VC.

On pourra trouver une introduction plus détaillée aux fondements statistiques des Machines à Vecteurs de Support dans [39] et [175], ou se référer à l'ouvrage de référence de Vapnik [236] pour une étude plus approfondie.

3.5 Noyaux

3.5.1 Introduction

Malgré une base théorique solide, les SVM restent toutefois fortement limitées par la restriction aux séparateurs linéaires. Il est en effet rare que des données réelles soient providentiellement réparties de chaque côté d'un hyperplan.

Le domaine des *classifieurs polynômiaux* étudie la mise en forme d'algorithmes basés sur des combinaisons multiplicatives de descripteurs. Ainsi, soit un exemple à d composantes $\boldsymbol{x} =$

$[x_1 \dots x_d]$, l'ensemble $\mathcal{M}^\delta$ des produits à l'ordre $\delta \in \mathbb{N}$ (appelés *monômes*) peut apporter une information non exprimée par les composantes originales :

$$\mathcal{M}^\delta = \{x_{j_1} \cdot x_{j_2} \cdot \ldots \cdot x_{j_\delta}\} \quad \text{où} \quad j_1 \leq \ldots \leq j_\delta \in \{1, \ldots, d\}\,.$$

On peut ainsi définir une transformation Φ qui porte des exemples bi-dimensionnels de l'espace originel $\mathbb{R}^2$ vers un espace de dimension supérieure contenant tous les monômes à l'ordre 2 :

$$\begin{array}{rcl} \Phi : \mathbb{R}^2 & \rightarrow & \mathbb{R}^4 \\ \boldsymbol{x} = (x_1,\, x_2) & \mapsto & (x_1^2,\, x_2^2,\, x_1x_2,\, x_2x_1). \end{array} \tag{3.20}$$

On remarque que les formulations primale et duale du problème d'optimisation des SVM (éq. 3.7 et 3.15) n'impliquent que des produits scalaires d'éléments de l'espace de classification. Ainsi, si l'on souhaite appliquer la transformation Φ comme pré-traitement pour enrichir la collection de descripteurs, il nous suffit d'évaluer la fonction k suivante :

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle = \left(x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 x_2 y_2\right) \tag{3.21}$$

$$= \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2 . \tag{3.22}$$

On peut montrer [212] que ce résultat se généralise pour tout ordre δ : si Φ est la transformation associant à un exemple $\boldsymbol{x}$ l'ensemble des monômes à l'ordre δ, alors :

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^\delta .$$

Ce résultat est particulièrement intéressant puisqu'il montre qu'il est possible d'appliquer la transformation Φ vers un espace de dimension supérieure sans calculer explicitement la fonction Φ. Celle-ci se trouve exprimée implicitement au travers de la fonction k, dont l'expression est beaucoup plus simple que celle de Φ. En effet, la dimension de l'espace image de ϕ peut très facilement excéder les capacités computationnelles d'une machine puisque l'on dénombre, pour des exemples de dimension d, N_δ monômes d'ordre δ :

$$N_\delta = \left(\begin{array}{c} \delta + d - 1 \\ \delta \end{array} \right) = \frac{(\delta + d - 1)!}{\delta!(d-1)!} .$$

Soit, par exemple, pour des données regroupant $d = 100$ composantes, un espace des monômes d'ordre $\delta = 5$ de dimension proche de 10^7.

La structure de cet espace se trouve cependant synthétisée par l'expression du produit scalaire dans la fonction k. La Théorie des Noyaux Reproduisants, introduite dans le paragraphe suivant, formalise ce résultat et l'étend à d'autres types de transformations.

3.5.2 Théorie des Noyaux Reproduisants

La Théorie des Noyaux Reproduisants a été introduite par Azonszajn en 1950 [15]. Elle formalise la relation entre la fonction k introduite précédemment et la transformation Φ, par le biais de concepts d'analyse fonctionnelle sur les espaces de Hilbert. On considère que les données évoluent dans un espace $\mathcal{X}$ quelconque.

Définition : Soit une fonction $k : \mathcal{X}^2 \rightarrow \mathbb{R}$. k est un *noyau semi-défini positif* si

$$\int_{\mathcal{X}} k(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{x}) g(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \geq 0 \qquad \forall g \in \mathcal{C}(\mathcal{X}).$$

Il est possible d'utiliser la définition équivalente ci-dessous, généralement plus exploitable, basée sur les échantillonnages possibles de $\mathcal{X}$.

Définition (alternative) : $k : \mathcal{X}^2 \rightarrow \mathbb{R}$ est un *noyau semi-défini positif* si

$$\sum_{i,j=1}^{n} c_i c_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \qquad \forall \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \qquad \forall c_1, \ldots, c_n \in \mathbb{R}.$$

Si l'on définit la *matrice de Gram* $\boldsymbol{K}$ d'un noyau k par rapport aux éléments $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ comme $[\boldsymbol{K}]_{ij} = k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, alors cette définition est équivalente à la positivité semi-définie (au sens matriciel) de $\boldsymbol{K}$ pour tout ensemble $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.

Les résultats d'algèbre matricielle nous permettent donc d'inférer les propriétés de *symétrie* du noyau

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i),$$

et de *positivité de la diagonale*

$$k(\boldsymbol{x}, \boldsymbol{x}) \geq 0 \qquad \forall \boldsymbol{x} \in \mathcal{X}.$$

Introduisons maintenant la transformation Φ suivante, de l'espace $\mathcal{X}$ vers l'espace fonctionnel $\mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbb{R}\}$:

$$\begin{array}{rccc} \Phi : \mathcal{X} & \to & \mathbb{R}^{\mathcal{X}} \\ \boldsymbol{x} & \mapsto & k(.\,, \boldsymbol{x}). \end{array} \tag{3.23}$$

On peut montrer que pour un ensemble arbitraire d'éléments $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, l'ensemble $\mathcal{F}$ de fonctions a une structure d'espace vectoriel :

$$\mathcal{F} = \left\{ f, \quad f = \sum_{i=1}^{n} \alpha_i \Phi(\boldsymbol{x}_i) \right\}.$$

Soit deux fonctions de cet espace $f = \sum_{i=1}^{n} \alpha_i \Phi(\boldsymbol{x}_i)$ et $g = \sum_{j=1}^{n} \beta_j \Phi(\boldsymbol{x}_j)$, on montre en outre [212] que l'opérateur défini par

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

est un produit scalaire dans l'espace $\mathcal{F}$. De plus on remarque que

$$\langle f, g \rangle = \sum_{j=1}^{n} \beta_i f(\boldsymbol{x}_j) = \sum_{i=1}^{n} \alpha_i g(\boldsymbol{x}_i). \tag{3.24}$$

On peut également exprimer la fonction noyau sur tout exemple $\boldsymbol{x}_i$ comme élément de l'espace $\mathcal{F}$: $k(.\,, \boldsymbol{x}_i) = \sum_{j=1}^{n} \alpha_j \Phi(\boldsymbol{x}_i)$, avec $\alpha_j = \delta_{ij}$ (où δ_{ij} est le symbole de Kronecker). La propriété 3.24 est particulièrement intéressante puisqu'elle implique les deux relations suivantes sur la fonction noyau :

$$\langle k(.\,, \boldsymbol{x}_i), f \rangle = f(\boldsymbol{x}_i)$$

$$\langle k(.\,, \boldsymbol{x}_i), k(.\,, \boldsymbol{x}_j) \rangle = k(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

Cette dernière observation justifie le nom de *noyaux reproduisants* donné aux noyaux définis positifs, puisque ces derniers présentent la particularité de pouvoir s'exprimer comme un produit scalaire dans un espace fonctionnel en bijection avec l'espace $\mathcal{X}$. On retrouve finalement, en remplaçant les $k(.\,, \boldsymbol{x})$ par la transformation Φ introduite, la relation qui nous avait dans un premier temps servi à introduire la fonction noyau (équation 3.21), formalisée dans le théorème ci-dessous.

Le **théorème de Mercer** [156] affirme que tout noyau défini positif peut s'exprimer comme un produit scalaire sur l'espace image d'une fonction de transformation Φ :

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle .$$

Un noyau défini positif est d'ailleurs généralement décrit comme vérifiant la *condition de Mercer*. Le terme noyau désignera implicitement par la suite un noyau semi-défini positif.

On remarque que le théorème de Mercer stipule l'existence d'une fonction Φ mais n'apporte pas de moyen de la construire analytiquement. Il n'existe pas en fait de correspondance bijective entre le noyau k et la transformation Φ, parce que l'expression de cette dernière dépend de l'espace dans lequel on décrit le noyau reproduisant. Schölkopf et Smola présentent d'ailleurs une manière alternative de construire la fonction Φ à partir du noyau [212]. Il n'est pas rare, en outre, que Φ ne soit pas exprimable analytiquement (on parle alors de fonction *implicite*), comme dans le cas

d'espaces transformés de dimension infinie, ce qui vaut pour le noyau Gaussien RBF (que nous présentons dans la section suivante).

L'exploitation de noyaux fut introduite pour la première fois dans le domaine de l'apprentissage statistique en 1964 par Aizerman et al. [8], qui en présentent en outre l'interprétation géométrique. L'usage de la fonction noyau prend généralement le nom de *kernel trick* (littéralement « l'astuce du noyau ») puisqu'il permet, avec simplicité, d'introduire la non-linéarité dans un algorithme exclusivement basé sur des produits scalaires, c'est-à-dire invariant en rotation [212]. Bien qu'il leur soit antérieur de plusieurs décennies, le *kernel trick* ne fut réellement compris et largement utilisé qu'à partir du début des années 90 avec l'apparition des Machines à Vecteurs de Support [208].

3.5.3 Noyaux d'usage courant

Il est généralement difficile de vérifier analytiquement la condition de Mercer. Néanmoins, un certain nombre de noyaux sont connus comme étant définis positifs et largement exploités par la communauté :

- **Linéaire :** $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$
  celui-ci correspond au produit scalaire sans transformation. Il traduit donc la forme traditionnelle des algorithmes, sans l'usage du *kernel trick*.
- **Polynômial homogène :** $k(\boldsymbol{x}, \boldsymbol{y}) = \left(\boldsymbol{x}^T \boldsymbol{y}\right)^\delta$
  présenté dans la section 3.5.1. Celui-ci permet indirectement d'appliquer le principe de maximisation de la marge aux classifieurs polynômiaux.
- **Polynômial inhomogène :** $k(\boldsymbol{x}, \boldsymbol{y}) = \left(1 + \frac{c}{d} \boldsymbol{x}^T \boldsymbol{y}\right)^\delta$
  L'ajout d'une constante au produit scalaire permet d'inclure dans la transformation Φ tous les monômes d'ordre inférieur ou égal à δ. Le noyau polynômial inhomogène implique donc un espace transformé de dimension supérieure au noyau homogène.
- **Gaussien RBF :** $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{y}||^2}{d\,\sigma^2}\right)$
  Les fonctions à base radiale (RBF *Radial Basis Functions*) sont définies par le fait qu'elles ne dépendent que de la distance entre leurs arguments : $\phi(\boldsymbol{x}, \boldsymbol{y}) = \phi(||\boldsymbol{x} - \boldsymbol{y}||)$.
  Le noyau Gaussien RBF applique ainsi une gaussienne sur la distance entre les exemples. On peut montrer que l'espace transformé dans ce cas est de dimension infinie, puisque les exemples d'une collection arbitrairement grande y sont linéairement indépendants. Le caractère radial présente en outre la particularité de placer tous les exemples sur la sphère unité dans l'espace transformé ($||\Phi(\boldsymbol{x})||^2 = k(\boldsymbol{x}, \boldsymbol{x}) = \exp(0) = 1 \quad \forall \boldsymbol{x}$).
- **Sigmoïdal :** $k(\boldsymbol{x}, \boldsymbol{y}) = \tanh\left(\frac{c}{d}\, \boldsymbol{x}^T \boldsymbol{y} + \theta\right)$
  La fonction de décision construite avec un noyau sigmoïdal est égale à celle d'un réseau de neurones à deux couches [187][39][236]. Le noyau sigmoïdal ne respecte pas la condition de Mercer pour toutes les valeurs de c et θ. Il demeure cependant couramment utilisé et reste généralement exploitable, malgré sa possible inadéquation théorique.

On remarquera que contrairement à la plupart des publications de la littérature, nous avons introduit dans les 3 derniers noyaux un facteur de normalisation dépendant de la dimension d des exemples. Celle-ci, suggérée par Schölkopf et al. [208], permet de compenser l'influence de la dimension sur les paramètres du noyau (σ, c, ou encore θ ; la détermination de ces paramètres sera abordée dans la section 4.2).

Il est également possible de construire des noyaux à partir de noyaux existants. On peut en effet montrer [212][230] que :

- Toute combinaison linéaire positive de noyaux $(k_i)_{i=1,\ldots,N}$ est un noyau :
  $k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{N} \alpha_i k_i(\boldsymbol{x}, \boldsymbol{y}) \qquad \alpha_i > 0 \; \forall i$
- Tout produit de noyaux $(k_i)_{i=1,\ldots,N}$ est un noyau :
  $k(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{N} k_i(\boldsymbol{x}, \boldsymbol{y})$

3.5.4 Machines à Vecteurs de Support non-linéaires

L'introduction du *kernel trick* laisse le problème d'optimisation dual inchangé (voir équation 3.15) :

$$\begin{array}{rl} \text{maximiser} & L_D(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \mathbf{1} - \dfrac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H} \boldsymbol{\alpha} \\ \text{sous les contraintes} & \alpha_i \geq 0, \quad i = 1, \ldots, n \\ \text{et} & \boldsymbol{\alpha}^T \boldsymbol{y} = 0. \end{array} \tag{3.25}$$

Seule la matrice $\boldsymbol{H}$ est modifiée pour substituer aux produits scalaires la fonction noyau : $[\boldsymbol{H}]_{ij} = y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. L'usage du noyau n'introduit donc aucune complexification de la méthode de résolution du problème. La fonction de décision prend la forme suivante :

$$f(\boldsymbol{x}) = \sum_{i \in \mathcal{S}_{SV}} \alpha_i \, y_i \, k(\boldsymbol{x}_i, \boldsymbol{x}). \tag{3.26}$$

L'adjonction du noyau apporte une grande souplesse aux Machines à Vecteurs de Support. La transformation implicite dans un espace à haute dimension élargit considérablement le champ des surfaces de séparation applicables tout en maintenant un contrôle sur le risque structurel (les considérations sur la dimension VC d'un hyperplan présentée plus haut s'appliquant également dans l'espace transformé).

Toutefois, le formalisme présenté jusqu'ici suppose la séparabilité des données, nécessaire pour définir la marge M (équation 3.4), qui permet de déduire le problème d'optimisation sur le vecteur normal $\boldsymbol{w}$. De plus, la présence éventuelle d'un exemple mal étiqueté a un impact considérable sur l'hyperplan de séparation. Nous présentons dans la section suivante le concept des hyperplans à marge souple (*soft margin*) qui permettent de s'affranchir de la contrainte de séparabilité.

3.6 Machines à Marge souple

Afin de relâcher les contraintes du problème (équation 3.6), Vapnik et Cortes introduisent [54] les *variables d'écart* positives $\xi_i \geq 0$, introduites par Smith en 1968 [219] et reprises par Bennett et Mangasarian [23] pour la résolution du problème de séparation par programmation linéaire. Les contraintes *relâchées* deviennent donc

$$\begin{array}{l} y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b \right) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{array} \qquad i = 1, \ldots, n. \tag{3.27}$$

Un exemple est donc mal classifié si $\xi_i > 1$, puisqu'il se situe alors du mauvais côté de l'hyperplan de séparation. Ainsi, pour des valeurs arbitraitement grandes de ξ_i, les contraintes peuvent être toujours respectées. Cependant, afin de minimiser l'erreur de classification, il convient d'ajouter à la fonction objectif une pénalité sur les variables d'écart dans le problème :

$$\min \frac{1}{2} ||\boldsymbol{w}||^2 + C \left(\sum_{i=1}^{n} \xi_i \right)^k, \tag{3.28}$$

où l'on introduit le *facteur d'erreur* $C > 0$ ajustant le compromis entre les deux critères. Le choix de l'exposant k implique diverses formes de pénalisation. Par exemple $k = 0$ implique une pénalité basée sur le nombre d'exemples hors de la marge. Seuls les cas $k = 1$ et $k = 2$ évitent de rendre le problème NP-complet [54] en conservant une structure de programme quadratique. On peut montrer que dans le cas $k = 2$ ($L2$ SVM), le problème dual prend la forme :

$$\begin{array}{ll} \max\limits_{\boldsymbol{\alpha} \in \mathbb{R}^n} & \boldsymbol{\alpha}^T \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T \left(\boldsymbol{H} + \frac{1}{C}\boldsymbol{I} \right) \boldsymbol{\alpha} \\ \text{avec} & \boldsymbol{\alpha}^T \boldsymbol{y} = 0 \;\text{ et }\; 0 \leq \alpha_i \qquad i = 1, \ldots, n \end{array} \qquad L2 \text{ SVM}, \tag{3.29}$$

ce qui revient à appliquer le même algorithme que dans le cas séparable en ajoutant la constante $\frac{1}{C}$ aux termes diagonaux de la matrice de Gram du problème.

Cependant, on précédent généralement les $L1$ SVM ($k = 1$) qui conservent la forme originale du Lagrangien en imposant une nouvelle contrainte :

$$\begin{array}{ll} \max\limits_{\boldsymbol{\alpha}\in\mathbb{R}^n} & \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha} \\ \text{avec} & \boldsymbol{\alpha}^T\boldsymbol{y} = 0 \;\text{ et }\; 0 \leq \alpha_i \leq C \qquad i = 1, \ldots, n \end{array} \qquad L1 \text{ SVM.} \tag{3.30}$$

Dans ce cas, la variable C vient borner les multiplicateurs de Lagrange α_i. À nouveau, seuls les vecteurs de support ($\alpha_i > 0$) interviennent dans la fonction de décision, mais ceux-ci incluent désormais des exemples se situant hors de la marge ($\alpha_i = C$), correspondant aux valeurs non nulles des variables d'écart ($\xi_i > 0$) ; on parle alors de *marge souple*. Si la résolution du problème de maximisation s'en trouve modifiée, la solution demeure identique au cas séparable (équation 3.12) :

$$\boldsymbol{w} \;=\; \sum_{i\in\mathcal{S}_{SV}} \alpha_i y_i \Phi(\boldsymbol{x}_i). \tag{3.31}$$

Le choix de la constante C reste une question ouverte, cette dernière n'offrant malheureusement pas d'interprétation intuitive. Nous traiterons ce point dans le chapitre suivant en section 4.5.

Les ν-SVM ont été proposés comme alternative [211], impliquant à la place de C une variable ν plus intuitive permettant de contrôler le nombre de vecteurs de support, mais elles ne seront pas abordés dans ce document.

3.7 Méthodes à noyaux

On peut résumer les Machines à Vecteurs de Support comme la conjonction des trois points suivants :

- **Le principe de Maximisation de la Marge**, qui est choisi pour respecter le paradigme de Minimisation du Risque Structurel, implique une « éparsification » du problème en ne faisant intervenir dans le processus de décision que les exemples (les Vecteurs de Support) les plus porteurs d'information.
- **La fonction noyau**, se substituant aux produits scalaires, permet de transformer implicitement les exemples dans un espace de dimension supérieure où la surface de décision linéaire se traduit dans l'espace d'entrée par une surface beaucoup plus complexe.
- **L'utilisation des variables d'écart** permet de relâcher les contraintes et d'autoriser la prise en compte d'exemples mal classifiés, supprimant toute contrainte sur la répartition des exemples d'apprentissage.

Le champ d'application de ces principes ne se limite pas à la classification supervisée. Ils peuvent en fait s'appliquer aux trois problèmes fondamentaux de l'apprentissage statistique, énoncés par Vapnik [236] :

- La **reconnaissance de formes**, par le biais des SVM.
- La **régression** pour l'estimation de fonction [234][236].
- L'**estimation de densité** [209]. On trouve également ce problème sous le nom de classification à une classe dans la littérature, souvent exploitée pour la détection de nouveauté [214]. Nous exploiterons cette dernière dans la section 9.5.

L'utilisation de noyaux a également permis de « kerneliser » d'autres algorithmes exclusivement formulés en termes de produits scalaires :

- L'**Analyse en Composantes Non-Linéaires** (ou *Kernel PCA*) [210]
- L'**Analyse Discriminante de Fisher Kernelisée** (*Kernel FDA*) [158][203][19], que nous aborderons dans la section 7.5.5.

On pourra consulter [163] et [22] pour une vue d'ensemble des algorithmes liés à l'usage des noyaux.

3.8 Une méthode universelle d'apprentissage

On voit, au regard des trois principes énoncés dans le paragraphe précédent, ce qui pousse Vapnik à qualifier les SVM de « méthode universelle d'apprentissage » [236]. En effet, loin d'être un énième algorithme de reconnaissance des formes, les SVM apportent la rigueur de l'approche statistique, par le biais de la minimisation de bornes sur le risque, à une multitudes d'algorithmes existants, dont le comportement est « simulé » par le choix de la fonction noyau.

Ainsi on a vu que le noyau polynômial constitue une implémentation implicite des classifieurs polynômiaux. On peut montrer également que le noyau gaussien RBF simule la classification par réseaux de fonctions RBF (on pourra trouver une comparaison des deux approches dans l'article de Schölkopf et al. [213]). Enfin, Le noyau sigmoïdal reprend la fonction de décision d'un réseau de neurones à deux couches [236]. Pour chacun de ces cas, le principe des SVM améliore l'approche originale en y adjoignant la mise en évidence d'un ensemble restreint d'exemples (les vecteurs de support) exprimant à eux seuls la complexité du problème [208]. Plusieurs auteurs ont d'ailleurs montré l'équivalence entre la résolution par SVM linéaire et l'Analyse Discriminante Linéaire opérée sur le sous-ensemble des vecteurs de support [216][98][123].

De plus, le procédé d'optimisation permet de s'affranchir des affinages empiriques, souvent employés dans les techniques traditionnelles. Ainsi le choix de la structure des réseaux de neurones (nombre de couches et de neurones) reste l'une des carences principale de cette approche, tandis qu'elle se trouve implicitement déterminée lors de la phase d'apprentissage par SVM avec noyau sigmoïdal ; de même pour la détermination des centres dans la classification par fonctions RBF (généralement évalués par des algorithmes de clustering, type K-means).

Enfin, il est important de souligner que le problème de maximisation posé par l'équation 3.30 est convexe et implique donc la convergence vers un maximum global unique [40], contrairement à beaucoup d'autres approches, comme les réseaux de neurones, qui ne garantissent que la convergence vers un optimum local.

Toutefois, les SVM présentent deux problèmes pour leur mise en application :

1. Nous avons vu que les noyaux apportent une souplesse considérable au processus d'apprentissage. Néanmoins la question du choix d'un noyau optimal est fondamentale et loin d'être évidente ; de plus, si le champ des paramètres est considérablement restreint chaque noyau comporte généralement une ou deux variables qu'il faut déterminer. Ces questions sont abordées dans le chapitre 4.
2. Dans leur construction, les SVM sont une méthode discriminative. Il nous faut donc déterminer une stratégie permettant d'étendre leur champ d'application aux cas multi-classes. Ce point est traité dans le chapitre 5.