

## *Chapitre 2*

### *Les Machines à Vecteurs de Support*

## INTRODUCTION

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains paramètres descriptifs. Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples. Un exemple consiste en la description d'un cas avec la classification correspondante. Un système d'apprentissage doit alors, à partir de cet ensemble d'exemples, extraire une procédure de classification, il s'agit en effet d'extraire une règle générale à partir des données observées. La procédure générée devra classer correctement les exemples de l'échantillon et avoir un bon pouvoir prédictif pour classer correctement de nouvelles descriptions. Les méthodes utilisées pour la classification sont nombreuses, citons : la méthode des séparateurs à vastes Marges (SVM) (ou : les machines à vecteurs de support). Nous présentons une étude sur cette technique utilisée dans notre travail. Cette méthode a été montrée leur efficacité dans de nombreux domaines d'applications tels que le traitement des eaux propres, et leurs classifications correspondante.

### 1. Etat de l'art

L'origine des Machines à vecteurs de support (SVM) remonte à 1975 lorsque Vapnik et Chervonenkis proposèrent le principe du risque structurel et la dimension VC pour caractériser la capacité d'une machine d'apprentissage. A cette époque, ce principe n'a pas trouvé place et il n'existait pas encore un modèle de classification solidement appréhendé pour être utilisable. Il a fallu attendre jusqu'à l'an 1982 pour que Vapnik propose un premier classificateur basé sur la minimisation du risque structurel baptisé SVM [11]. Ce modèle était toutefois linéaire et l'on ne connaissait pas encore le moyen d'induire des frontières de décision non linéaires. En 1992, Boser et al. proposent d'introduire des noyaux non-linéaires pour étendre le SVM au cas non-linéaire [12]. En 1995, Cortes et al. proposent une version régularisée du SVM qui tolère les erreurs d'apprentissage tout en les pénalisant [13]. Les SVMs (le pluriel est utilisé pour désigner les différentes variantes du SVM) n'ont cessé de susciter l'intérêt de plusieurs communautés de chercheurs de différents domaines d'expertise. Par exemple, Cortes et al. dans [13], Scholkopf et al. [15], et Burges et al. [14] ont appliqué les SVM à la reconnaissance de chiffres manuscrits isolés, Blanz et al. [31] ont expérimenté le SVM sur des objets 2D de vues différentes. Schmidt et al. [16] ont exploré la tâche de reconnaissance d'orateur. Osuna et al. ont traité de la reconnaissance d'images de visages [17]. Dans la plupart des cas, la performance du SVM égale ou dépasse celle de modèles

classiques déjà établis. L'estimation de densité de probabilité [18] et la décomposition ANOVA [32] ont été aussi explorées. D'autres auteurs ont aussi étudié l'apport de la connaissance a priori pour ce type de classificateurs. Ainsi, le SVM virtuel de Burges et al. améliorent la généralisation en appliquant un bruit spécifique à l'ensemble de vecteurs de support [14]. Smola et al. ainsi que Wahba ont mis en évidence la ressemblance entre le SVM et la théorie de régularisation [19, 20, 21]. Ils ont démontré qu'associer un noyau particulier à un SVM revient à considérer une pénalisation différente de l'erreur d'apprentissage en maximisant la marge. Ce qui nous permet de dire que la maximisation de la marge dans l'espace augmenté est une forme de régularisation de l'apprentissage. Dès lors, le SVM permet de répondre à deux problèmes centraux de la théorie de l'apprentissage statistique que sont :

- le contrôle de la capacité du classifieur,
- le sur-apprentissage des données.

Nous définissons les hyper-plans de séparation et éclaircissons le lien entre la notion de capacité et la marge de séparation entre deux classes. La notion de marge optimale est par la suite abordée davantage. La dérivation de l'algorithme du SVM est ensuite établie en deux étapes. L'algorithme est adapté au moyen d'une heuristique particulière au cas de données non séparables. Enfin, le cas non-linéaire est toute la théorie de cette technique est traité dans les paragraphes suivante.

## 2. Risque structurel

Dans cette section, nous rappelons quelques éléments essentiels de la théorie de l'apprentissage statistique. Nous introduisons par la même occasion le principe du risque structurel que minimise le SVM [22].

Considérons un problème de classification à deux classes et soit les paires de données étiquetées :

$$(x_i, y_i), \dots, (x_l, y_l) \in R^N \times \{+1, -1\} \quad (2.1)$$

où  $x_i$  représente la  $i^{\text{eme}}$  observation de l'ensemble d'apprentissage et  $y_i$  son étiquette. et soit l'ensemble de fonctions  $f_\alpha$  défini :

$$\{f_\alpha: \alpha \in \Lambda\}, f_\alpha: R^N \rightarrow \{+1, -1\} \quad (2.2)$$

Supposons aussi qu'il existe une distribution  $P(x, y)$  des données dont on ignore le modèle. La tâche d'apprentissage de  $f_\alpha$  qui approxime au mieux cette distribution consiste à minimiser le risque réel donné par :

$$R(\alpha) = \int \frac{1}{2} |f_\alpha(x) - y| dP(x, y) \quad (2.3)$$

Ne connaissant pas  $P(x, y)$ , il est difficile d'estimer le risque  $R(\alpha)$ . Il est possible toutefois de considérer une fonction de risque empirique de la forme :

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^l \frac{1}{2} |f_\alpha(x_i) - y_i| \quad (2.4)$$

Si l'ensemble d'apprentissage est fini, la minimisation du risque empirique donné par l'équation 2.4 ne garantit pas un minimum pour le risque réel.

En 1979, Vapnik [23] a mis au point le principe de la minimisation du risque structurel qui évite le sur-apprentissage des données à la convergence de la procédure d'apprentissage. Le risque empirique, par ailleurs, représente une estimation assez optimiste du risque réel dont la minimisation ne garantit pas la convergence vers une solution acceptable.

En fait, si nous choisissons d'entraîner un PMC avec une méthode de descente de gradient quelconque, nous minimiserons l'erreur quadratique sur l'ensemble d'apprentissage en considérant un grand nombre d'époques à travers les données. Or, le modèle trouvé ainsi représente une solution biaisée à cause de l'erreur de variance dont souffre l'erreur quadratique. En effet, il est démontré qu'un minimum pour l'erreur de généralisation requiert un compromis entre l'erreur de biais et l'erreur de variance. La technique du 'early stopping' par exemple, est une des règles empiriques utilisées pour éviter cet inconvénient.

Le risque structurel constitue une borne supérieure de l'erreur de généralisation qui s'écrit :

$$R(\alpha) \leq R_{emp}(\alpha) + \phi\left(\frac{h}{l}, \frac{\log(\eta)}{l}\right) \quad (2.5)$$

pour  $\alpha \in \Lambda$  et  $l \geq h$  avec une probabilité d'au moins  $1 - \eta$ , et où  $\phi$  est un terme de confiance donnée par :

$$\phi\left(\frac{h}{l}, \frac{\log(\eta)}{l}\right) = \sqrt{\frac{h(\log\left(\frac{2l}{h} + 1\right) - \log\left(\frac{\eta}{4}\right))}{l}} \quad (2.6)$$

Le paramètre  $h$  est appelé V C ou dimension de Vapnik-Chervonenkis. Il décrit la capacité de l'ensemble de fonctions solutions. Pour un problème de classification binaire,  $h$  est le nombre maximum de points séparables selon  $2^h$  configurations par l'ensemble de

fonctions solutions. La minimisation de cette borne consiste à opérer une régularisation de l'erreur empirique  $R_{emp}(\alpha)$  via le terme de pénalité  $\phi\left(\frac{h}{l}, \frac{\log(\eta)}{l}\right)$ . Donc un minimum est garanti en minimisant à la fois l'erreur empirique et le terme de régularisation.

Le terme *machine d'apprentissage* est utilisé pour désigner l'ensemble de fonctions de décision  $f_\alpha$  dont la machine dispose, le principe d'induction afin d'approximer l'erreur de généralisation et l'algorithme mettant en oeuvre le principe d'induction.

La limite supérieure du risque réel (erreur de généralisation) définie dans l'équation 2.5 constitue un principe essentiel de la théorie des Machines à Vecteurs de Support qui nécessite encore quelques remarques pertinentes.

Selon l'équation 2.5, étant donné un ensemble d'apprentissage de taille  $l$ , il est possible de borner  $R(\alpha)$  en minimisant la somme des quantités :  $R_{emp}(\alpha)$  et  $h(\{f_\alpha : \alpha \in \Lambda'\})$ , tel que  $\Lambda'$  représente un sous-ensemble de  $\Lambda$ . Par ailleurs, l'erreur empirique dépend de la solution particulière trouvée par la machine d'apprentissage, à savoir  $f_\alpha$ , et peut être réduite en choisissant des valeurs appropriées pour les paramètres  $\alpha_i$  composant  $\alpha$ .

La capacité  $h$  (dimension VC) quant à elle, dépend de l'ensemble de fonctions  $\{f_\alpha : \alpha \in \Lambda'\}$ , que la machine d'apprentissage peut inférer. Dans le but de contrôler  $h$ , il est possible de considérer des structures  $S_n := \{f_\alpha : \alpha \in \Lambda_n\}$ , de plus en plus complexes, telles que :

$$S_1 \subset S_2 \subset \dots \subset S_n \subset \dots \quad (2.7)$$

et dont les capacités respectives vérifient l'inégalité :

$$h_1 \leq h_2 \leq \dots \leq h_n \leq \dots \quad (2.8)$$

Pour un ensemble observations  $(x_1, y_1), \dots, (x_l, y_l)$ , la minimisation du risque structurel vise à choisir  $f_\alpha$  parmi l'ensemble de fonctions  $\{f_\alpha : \alpha \in \Lambda_n\}$  possibles, qui minimise le risque garanti (le risque garanti a été utilisé par Guyon et Boser pour désigner la limite supérieure de l'erreur de généralisation donnée dans l'équation 2.5).

Le processus de choisir le bon sous-ensemble de fonctions solutions revient à contrôler la complexité du classifieur en cherchant le meilleur compromis entre une faible erreur empirique et une complexité moindre. Notons enfin, que plusieurs travaux ont abouti à des

résultats similaires quant à la nécessité d'un compromis entre l'erreur d'apprentissage et la complexité du modèle pour assurer une faible erreur de généralisation. Il en est ainsi avec la théorie de la régularisation [20], le '*Minimum Description Length*' [25, 26] et le dilemme Biais-Variance [33].

### 3. Espace augmenté

Soit l'observation  $x = (x_1, \dots, x_N) \in R^N$ , et supposons que son information utile soit contenue dans l'ensemble des monômes d'ordre  $d$  des composantes  $x_j$  de  $x$ . Dans ce cas, il est possible de considérer l'espace  $F$  des monômes d'ordre  $d$  comme espace de caractérisation. En reconnaissance d'images, ceci équivaut à calculer des produits de valeurs de pixels. Par exemple, dans l'espace d'entrée  $R^2$ , l'ensemble des monômes d'ordre 2 constitue un vecteur de caractérisation de dimension 3, comme montré ci-dessous :

$$\Phi : R^2 \rightarrow F = R^3, \quad (2.9)$$

qui associe pour chaque observation  $x = (x_1, x_2)$  une image de la forme :

$$(x_1, x_2) \rightarrow (x_1^2, x_2^2, x_1 x_2)$$

Cette approche permet de considérer un degré de non-linéarité allant jusqu'à :

$$N_F = \frac{(N + d - 1)!}{d! (N - 1)!} \quad (2.10)$$

où  $N_F$  représente la dimension de l'espace des monômes de degré  $d$  correspondant à l'espace d'entrée  $R^N$

#### 3.1. Noyau polynomial

Pour pouvoir représenter des produits scalaires de la forme  $\Phi(x) \cdot \Phi(y)$  (voir équation 3.9), nous utiliserons la notation :

$$k(x, y) = \Phi(x) \cdot \Phi(y) \quad (2.11)$$

qui décrit la valeur du produit scalaire dans l'espace augmenté  $F$ .

Ce calcul ne nécessite pas le calcul des images  $\Phi(x)$  et  $\Phi(y)$  explicitement. Cette propriété a été utilisée par Boser, Guyon et Vapnik pour étendre l'algorithme de l'hyper-plan généralisé au SVM non-linéaire [12]. L'espace augmenté  $F$  est alors appelé espace de linéarisation Figure 2.1.

En réalité, Aizerman et al. étaient les premiers à avoir élucidé le concept de manipulation de produits scalaires via un noyau non-linéaire  $k(x, y)$  [27].

Tout noyau décomposable selon la forme donnée dans l'équation 2.11 est un noyau Mercer [28]. On démontre que tout noyau continu, symétrique et semi-défini positif est un noyau de Mercer.

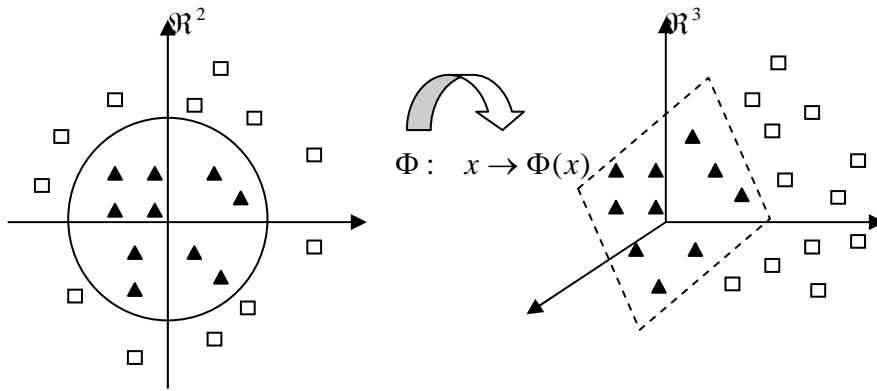


Figure 2.1 Illustration de l'effet de changement d'espace par une fonction noyau.

Le tableau 2.1 montre quelques noyaux de Mercer classiques utilisés pour le SVM.

TABLEAU 2.1 : Quelques noyaux classiques.

Noyau	Formule
Linéaire	$k(x, y) = x \cdot y + c$
Sigmoïdes	$k(x, y) = \tanh((a \cdot x \cdot y) + b)$
Polynomiale	$k(x, y) = (a \cdot x \cdot y + b)^d$
RBF	$k(x, y) = \exp(-\ x - y\ ^2 / 2\sigma^2)$
Laplace	$k(x, y) = \exp(- \gamma  \ x - y\ )$

#### 4. Formulation de SVM

Dans ce paragraphe, nous décrivons l'utilisation des machines à vecteurs de support pour la classification et dérivons la formulation du SVM linéaire. Nous y trouverons deux des-criptions distinctes. La première traite le cas de données séparables. Une version modifiée permet, par ailleurs, de considérer des données non séparables. L'extension au cas non-linéaire est décrite plus loin.

Soit donc le problème de classification binaire défini par  $l$  observations  $(x_1, y_1), \dots, (x_l, y_l)$ , tirées de manière indépendante d'une même distribution inconnue. Chacune des données  $x_i$ ,  $\forall i = 1, \dots, l$ , représente un vecteur de caractéristiques dans  $R^N$  de dimension  $N$ .

Les Variables  $y_i = \{+1, -1\} \forall i = 1, \dots, l$  représentent les classes d'appartenance correspondant aux données  $x_i$ .

Il s'agit d'estimer la fonction de décision  $f(x)$  qui approxime au mieux les exemples d'apprentissage, telle que  $y_i = f(x_i)$  pour toute donnée  $x_i$ . Aussi, définissons la fonction de coût associée à l'erreur de classification comme suit :

$$E(y, f(x)) = \begin{cases} 0 & \text{si } y = f(x) \\ 1 & \text{sinon} \end{cases} \quad (2.12)$$

#### 4.1. Hyper-plans de séparation

Nous avons énoncé dans l'équation 2.7 que des structures imbriquées  $S_n$  d'espaces de fonctions mettent en oeuvre le principe de la minimisation du risque structurel si un compromis entre la complexité du modèle et une faible erreur d'apprentissage est trouvé. L'algorithme de construction du SVM considère des hyper-plans de séparation entre les classes à discriminer. Soit alors  $z_1, \dots, z_l \in F$  les images de  $x_1, \dots, x_l \in R^N$  dans l'espace augmenté  $F$ . L'hyper-plan de séparation dans l'espace  $F$  peut s'écrire :

$$\{z \in F : (w_0 \cdot z) + b_0 = 0\}, \quad (2.13)$$

Où  $(w, b) \in F \times R$ .

Cette formulation permet toujours de choisir des surfaces dont le couple  $(w, b)$  est un multiple de  $(w_0, b_0)$  de l'équation 2.13. Cependant, il est d'usage de considérer une formulation canonique telle que :

$$\min_{i=1, \dots, l} |(w \cdot z_i) + b| = 1 \quad (2.14)$$

qui garantit que les points les plus proches de la surface de séparation aient une sortie de valeur absolue égale à 1. Il est facile de démontrer, dans ce cas, que la distance séparant les exemples de la classe positive et les exemples de la classe négative les plus proches du plan de séparation (appelée aussi marge de séparation) est égale à :

$$D = \frac{2}{\|w\|} \quad (2.15)$$

Le principe de minimisation du risque structurel dans la théorie des machines à vecteurs de support découle du fait qu'il est démontré que la dimension VC associée à des fonctions de décision de la forme :

$$f_{w,b} = \text{sgn}((w \cdot z) + b) \quad (2.16)$$

vérifie :

$$h < R^2 A^2 + 1 \quad (2.17)$$

Dans l'équation 2.19,  $R$  est le rayon de la plus petite hyper-sphère englobant les données d'apprentissage  $z_1, \dots, z_l$  et  $A$  un scalaire tel que  $A \geq 2/D$ . Si  $\|w\| > A$ , la borne supérieure de la dimension VC devient  $N_F + 1$ , où  $N_F$  est la dimension effective de l'espace augmenté  $F$ .

Il est établi que pour  $\|w\| \leq A$ , il est possible d'obtenir des dimensions VC largement inférieures à  $N_F$ . Cette propriété permet de travailler dans des espace de très haute dimension avec une dimension VC minimale. En résumé, maximiser la limite inférieure de la marge  $D$  permet de minimiser la dimension VC de la machine d'apprentissage. Bref, nous cherchons à minimiser à la fois le nombre d'exemples d'apprentissage mal classifiés et la dimension VC en maximisant la marge.

#### 4.2. Hyper-plans à marge optimale

Supposons que nous ayons un ensemble d'observations  $(z_1, y_1), \dots, (z_l, y_l)$  tel que  $z_i \in F$  et  $y_i \in \{+1, -1\}$ .

Idéalement, nous cherchons à trouver la fonction de décision

$$f_{w,b} = \text{sgn}((w \cdot z) + b)$$

telle que :

$$f_{w,b}(z_i) = y_i, i = 1, \dots, l \quad (2.18)$$

Si cette fonction existe (le cas non-séparable sera traité dans la prochaine section), l'inégalité suivante est vérifiée :

$$y_i((z_i \cdot w) + b) \geq 1, i = 1, \dots, l \quad (2.19)$$

Le lagrangien primaire s'écrit alors

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^l \alpha_i (y_i((z_i \cdot w) + b) - 1) \quad (2.20)$$

Les variables  $\alpha_i$  correspondent aux facteurs de Lagrange des contraintes définies dans l'équation 2.19. Au minimum de la fonction objective  $L(w, b, \alpha)$ , ses gradients par rapport à  $w$  et  $b$  deviennent :

$$\frac{\partial}{\partial b} L(w, b, \alpha) = 0, \quad (2.21)$$

et

$$\frac{\partial}{\partial w} L(w, b, \alpha) = 0, \quad (2.22)$$

Résoudre les équations 2.21 et 2.22 donne :

$$w = \sum_{i=1}^l \alpha_i y_i z_i \quad (2.23)$$

Avec

$$\alpha_i [(y_i((z_i \cdot w) + b) - 1)] = 0, i = 1, \dots, l \quad (2.24)$$

En remplaçant  $w$  par son expression de l'équation 2.23 dans l'équation 2.20, et en prenant en compte la contrainte de l'équation 2.24, la fonction objective  $L(w, b, \alpha)$  prend la forme d'un Lagrangien dual à maximiser fonction de  $\alpha_i$  seulement. Il s'écrit :

$$\begin{aligned} W(\alpha) \\ = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j}^l \alpha_i \alpha_j y_i y_j (z_i \cdot z_j) \end{aligned} \quad (2.25)$$

avec les contraintes :

$$\alpha_i \geq 0, i = 1, \dots, l \quad (2.26)$$

et

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (2.27)$$

La fonction de décision finale du SVM linéaire s'écrit :

$$f(z) = \text{sgn} \left( \sum_{i=1}^l \alpha_i y_i (z \cdot z_i) + b \right) \quad (2.28)$$

Les données d'apprentissages  $z_i$  associées à des paramètres  $\alpha_i$  non nuls sont appelées vecteurs de support. Pour une observation de test  $z$  quelconque, la valeur de  $f(z)$  indique sa classe d'appartenance inférée par le SVM.

### 4.3. Hyper-plans à marge molle

Le problème d'optimisation quadratique énoncé dans l'équation 2.25 a une solution dans le cas de données séparables uniquement. Dans le cas contraire, les conditions de Tucker (KKT) ne sont jamais satisfaites.

Cortes et al. [13] utilisent une technique qui consiste à accepter des erreurs d'apprentissage tout en les pénalisant. Les auteurs ont introduit un paramètre de pénalisation  $C$  qui règle le degré de compromis désiré entre la séparabilité des classes et l'étanchéité du modèle aux erreurs d'apprentissage. Pour ce faire, redéfinissons la contrainte de l'équation 3.19 en considérant des variables :

$$\xi_i \geq 0, i = 1, \dots, l$$

associées aux données  $z_i \in F, \forall i = 1, \dots, l$  de l'ensemble d'apprentissage.

La variable  $\xi_i$  est la distance séparant  $z_i$  de la frontière de la marge qui est définie :

$$\xi_i = [1 - f(z_i)y_i]_+ \text{ pour } i = 1, \dots, l$$

L'opérateur  $[\ ]_+$  est défini :

$$\begin{aligned} [x]_+ &= x \text{ si } x \geq 0 \\ [x]_+ &= 0 \text{ si } x < 0 \end{aligned}$$

La contrainte de l'équation 3.19 devient alors :

$$y_i((z_i \cdot w) + b) \geq 1 - \xi_i, i = 1, \dots, l \quad (2.29)$$

Dans ce cas, la fonction objective à optimiser s'écrit :

$$L(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i; \quad (2.30)$$

où  $C$  représente le facteur de Lagrangien associé au terme de pénalisation rajouté.

Cortes et Vapnik démontrent [9] que la fonction de décision dans ce cas est caractérisée par :

$$w = \sum_{i=1}^l \alpha_i y_i z_i \quad (2.31)$$

et que le Lagrangien dual à maximiser s'écrit :

$$W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (z_i \cdot z_j) \quad (2.32)$$

avec les contraintes :

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, l \quad (2.33)$$

et

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (2.34)$$

#### 4.4. Le SVM non-linéaire

Boser et al. [22], ont su produire des frontières de décision non-linéaire avec le SVM. L'idée est d'utiliser un noyau de Mercer qui permet de projeter les données dans un espace éventuellement plus grand dans lequel une séparation linéaire des classes est possible [32]. Il est alors important que la formulation du produit scalaire dans l'équation 2.25 reste intacte après l'introduction du noyau. Boser et al. démontrent qu'un noyau de Mercer ayant la forme :

$$\phi(x) \cdot \phi(y) = k(x, y) \quad (2.35)$$

ne change pas la nature de la fonction objective à optimiser qui équivaut toujours à un problème de maximisation quadratique.

Le Lagrangien dual de la fonction objective à maximiser est alors :

$$W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (2.36)$$

La fonction de décision s'écrit encore :

$$f(x) = \text{sgn} \left( \sum_{i=1}^l \alpha_i y_i k(x, x_i) + b \right) \quad (2.37)$$

Notons que  $f(x)$  est une combinaison linéaire de termes non-linéaires  $k(x, x_i)$  qui représente la similarité entre les images des points  $x$  et  $x_i$ . Pour une observation de test  $x$  quelconque, la valeur de  $f(x)$  indique la classe d'appartenance inférée par le SVM.

#### 4.5. Conditions de Karush-Kuhn-Tucker

La résolution de l'optimisation quadratique de l'équation 2.36 est basée sur les conditions de convergence dites de «Karush-Kuhn-Tucker» (KKT) qui établissent les conditions nécessaires (mais parfois suffisantes) de convergence de la fonction objective duale. Ces conditions sont relativement simples et s'écrivent :

$$\begin{aligned} \alpha_i = 0 &\Rightarrow y_i f(x_i) \geq 1 \text{ et } \xi_i = 0 \\ 0 < \alpha_i < C &\Rightarrow y_i f(x_i) = 1 \text{ et } \xi_i = 0 \\ \alpha_i = C &\Rightarrow y_i f(x_i) \leq 1 \text{ et } \xi_i \geq 0 \end{aligned} \quad (2.38)$$

Les équations 2.38 reflètent une propriété importante du SVM stipulant qu'une grande proportion des exemples d'apprentissage sont situés en dehors de la marge et ne sont pas retenus par le modèle. Par conséquent, leurs multiplicateurs  $\alpha_i$  sont nuls.

Les conditions de KKT traduisent le fait que seulement les variables  $\alpha_i$  des points situés sur la frontière de la marge ( $0 < \alpha_i < C$ ) ou à l'intérieure de celle-ci ( $\alpha_i = C$ ) sont non nulles. Ces points sont les vecteurs de support du classifieur.

Le SVM produit alors une solution clairsemée n'utilisant qu'un sous ensemble réduit des données d'apprentissage. Sans cette propriété, l'entraînement du SVM sur de gros ensembles de données ainsi que son stockage deviennent extrêmement prohibitifs.

Afin de trouver les paramètres du SVM, il est nécessaire de résoudre le problème d'optimisation quadratique convexe donné par l'équation 2.36 dont la formulation matricielle :

$$\begin{array}{ll} \text{Maximiser}_{\alpha_i} & W(\alpha) = \frac{1}{2} \alpha^T H \alpha - 1^T \alpha \\ \text{telque} & \begin{cases} y^T \alpha = 0 \\ 0 \leq \alpha_i \leq C \quad \forall i = 1, \dots, n \end{cases} \end{array} \quad (2.39)$$

où la matrice  $H$  telle que  $H_{ij} = y_i y_j k(x_i, x_j)$ . Cette matrice, appelée *Hessienne*, est donc très proche de la matrice de *Gram* et possède les propriétés de symétrie et de définition positive [35]

Pour résoudre le problème 2.39 qui est un problème d'optimisation quadratique, il est souvent amené en pratique à résoudre certaines équations par des méthodes numériques. Plusieurs techniques sont proposées qui abordent plusieurs aspects pratiques intéressants pour les SVM dans la reconnaissance de formes [35] :

- **Le gradient conjugué avec contraintes** : c'est un gradient conjugué classique, dont les directions sont projetées dans les sous-espaces définis par les contraintes  $\sum_{i=1}^n \alpha_i y_i = 0$ .
- **Des méthodes de projection** : également basées sur le gradient conjugué.
- **La décomposition de Bunch-Kaufman** : qui utilise la *Hessienne* tout en s'appuyant sur le fait que la plupart des  $\alpha_i$  sont nuls.

- **Les méthodes de points intérieurs** : comme l'algorithme de *Vanderbei* ; une méthode qui semble particulièrement intéressante lorsque les « vecteurs de support VS » sont nombreux par rapport à la taille de la base d'exemples.

#### 4.6. Calcul du biais $b$

Le paramètre de biais  $b$  permet de définir des surfaces de séparation ne passant pas par l'origine. Son calcul exploite les vecteurs de support respectant l'inégalité  $0 < \alpha_i < C$  dont les  $\xi_i$  correspondants sont nuls. L'égalité suivante

$$y_i \left( b + \sum_{j=1}^l y_j \alpha_j k(x_i, x_j) \right) = 1,$$

est alors vérifiée.

En considérant la moyenne calculée sur cet ensemble des vecteurs de support, une valeur stable de  $b$  peut s'écrire :

$$b = \frac{1}{\#sv} y_i \left( b + \sum_{j=1}^l y_j \alpha_j k(x_i, x_j) \right) = 1,$$

où  $\#sv$  représente le nombre de vecteurs de support considérés

### 5. Classification de données multiclasses

Le SVM est un classifieur binaire qui ne traite habituellement que des données appartenant à deux classes. Cependant, il existe des versions plus élaborées prenant en compte plus de deux classes simultanément au sein de la même fonction objective. Il n'est pas prouvé toutefois que celles-ci minimisent le risque structurel. La recherche demeure, cependant, très active sur ce sujet [29].

Nous cherchons à modéliser une fonction  $G : \Omega \rightarrow 1, \dots, K$ , qui définit  $K$  partitions dans l'espace des caractéristiques  $\Omega$ . Connaissant  $G$ , les partitions des classes sont définies par  $G^{-1}(c)$ , où  $c \in [1..K]$ .

Pour un problème à deux classes, l'hyper-plan  $(w, b)$  du SVM délimite les deux partitions selon  $\text{sign}f(x)$  où  $f(x) = w \cdot x + b$ . Par ailleurs, bien que le SVM soit un

classifieur binaire, il peut facilement être étendu pour décider de l'appartenance de données multiclassées. En particulier, on trouve deux schémas de classification :

1. *Un-contre-Tous*. Cette méthode est simple d'usage et donne des résultats raisonnables. Elle consiste à entraîner  $K$  SVM différents, séparant chaque classe des  $K - 1$  restantes. Ainsi, pour chaque exemple de test,  $K$  valeurs de sortie  $f_c(x)$  sont Disponibles. Une façon naïve mais simple de classification consiste à attribuer l'exemple à la sortie de plus grande amplitude.
2. *Un-contre-Un*. Cette méthode requiert l'apprentissage de  $\frac{1}{2}K(K - 1)$  classifieurs pour tous les couples de classes possibles. Durant le test, la méthode requiert la combinaison de toutes les sorties de classifieurs pour qu'une décision soit émise.

### 5.1. Approche Un-contre-Tous

L'idée de cette stratégie est de construire autant de classifieurs que de classes. Ainsi, durant l'apprentissage, tous les exemples appartenant à la classe considérée sont étiquetés positivement (+1) et tous les exemples n'appartenant pas à la classe sont étiquetés négativement (-1). A la fin de l'apprentissage, nous disposons de  $K$  modèles correspondant aux hyper-plans  $(W_i, b_i)$  tels que  $i = 1, \dots, K$ .

Durant le test, l'exemple est associé à la classe dont la sortie est positive selon la règle

$$x \in C_k \text{ si } w_i \cdot x + b_i > 0 \text{ pour } i = k$$

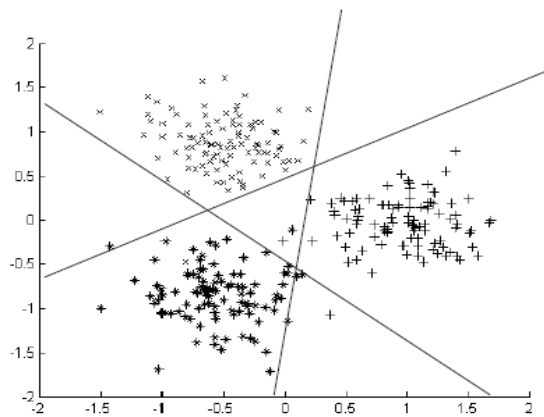


Figure 2.2 Problème à trois classes : frontières de décision linéaires dans la stratégie Un-contre-Tous

Or, il est possible que plusieurs sorties soient positives pour un exemple de test donné. Ceci

est particulièrement le cas des données ambiguës situées près des frontières de séparation des classes. On utilise dans ce cas un vote majoritaire pour attribuer l'exemple  $x$  à la classe  $C_k$  selon la règle de décision

$$C = \arg \max_i (W_i \cdot x + b_i)$$

Si dans beaucoup de cas, la règle énoncée ci-haut est suffisante, il se peut qu'elle faille lorsque les sorties des  $K$  classifieurs ne sont pas comparables. En effet, quelques chercheurs ont mis en évidence ce phénomène pour des données de difficulté moyenne ou grande. Il est facile d'expliquer ceci en considérant le SVM comme étant un perceptron opérant dans un espace défini par l'ensemble de vecteurs de support. D'ailleurs, une analogie intéressante peut être établie entre le SVM et le PMC dans la mesure où l'on sait que la couche de sortie du PMC est un perceptron opérant dans l'espace des neurones cachés. Les poids de sa couche de sortie sont alors équivalents aux paramètres  $\alpha_i$  des vecteurs de support. Dans l'approche Un-contre-Tous, les différents SVM sont entraînés indépendamment les uns des autres. Ils produisent alors des ensembles de vecteurs de support différents. Il en résulte que leurs sorties fournissent des mesures dont les métriques ne sont pas comparables. Nous reportons plus bas les techniques utilisées pour remédier à cet inconvénient.

La figure 2.3 montre l'architecture du système en stratégie Un-Contre-Tous. Notons, que l'étage étiqueté «fusion» désigne le schéma de vote utilisé et aussi toute sorte d'expert capable de fusionner les sorties pour décider de la classe d'appartenance.

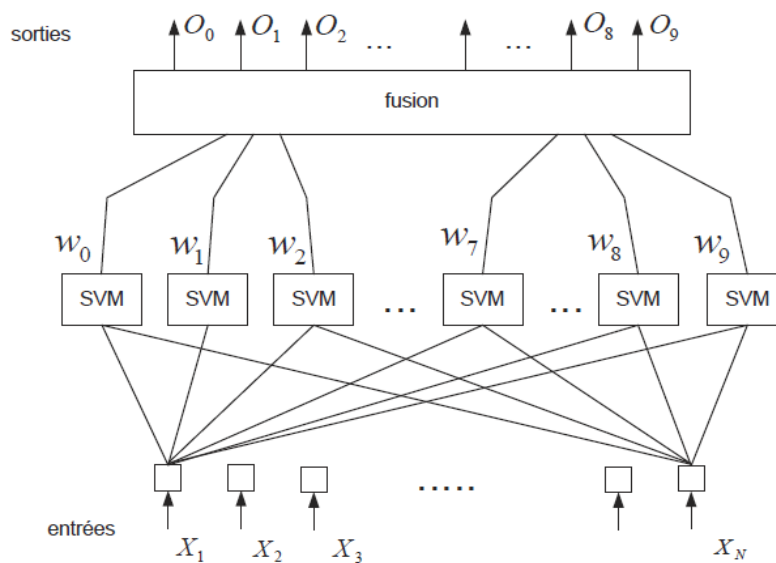


Figure 2.3 Architecture du système en stratégie Un-contre-Tous [34].

## 6. Avantages et inconvénients

### Avantages

- ✓ Absence d'optimum local.
- ✓ Contrôle explicite du compromis entre la complexité du classifieur et l'erreur.
- ✓ Possibilité d'utilisation de structure de données comme les chaînes de caractères et arbres comme des entrées.
- ✓ Traitement des données à grandes dimensions.

### Inconvénients :

- ✓ Demande des données négatives et positives en même temps.
- ✓ Besoin d'une bonne fonction Kernel.
- ✓ Problèmes de stabilité des calculs dans la résolution de certains programmes quadratiques à contraintes.

### CONCLUSION

Ce chapitre a fait l'objet de quelques définitions et généralités sur les machines à vecteurs de support. La description de cette technique est organisée selon trois parties majeures. Après un bref historique sur le SVM, la première partie introduit la théorie de l'apprentissage statistique et le principe du risque structurel. La notion de régularisation par la maximisation de la marge  $\gamma$  est expliquée. Ensuite, nous analysons le principe de la transformation de données dans l'espace augmenté à l'aide des noyaux de Mercer. La formulation mathématique du SVM  $\gamma$  est présentée. Une étude en simulation ayant pour but d'évaluer les performances de cette technique appliquée au domaine de contrôle et de surveillance des eaux propres fera l'objet du dernier chapitre. La sélection optimale des paramètres propres de la technique SVM utilisant la méthode PSO et le principe de cette technique fera l'objet du chapitre suivant.