



CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL RHÔNE-ALPES

CENTRE D'ENSEIGNEMENT DE GRENOBLE

UE ENG111 - Epreuve TEST

Travail d'Etude et de Synthèse Technique

en INFORMATIQUE

---

# Les machines à vecteurs support pour la classification en imagerie hyperspectrale : implémentation et mise en œuvre.

---

Présenté par Ludovic Mercier

le 11 février 2010, à Grenoble

devant le jury :

Président : M. Eric Gressier-Soudan  
Examineurs : M. Jean-Pierre Giraudin  
M<sup>me</sup> Véronique Panne  
M. André Plisson  
M. Eric Sellin  
M. Mathias Voisin-Fradin  
Tuteur : M. Mathieu Fauvel



# Avant propos

## Conventions typographiques

Afin de faciliter la lecture, le symbole « † » accolé à un mot spécifie une entrée dans le glossaire situé à la fin de ce document.

## Remerciements

Je remercie Mathieu Fauvel de l'INRIA et Jocelyn Chanussot du Gipsa-lab pour avoir répondu favorablement à ma demande pour le suivi de cette épreuve.

Je souhaite remercier également toute l'équipe du Laboratoire de Planétologie avec laquelle je travaille et tout particulièrement Sylvain Douté au sein du projet Vahiné.

Merci également à Frédéric Schmidt pour les documents extraits de sa thèse.

Par ailleurs un merci tout particulier à Benoît pour son amitié et le temps qu'il m'a accordé pour la relecture de ce document.

Enfin je remercie du fond du cœur ma « petite » Karen qui me soutient et m'encourage depuis le jour de ma première inscription au CNAM.



# Sommaire

<b>Table des figures</b>	<b>v</b>
<b>Introduction</b>	<b>1</b>
Imagerie hyperspectrale . . . . .	1
Machine à vecteurs support comme méthode de classification . . . . .	2
<b>1 Apprentissage, classification et SVM</b>	<b>3</b>
1.1 Notion d'apprentissage . . . . .	3
1.2 Notion de classification . . . . .	3
1.3 Machines à vecteurs support . . . . .	4
<b>2 Fonctionnement des machines à vecteurs support</b>	<b>7</b>
2.1 Principe . . . . .	7
2.2 Forme primale . . . . .	8
2.3 Forme duale . . . . .	8
2.4 Astuce du noyau . . . . .	9
2.5 Marges souples . . . . .	10
2.6 Bilan . . . . .	11
2.7 Deux approches d'optimisation . . . . .	12
<b>3 Sélection des paramètres du modèle</b>	<b>13</b>
3.1 Mesure d'erreur en apprentissage . . . . .	13
3.2 Validation croisée . . . . .	14
3.3 Grille de recherche . . . . .	14
3.4 Approche biomimétique . . . . .	15
3.4.1 Algorithmes génétiques . . . . .	15
3.4.2 Optimisation des machines à vecteurs support . . . . .	16
3.4.3 Bilan . . . . .	17
<b>4 Mise en œuvre</b>	<b>19</b>
4.1 Projet Vahiné . . . . .	19
4.2 Bibliothèque LIBSVM . . . . .	19
4.3 Volume et taille des données . . . . .	20
4.4 Classification multi-classes . . . . .	20
4.4.1 Un contre tous . . . . .	20
4.4.2 Un contre un . . . . .	21
4.4.3 Parallélisation . . . . .	21
4.5 Choix des hyperparamètres . . . . .	21

4.5.1	Fonction d'évaluation SV . . . . .	21
4.5.2	Multi-classes . . . . .	22
	<b>Conclusion</b>	<b>23</b>
	<b>Glossaire</b>	<b>25</b>
	<b>Index</b>	<b>26</b>
	<b>Bibliographie</b>	<b>27</b>

# Table des figures

1	Décomposition d'une image hyperspectrale. . . . .	2
1.1	Classification de spectres pour déterminer des types de surface : glace, H <sub>2</sub> O, roche, CO <sub>2</sub> par exemple. . . . .	3
1.2	Schéma d'un mélange granulaire auquel sera associé un mélange spectral. . . . .	4
1.3	Exemples de spectres du givre sur la calotte polaire martienne à différentes longitudes subsolaires ( $L_s$ ) [Appéré, 2010]. . . . .	5
1.4	Exemple de classification binaire linéaire et non-linéaire dans $\mathbb{R}^2$ . . . . .	5
2.1	Hyperplan de séparation optimal avec marge souple dans un cas non linéairement séparable. . . . .	8
2.2	La transformation linéaire des données permet une séparation linéaire dans un nouvel espace. Adapté de [Cornuéjols <i>et al.</i> , 2002]. . . . .	10
2.3	Exemple de classification binaire avec différents noyaux. Le choix du noyau influence la résolution. Source : « AT&T Research Lab », laboratoire de recherche AT&T <a href="http://svm.dcs.rhbnc.ac.uk/">http://svm.dcs.rhbnc.ac.uk/</a> . . . . .	11
3.1	Exemple de grille de recherche des paramètres ( $\gamma, C$ ) d'un modèle. La légende représente la précision obtenue (en pourcentage) sur un échantillon de test. . . . .	14
3.2	Principe des opérateurs génétiques dans le cas d'individu codé sur 7 bits. . . . .	15
3.3	Codage de l'information d'un chromosome pour une optimisation d'un séparateur à vaste marge. . . . .	16
4.1	Architecture parallèle de classifieur SVM binaire. . . . .	20



# Introduction

L'objectif de ce rapport est de présenter les machines à vecteurs support (SVM<sup>1</sup>) et les différentes approches d'optimisation qui leur sont associées. Dans un premier chapitre, nous rappelons la notion d'apprentissage et de classification et nous présentons les SVM. Dans un second chapitre, nous décrivons le fonctionnement des SVM et certaines approches d'optimisation. Le troisième chapitre traitera des méthodes pour sélectionner les paramètres optimaux. Enfin, dans le dernier chapitre nous discuterons de leurs mises en œuvre dans le cadre de l'imagerie hyperspectrale.

## Imagerie hyperspectrale

L'imagerie hyperspectrale est une technique permettant la représentation d'une même scène suivant de nombreuses bandes spectrales étroites dans des gammes de longueurs d'ondes variées (visible, infrarouge, etc.). C'est une technologie en plein développement qui permet d'accéder à de nombreuses informations sur les propriétés physiques des objets observés comparativement à l'imagerie couleur classique. L'imagerie hyperspectrale est utilisée dans de multiples domaines comme la géologie, l'écologie, l'urbanisme, la foresterie, l'agriculture, dans le domaine militaire ou encore en planétologie.

Les données issues de la télédétection hyperspectrale sont agencées en cube. Un cube est constitué de deux dimensions spatiales  $w$  et  $h$  et d'une dimension spectrale  $\lambda$ . Le nombre de pixels dans la direction  $w$ , respectivement  $h$ , est noté  $N_w$ , respectivement  $N_h$ . Le nombre de bandes spectrales ou spectels<sup>†</sup> dans la direction  $\lambda$  est noté  $N_\lambda$ . Un exemple d'image hyperspectrale et le principe de sa composition sont présentés à la figure 1. On notera qu'il existe une imagerie dite « multi-spectrale » qui se contente d'une dizaine de canaux alors que l'imagerie « hyperspectrale » dépasse la centaine de canaux ( $N_\lambda > 100$ ).

Les images hyperspectrales sont acquises par des spectro-imageurs associés à des microscopes, à des satellites (études de la surface terrestre) ou encore à des sondes spatiales (études planétologiques de mars).

Dans le domaine multi-spectral on trouvera par exemple, les satellites PLEIADES (CNES<sup>2</sup>) qui permettront l'acquisition d'images multi-spectrales de 70 cm à 250 cm de résolution spatiale [CNES, 2009]. Chaque image aura un volume compris entre 1,8 et 3,6 Go pour seulement 5 canaux spectraux. Ce type d'instrument est développé pour avoir un bon rendu spatial.

Par comparaison dans le domaine hyperspectral nous avons comme exemple les images fournies par le spectro-imageur OMEGA<sup>3</sup> (ESA<sup>4</sup>) [ESA, 2009]. Chaque image de résolution spatiale comprise entre 350 m à 4000 m. Leur taille varie de 50 à 100 Mo pour environ 100000 spectres

---

<sup>1</sup>« Support Vector Machine », machine à vecteurs support.

<sup>2</sup>Centre National d'Etude Spatiale.

<sup>3</sup>Observatoire pour la Minéralogie, l'Eau, les Glaces et l'Activité.

<sup>4</sup>« European Space Agency », l'agence spatiale européenne.

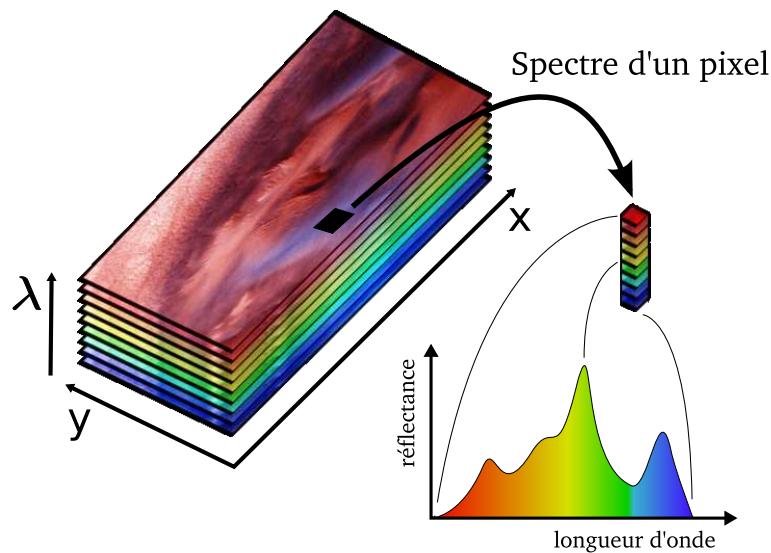


FIG. 1 – Décomposition d'une image hyperspectrale.

composés eux même de 184 spectels.

Ces exemples montrent que la taille et la complexité des données générées font apparaître de nouveaux défis méthodologiques pour l'analyse à la fois mathématique et informatique. Un des problèmes à résoudre est d'automatiser l'extraction d'informations pertinentes de ces images : détection de zone d'intérêt, quantification des matériaux etc. De récents travaux cités notamment dans [Bazi et Melgani, 2006] montrent que les méthodes de machine à vecteurs support surpassent les méthodes statistiques classiques dans les problématiques de classification en imagerie hyperspectrale. Elles se distinguent par une meilleure précision de reconnaissance et une faculté de généralisation qui permet d'obtenir une bonne classification des nouveaux exemples.

## Machine à vecteurs support comme méthode de classification

Les machines à vecteurs support ou SVM, constituent une méthode de classification supervisée particulièrement bien adaptée pour traiter des données de grande dimension. Par rapport aux techniques classiques d'apprentissage, les SVM ne dépendent pas de la dimension de l'espace de représentation des données. Grâce à l'usage d'une fonction noyau, elles permettent une classification non linéaire comme nous le verrons au paragraphe 2.4 . L'inconvénient des SVM est le choix empirique de la fonction noyau adaptée au problème. Un deuxième inconvénient est le temps de calcul qui croît de façon cubique en fonction du nombre de données à traiter. La complexité d'un algorithme SVM est cubique par rapport au nombre de données et linéaire par rapport au nombre de variables. Si le nombre de données d'apprentissage est  $n$  la dimension des données à classer est  $d$ , la complexité est alors en  $O(dn^3)$  [Bousquet, 2001].

# Apprentissage, classification et SVM

## 1.1 Notion d'apprentissage

L'apprentissage c'est l'acquisition de connaissances et compétences permettant la synthèse d'information [Bisson, 2009]. Un algorithme d'apprentissage va permettre de passer d'un espace des exemples  $X$  à un espace dit des hypothèses  $H$ . L'algorithme SVM va explorer l'espace  $H$  pour obtenir le meilleur hyperplan séparateur (voir le paragraphe 1.3). L'apprentissage permet, à partir d'un ensemble de paramètres en entrée, d'obtenir un ensemble de résultats en sortie. On va donc préparer un jeu de données à apprendre constitué de couples paramètre/résultat. Le but recherché est d'apprendre une fonction permettant de prédire de nouveaux résultats pour de nouvelles entrées. L'apprentissage est dit supervisé ou non supervisé. Les SVM se situent dans le groupe des algorithmes d'apprentissage supervisés puisque que l'on utilise une base d'exemples pour obtenir la règle de classification.

## 1.2 Notion de classification

La classification<sup>†</sup> est une opération de structuration qui vise à organiser un ensemble d'observation en groupes homogènes et contrastés afin de faciliter l'analyse des informations et d'effectuer des prédictions [Bisson, 2009]. En imagerie hyperspectrale une classe est un ensemble de spectres ayant des caractéristiques similaires ou communes. Sur la figure 1.1 nous présentons une classification imagée de différents spectres.

Dans le cadre de la télédétection des surfaces planétaires, les caractéristiques communes sont :

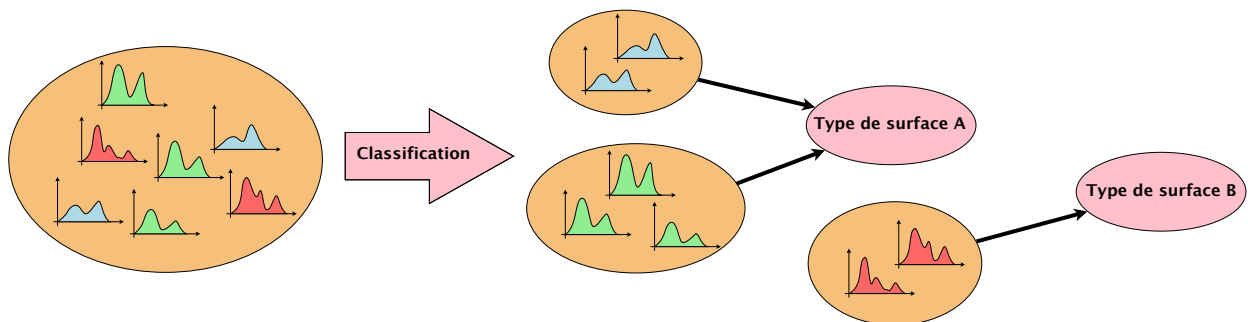


FIG. 1.1 – Classification de spectres pour déterminer des types de surface : glace, H<sub>2</sub>O, roche, CO<sub>2</sub> par exemple.

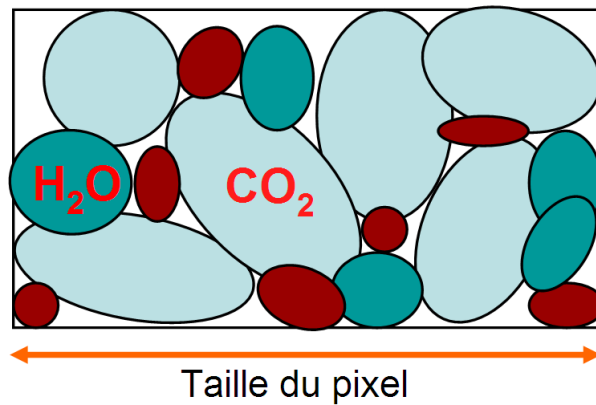


FIG. 1.2 – Schéma d'un mélange granulaire auquel sera associé un mélange spectral.

la présence d'un même corps chimique, la même représentation de surface, les mêmes paramètres physiques au sein de la représentation de surface (granulométrie, abondances, ...).

La définition d'une classe se fait soit par l'ensemble des spectres considérés comme similaires, soit par un spectre de référence<sup>†</sup>. Il est aussi nécessaire de définir une notion de similitude. On notera que la similitude en apprentissage se base essentiellement sur la notion de distance<sup>†</sup> [Bisson, 2009]. On utilisera alors la distance de Minkowski<sup>1</sup>  $D_p(X_i, X_j) = (\sum_k W_k |x_{ik} - x_{jk}|^p)^{\frac{1}{p}}$  avec  $p > 0$ . Avec les définitions précédentes, la classification cherche à minimiser la distance intra-classes (classe homogène) et par contre à maximiser la distance inter-classes (classes contrastées).

Concernant la classification d'images de surface, on peut travailler soit avec des spectres typiques de composé chimique pur, soit avec des spectres de mélange (ils seront appelés spectres synthétiques). En effet un pixel correspond à une surface de terrain hétérogène comme le montre la figure 1.2. La surface du pixel varie de plusieurs dizaines de centimètre carré à plusieurs centaines de mètre carré suivant la résolution du spectro-imageur.

La figure 1.3 montre des exemples de spectre de la surface de mars mesurés par le spectro imageur OMEGA. La complexité des surfaces planétaires va impliquer de nombreuses combinaisons de spectres synthétiques. Cette situation physique conduit à des bases d'apprentissage qui peuvent contenir plusieurs millions d'individus. Cet état de fait va impacter le temps d'apprentissage de part la complexité cubique de l'algorithme.

### 1.3 Machines à vecteurs support

Les machines à vecteurs support ont été introduites en 1995 par Cortes et Vapnik [Cortes et Vapnik, 1995, Bottou et Chih-Jen, 2007]. Elles sont utilisées dans de nombreux problèmes d'apprentissage : reconnaissance de forme, catégorisation de texte ou encore diagnostic médical.

Les SVM reposent sur deux notions : celle de marge maximale et celle de fonction noyau. Elles permettent de résoudre des problèmes de discrimination non linéaire. Nous allons voir comment un problème de classification se ramène à résoudre un problème d'optimisation quadratique.

La marge est la distance entre la frontière de séparation et les échantillons les plus proches appelés vecteurs support. Dans un problème linéairement séparable les SVM trouvent une séparatrice qui maximise cette marge. Dans le cas d'un problème non linéaire on utilise une fonction noyau pour projeter les données dans un espace de plus grande dimension où elles seront linéairement séparables (figure 1.4).

<sup>1</sup>On retrouve la distance euclidienne avec  $p = 2$  et la pondération  $W_k = 1$ .

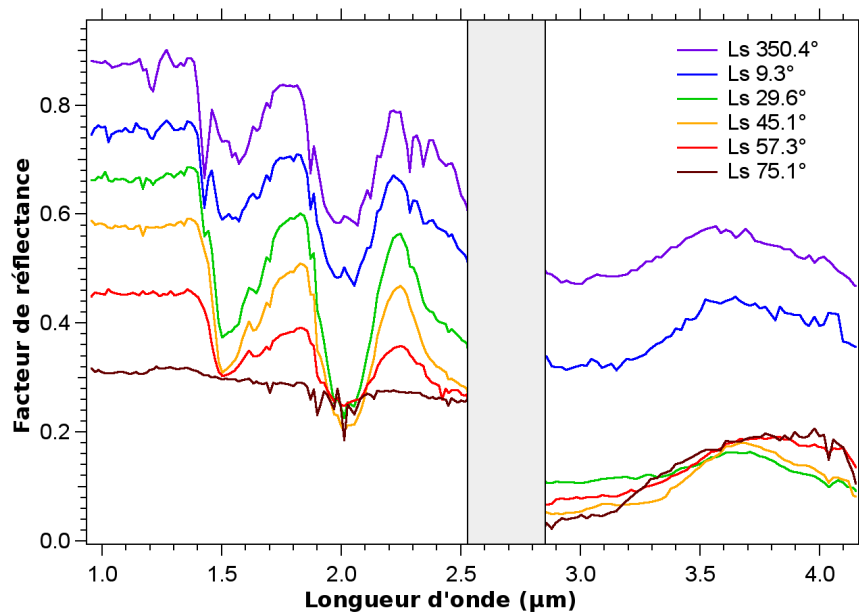


FIG. 1.3 – Exemples de spectres du givre sur la calotte polaire martienne à différentes longitudes subsolaires ( $L_s$ ) [Appéré, 2010].

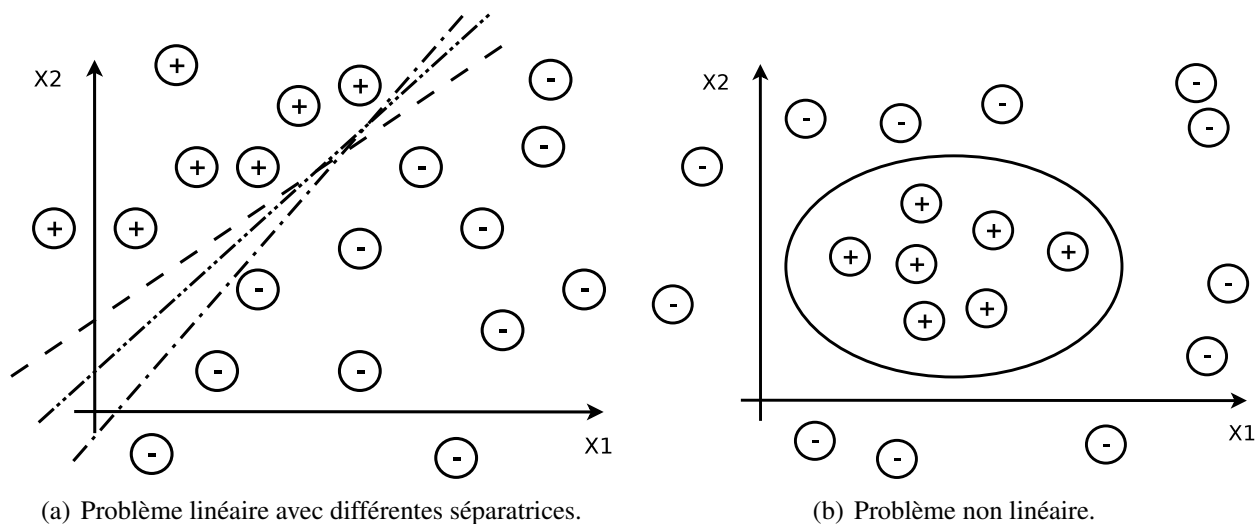


FIG. 1.4 – Exemple de classification binaire linéaire et non-linéaire dans  $\mathbb{R}^2$ .



# Fonctionnement des machines à vecteurs support

## 2.1 Principe

Les machines à vecteurs support forment une classe d'algorithmes d'apprentissage supervisés. Nous nous intéressons à une fonction notée  $f$  qui à toute entrée  $x$  fait correspondre une sortie  $y = f(x)$ . Le but est d'essayer d'apprendre  $f$  à partir d'un ensemble de couple  $(\mathbf{x}_i, \mathbf{y}_i)$ . Dans ce problème les machines à vecteurs support vont être utilisées pour classifier une nouvelle observation  $\mathbf{x}$  en se limitant à deux classes  $y \in \{-1, 1\}$ .

Nous allons donc construire une fonction  $f$  qui à chaque valeur d'entrée dans un ensemble  $\mathbb{R}^d$  va faire correspondre une valeur de sortie  $y \in \{-1, 1\}$  :

$$f : \mathbb{R}^d \rightarrow \{-1, 1\}, f(\mathbf{x}) = y.$$

Dans le cas linéaire, une fonction discriminante  $h$  est obtenu par combinaison linéaire d'un vecteur d'entrée  $\mathbf{x} = (x_1, \dots, x_d)$  et s'écrit :

$$h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b. \tag{2.1}$$

La classe est donnée par le signe de  $h(\mathbf{x})$  :  $f(\mathbf{x}) = \text{sign}(h(\mathbf{x}))$ . Si  $h(\mathbf{x}) \geq 0$  alors  $\mathbf{x}$  est de classe 1 sinon  $\mathbf{x}$  est de classe  $-1$ . La séparatrice est alors un hyperplan affine d'équation :  $\mathbf{w} \cdot \mathbf{x} + b = 0$ . Si  $(\mathbf{x}_i, \mathbf{y}_i)$  est un des  $p$  elements de la base d'apprentissage noté  $A_p$ , on veut trouver le classifieur  $h$  tel que :

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 0, i \in [1, p]. \tag{2.2}$$

Dans le cas simple linéairement séparable il existe de nombreux hyperplans séparateurs comme nous pouvons le voir sur la figure 1.4. Selon la théorie de Vapnick [Cortes et Vapnik, 1995, Vapnik, 1995] l'hyperplan optimal (optimum de la distance inter-classe) est celui qui maximise la marge. Cette dernière étant définie comme la distance entre un hyperplan et les points échantillons les plus proches. Ces points particuliers sont appelés vecteurs support. La distance entre un point  $\mathbf{x}$  quelconque et l'hyperplan est donnée par l'équation 2.3.

$$d(\mathbf{x}) = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|}. \tag{2.3}$$

Nous allons voir au paragraphe suivant que maximiser la marge va revenir à minimiser  $\|\mathbf{w}\|$ .

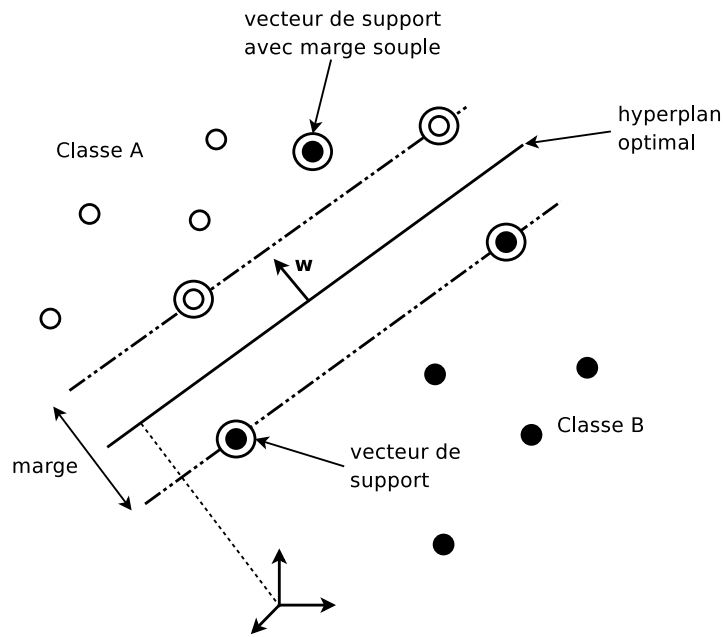


FIG. 2.1 – Hyperplan de séparation optimal avec marge souple dans un cas non linéairement séparable.

## 2.2 Forme primale

Les paramètres  $w$  et  $b$  étant définis à un coefficient multiplicatif près, on choisit de les normaliser pour que les échantillons les plus proches ( $\mathbf{x}_s$ ) vérifient l'égalité suivante :

$$y_s(\mathbf{w} \cdot \mathbf{x}_s + b) = 1$$

donc quelque soit l'échantillon  $\mathbf{x}_i$  on obtient :

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \quad (2.4)$$

La distance entre l'hyperplan et un point support est donc définie par  $\frac{1}{\|\mathbf{w}\|}$ . La marge géométrique entre deux classes est égale à  $\frac{2}{\|\mathbf{w}\|}$ . La forme primale (qui dépend seulement de  $\mathbf{w}$  et  $b$ ) des SVM est donc un problème de minimisation sous contrainte qui s'écrit :

$$\begin{cases} \min(\frac{1}{2} \|\mathbf{w}\|^2) \\ \forall (\mathbf{x}_i, y_i) \in A_p, y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \end{cases} \quad (2.5)$$

## 2.3 Forme duale

La formulation primale peut être transformée en formulation duale en utilisant les multiplicateurs de Lagrange. L'équation 2.5 s'écrit alors sous la forme suivante :

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^p \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1). \quad (2.6)$$

La formulation de Lagrange permet de trouver les extremums en annulant les dérivées partielles de la fonction  $L(\mathbf{w}, b, \alpha)$ . Le lagrangien  $L$  doit être minimisé par rapport à  $\mathbf{w}$  et  $b$  et maximisé par rapport à  $\alpha$ . On résout ce nouveau problème en calculant les dérivées partielles :

$$\frac{\partial L}{\partial \mathbf{w}} = w - \sum_{i=1}^p \alpha_i y_i \mathbf{x}_i = 0 \quad (2.7)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^p \alpha_i y_i = 0. \quad (2.8)$$

En réinjectant les deux premières dérivées partielles 2.7 et 2.8 dans l'équation 2.6 nous obtenons :

$$L(\alpha) = \frac{1}{2} \sum_{i=1}^p \alpha_i y_i \sum_{j=1}^p \alpha_j y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^p \alpha_i y_i \sum_{j=1}^p \alpha_j y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^p \alpha_i y_i b + \sum_{i=1}^p \alpha_i$$

on en extrait la formulation duale (dépendant des  $\alpha_i$ ) suivante :

$$L(\alpha) = \sum_{i=1}^p \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \quad (2.9)$$

On cherche donc à maximiser  $L(\alpha)$  sous les contraintes  $\alpha_i \geq 0$  et  $\sum_i \alpha_i y_i = 0$ . A l'optimal,  $\alpha^*$ , les conditions de Karush Kuhn Tucker (conditions KKT) sont satisfaites et permettent d'écrire l'égalité suivante :

$$\alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0, \forall i \in [1, p]. \quad (2.10)$$

Cela nous donne  $\alpha_i = 0$  ou  $(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = 0$ . Ces deux possibilités impliquent que seuls les  $\alpha_i$  associés à des exemples situés sur la marge peuvent être non nuls. Autrement dit, ces exemples sur la marge constituent les vecteurs support, qui seuls contribuent à définir l'hyperplan optimal.

Cette maximisation est un problème de programmation quadratique de dimension égale au nombre d'exemple. L'équation 2.7 nous donne la valeur optimale pour  $w$  noté  $w^*$  :  $w^* = \sum_{i=1}^p \alpha_i^* y_i \mathbf{x}_i$ , avec  $\alpha_i^*$  les coefficients de Lagrange optimaux. En utilisant l'équation de l'hyperplan 2.1 nous obtenons l'hyperplan de marge maximale :

$$h(x) = \sum_{i=1}^p \alpha_i^* y_i \mathbf{x} \cdot \mathbf{x}_i + b. \quad (2.11)$$

## 2.4 Astuce du noyau

Le cas linéairement séparable est peu intéressant, car les problèmes de classification sont souvent non linéaires. Pour résoudre ce point la méthode classique est de projeter les données dans un espace de dimension supérieur appelé espace de redescription. L'idée étant qu'en augmentant la dimensionnalité du problème on se retrouve dans le cas linéaire vu précédemment. Nous allons donc appliquer une transformation non linéaire  $\Phi(\bullet)$  aux vecteurs d'entrée  $\mathbf{x}_i$  tel que  $\mathbf{x}_i \in \mathbb{R}^d$  et  $\Phi(\mathbf{x}_i) \in \mathbb{R}^e$ , ( $e > d$ ). Ce changement va conduire à passer d'un produit scalaire dans l'espace d'origine  $\mathbf{x}_i \cdot \mathbf{x}_j$  à un produit scalaire  $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$  dans l'espace de redescription (voir la figure 2.2). L'astuce est d'utiliser une fonction noyau notée  $K$  qui évite le calcul explicite du produit scalaire dans l'espace de redescription. Les fonctions noyaux doivent satisfaire le théorème de Mercer<sup>†</sup>. Nous avons alors l'égalité suivante :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \quad (2.12)$$

Il existe de nombreuses fonctions noyau prédéfinies, les deux les plus usitées sont le noyau gaussien (équation 2.13) et le noyau polynomial (équation 2.14) :

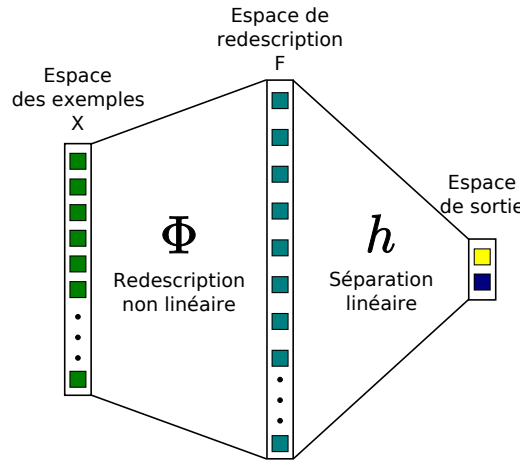


FIG. 2.2 – La transformation linéaire des données permet une séparation linéaire dans un nouvel espace. Adapté de [Cornuéjols *et al.*, 2002].

$$K_{\gamma}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2} \quad (2.13)$$

$$K_{\gamma, d, r}(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i \cdot \mathbf{x}_j + r)^d. \quad (2.14)$$

Les noyaux gaussien sont des noyaux dits de type radial (fonction à base radial abrégé RBF [Bottou et Chih-Jen, 2007]), indiquant qu'ils dépendent de la distance entre deux exemples.

L'hyperplan séparateur se réécrit avec la fonction noyau sous la forme suivante :

$$h(\mathbf{x}) = \sum_{i=1}^p \alpha_i^* y_i K_{\gamma}(\mathbf{x}, \mathbf{x}_i) + b.$$

## 2.5 Marges souples

Les machines à vecteurs support sont efficaces quand le problème est séparable. Nous avons vu dans le paragraphe précédent 2.4 que l'utilisation d'une méthode noyau permettait de traiter les cas non linéaires mais cela n'est pas utilisable dans le cas de données non séparables par exemple pour des données bruitées. En effet, on peut avoir des éléments à classer du mauvais côté de l'hyperplan comme le montre la figure 2.1. Cortes et Vapnik en 1995 ont donc introduit le concept de marge souple [Cortes et Vapnik, 1995]. Certains exemples d'apprentissage peuvent violer la contrainte 2.4 que l'on retrouve dans l'équation de la forme primale 2.5. On introduit par conséquent des variables dites « ressorts »  $\xi = (\xi_1, \dots, \xi_p)$  qui permettent d'assouplir la contrainte pour chaque exemple. Un paramètre supplémentaire de régularisation  $C$  est ajouté pour contrôler la pénalité associée aux exemples. La nouvelle forme primale décrite en 2.5 devient alors :

$$\begin{cases} \min (\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^p \xi_i) \\ \forall (\mathbf{x}_i, y_i) \in A_p, y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i. \end{cases} \quad (2.15)$$

De même que pour la forme primale nous obtenons une nouvelle formulation duale qui est alors similaire à celle décrite dans la partie 2.3. Si on utilise en plus la fonction noyau  $K$  dans la formulation duale 2.9 en appliquant la méthode des multiplicateurs de Lagrange on cherche alors à maximiser la nouvelle fonction  $L(\alpha)$ .

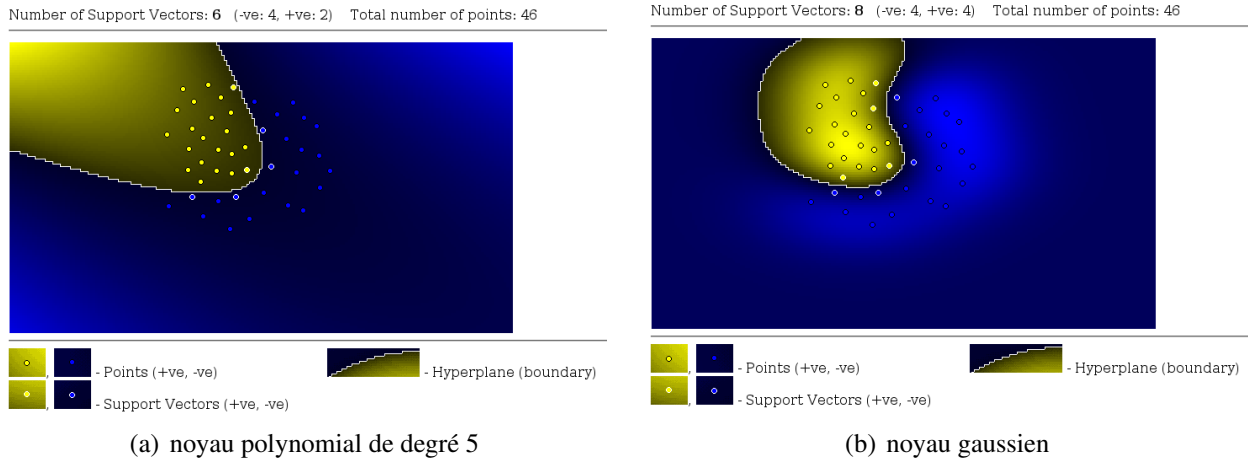


FIG. 2.3 – Exemple de classification binaire avec différents noyaux. Le choix du noyau influence la résolution. Source : « AT&T Research Lab », laboratoire de recherche AT&T <http://svm.dcs.rhbnc.ac.uk/>.

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

En appliquant la même méthode qu' au paragraphe 2.3 on obtient  $L(\alpha)$  à partir de l'expression du Lagrangien précédent.

$$L(\alpha) = \sum_{i=1}^p \alpha_i - \frac{1}{2} \sum_{i,j=1}^p \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (2.16)$$

$$\forall (\mathbf{x}_i, y_i) \in A_p, 0 \leq \alpha_i \leq C \text{ et } \sum_i y_i \alpha_i = 0.$$

Le seul changement est la contrainte supplémentaire sur les coefficients  $\alpha_i$ , qui se traduit par une borne supérieure  $C$ . La solution de l'équation précédente 2.16 est de la forme :

$$h(\mathbf{x}) = \sum_{i=1}^p \alpha_i^* y_i K(\mathbf{x}, \mathbf{x}_i) + b. \quad (2.17)$$

## 2.6 Bilan

Le calcul des  $\alpha_i^*$  du Lagrangien vu précédemment est obtenu par programmation quadratique. C'est un problème bien connu pour lequel il existe de nombreux algorithmes d'optimisation. Ce type de problème a un optimum global unique, élément important car nous sommes sûrs de ne pas atteindre un optimum local. Il s'ensuit un seul  $\alpha^*$  qui maximise l'équation 2.16.

Une fois que l'on a obtenu les coefficients  $\alpha^*$ , le coefficient  $b$  peut être obtenu en utilisant n'importe quel vecteur  $\mathbf{x}_i$  avec l'égalité 2.10.

Des paragraphes précédents on soulignera que la solution dépend uniquement d'un nombre restreint d'exemple de l'ensemble d'apprentissage et non pas de la dimension  $d$  de l'espace des exemples. La figure 2.3 montre deux exemples de classification binaire obtenus avec deux noyaux différents. On remarquera la faible proportion (17%) des points dits « vecteurs support » par rapport au nombre de points constituant l'ensemble des exemples d'apprentissage.

## 2.7 Deux approches d'optimisation

Généralement les algorithmes SVM se concentrent sur la résolution du problème duale. Néanmoins un article de O. Chapelle [Chapelle, 2007] présente une optimisation de la forme primale<sup>1</sup> grâce à un algorithme de type Newton. La complexité obtenue est du même ordre que pour une optimisation de la forme duale ( $O(n^3)$ ).

L'équation 2.15 peut se réécrire comme un problème d'optimisation sans contrainte :

$$\|w\|^2 + C \sum_{i=1}^n L(y_i, \mathbf{w} \cdot \mathbf{x}_i + b) \quad (2.18)$$

avec  $L(y, t) = \max(0, 1 - yt)^p$  une fonction de perte<sup>†</sup>. Cette équation 2.18 peut être reformulée sous une forme non détaillée ici, mais qui peut s'optimiser par différentes techniques (gradient conjugué, méthode de Newton, ...).

L'expérimentation comparée sur les deux formulations primale et duale montre des temps de calcul inférieurs pour l'optimisation primale (respectivement duale) quelque soit la taille de l'ensemble d'entraînement si le problème est linéaire (respectivement non linéaire). Néanmoins si le problème à un grand nombre d'exemples à apprendre une solution exacte n'est pas envisageable. L'optimisation de la forme primale est alors meilleure car on minimise directement la quantité qui nous intéresse. O. Chappel montre d'une part un gain d'itération de 30% pour atteindre la convergence et d'autre part l'utilisation possible de méthodes numériques plus simples (comme le gradient conjugué<sup>†</sup>). Cela rend l'optimisation de la forme primale intéressante. Néanmoins historiquement la forme duale est la plus employée, nous allons présenter par la suite certaines méthodes d'optimisation consacrées à cette dernière. Les résultats de la formulation duale sont influencés par le choix des hyperparamètres ( $C, \gamma$ ). Dans le chapitre suivant nous allons donc voir les méthodes de sélection des hyperparamètres optimaux.

---

<sup>1</sup>Il existe d'autres références antérieures concernant l'optimisation de la forme primale, on se référera à l'introduction des travaux de O. Chapelle.

## Sélection des paramètres du modèle

La résolution de la méthode des machines à vecteurs support implique la sélection de plusieurs paramètres : le type de noyau, le ou les paramètres du noyau ( $\gamma, \dots$ ) et le paramètre de régularisation  $C$ . Plusieurs méthodes existent pour sélectionner ces paramètres. Une technique classique consiste à choisir une grille de recherche pour  $C$  et  $\gamma$  puis à appliquer une méthode de sélection en cherchant l'optimum d'un critère de qualité associé à un couple  $(C, \gamma)$ . Au cours de ce chapitre nous allons considérer que nous travaillons avec un noyau gaussien. Nous avons donc un seul paramètre noyau  $\gamma$  à sélectionner.

### 3.1 Mesure d'erreur en apprentissage

En utilisant des algorithmes d'apprentissage nous devons considérer deux critères : l'erreur d'apprentissage  $E_a$  et l'erreur de généralisation  $E_g$  [Bisson, 2009]. L'erreur d'apprentissage correspond au taux de mauvais classement sur l'ensemble d'apprentissage. L'erreur de généralisation correspond au taux de mauvais classement sur l'ensemble de test. Pour comparer les modèles d'apprentissage il est intéressant de calculer une table de contingence (table 3.1).

A partir de celle-ci on peut alors obtenir différents critères tel que le taux de reconnaissance donné par la formule 3.1 [Bisson, 2009].

$$T_{rec} = \frac{A + D}{A + B + C + D}. \quad (3.1)$$

Ou encore la précision et le rappel du modèle donné respectivement par 3.2 et 3.3.

$$T_{prec} = \frac{A}{A + B}. \quad (3.2)$$

$$T_{rappel} = \frac{A}{A + C}. \quad (3.3)$$

		classe réelle	
		classe +	classe -
classe prédite	classe +	A vrai positifs	B faux positifs
	classe -	C faux négatifs	D vrais négatifs

TAB. 3.1 – table de contingence

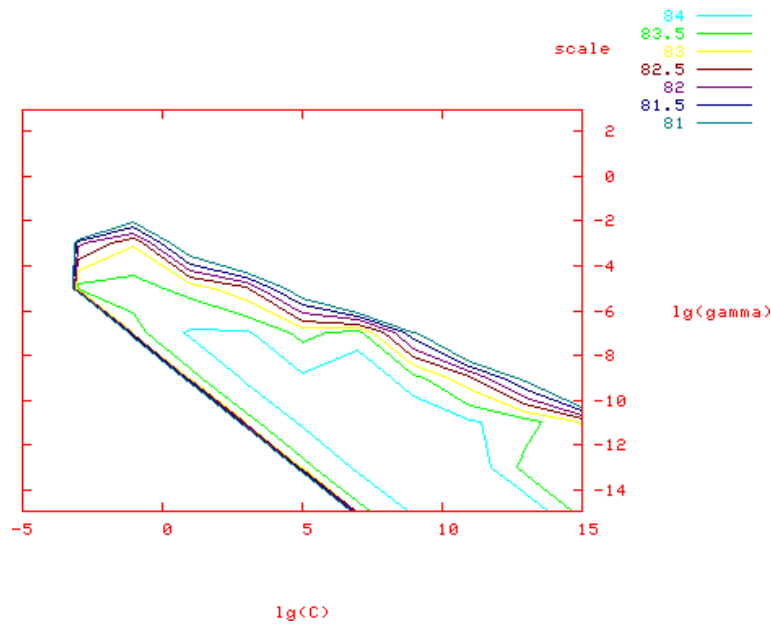


FIG. 3.1 – Exemple de grille de recherche des paramètres  $(\gamma, C)$  d'un modèle. La légende représente la précision obtenue (en pourcentage) sur un échantillon de test.

## 3.2 Validation croisée

La validation croisée est une technique qui permet de tester un modèle d'apprentissage. La validation croisée se décline en plusieurs sous-méthodes. La plus répandue est la méthode « k-Fold » avec typiquement  $k \in [4, 10]$ . Si l'on a une base d'apprentissage  $A_p$  contenant  $p$  éléments :  $A_p = \{\mathbf{x}_1, \dots, \mathbf{x}_p\}$  la validation croisée consiste à appliquer les cinq étapes suivantes :

1. Découper l'ensemble des exemples en  $k$  sous-ensembles disjoints de taille  $\frac{p}{k}$ .
2. Apprendre sur les  $k - 1$  sous-ensembles.
3. Calculer l'erreur sur la  $k^{ime}$  partie.
4. Répéter le processus  $p$  fois.
5. Obtenir l'erreur finale en calculant la moyenne des  $k$  erreurs précédentes.

La validation croisée est simple à mettre en œuvre et utilise toutes les données. Elle permet d'obtenir une estimation de l'erreur de généralisation. Cela permet d'éviter le surapprentissage [Hsu *et al.*, 2009].

## 3.3 Grille de recherche

Comme énoncé précédemment, nous cherchons le meilleur couple  $(C, \gamma)$ . Nous allons donc par exemple utiliser des séquences exponentielles [Hsu *et al.*, 2009] :  $C = 2^{-5}, 2^{-4}, \dots, 2^{15}$  et  $\gamma = 2^{-15}, 2^{-14}, \dots, 2^3$ . En déterminant une heuristique sur la qualité des résultats on pourrait éviter une recherche exhaustive de toutes les combinaisons de paramètre. Une recherche exhaustive est quand même envisageable car les paramètres sont indépendants et nous pouvons donc facilement paralléliser la recherche. Nous pourrions néanmoins envisager une recherche par gradient ou par faisceau. Cette dernière pourrait aussi être facilement parallélisée.

La figure 3.1 montre un exemple de calcul exhaustif du taux de validation croisée pour une grille de valeurs  $(C, \gamma)$ . Après une première évaluation on peut relancer la procédure en raffinant la grille. Dans notre exemple (figure 3.1) nous pourrions relancer sur les intervalles  $[2^3, 2^{14}]$  pour  $C$  et  $[2^{-14}, 2^{-7}]$  pour  $\gamma$ . On pourra trouver d'autres exemples dans le document [Hsu *et al.*, 2009].

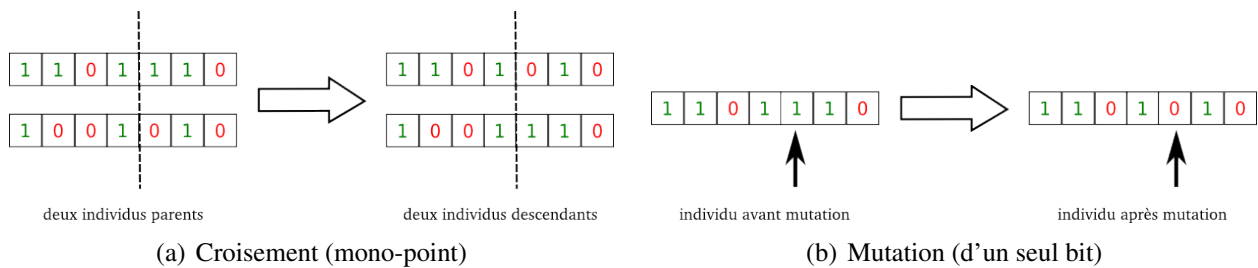


FIG. 3.2 – Principe des opérateurs génétiques dans le cas d'individu codé sur 7 bits.

Une méthode de recherche exhaustive sur grille a été appliquée dans le cas hyperspectral par l'équipe VAHINE<sup>1</sup>. Le principal problème lors de la mise en œuvre consiste à trouver une fonction pour estimer les meilleurs résultats.

## 3.4 Approche biomimétique

S'inspirant de principes issus de la biologie, les algorithmes biomimétiques comme les algorithmes génétiques (AG) ont été utilisés avec succès dans de nombreux domaines d'application. Dans le cas du traitement d'images, les AG sont employés pour résoudre des problèmes de classification et de réduction de dimensionnalité. Dans le cadre de l'imagerie hyperspectrale, une approche proposée [Bazi et Melgani, 2006] est d'utiliser les AG pour deux choses : d'une part sélectionner les meilleures composantes discriminantes des individus à classifier (réduction de dimensionnalité) et d'autre part estimer les meilleures valeurs pour les paramètres de régularisation et les paramètres noyau.

### 3.4.1 Algorithmes génétiques

Les algorithmes génétiques ont été conçus par J. Holland [Holland, 1975]. Ce sont des algorithmes d'optimisation. On trouvera une présentation détaillée de ces processus dans de nombreux ouvrages [Rennard, 2002]. Les AG reposent sur plusieurs hypothèses :

- un codage homogène des informations,
- une utilisation d'une population de modèles et non un seul modèle,
- une mesure d'adaptation des modèles à l'environnement,
- une reproduction des modèles avec modification et échange d'informations non déterministes.

La recherche de la meilleure solution se fait donc par l'évolution d'une population d'individu qui sont soumis à deux mécanismes tirés de la génétique : la mutation et le croisement. D'une façon générale, les étapes d'un algorithme génétique sont décrites dans l'algorithme 1. Des variantes existent mais les points communs sont les hypothèses précédemment listées.

Plusieurs remarques à cette étape s'imposent :

- le codage des individus ou chromosomes se fait sous la forme d'une chaîne de bits,
- le croisement produit deux descendants à partir de deux parents à partir d'un point de coupe choisit aléatoirement dans le chromosome des parents,
- la mutation produit un nouvel individu à partir d'un seul individu ; on opère un complément à 1 de certain bit.

Du point de vue algorithmique, les AG sont intéressants comparativement à d'autres algorithmes tel que le recuit-simulé car ils traitent et font évoluer une population de solutions et non une seule

<sup>1</sup>« Visualization and analysis of multi-dimensional hyperspectral images in Astrophysics », visualisation et analyse d'image hyperspectrale en Astrophysique.

**Algorithme 1** Principe d'un algorithme génétique.

- 1: **début**
- 2: génération aléatoire d'une population de  $N$  individus
- 3: **répéter**
- 4: reproduction et donc production de  $M$  descendants par croisement et mutation de la population
- 5: évaluation de chaque individu avec une fonction d'adaptation au problème
- 6: remplacement de tout ou partie de la population par les  $N$  meilleurs individus parmi les  $N + M$  disponibles
- 7: **jusqu'à** convergence d'un critère de qualité ou nombre maximal de génération atteint
- 8: **fin**

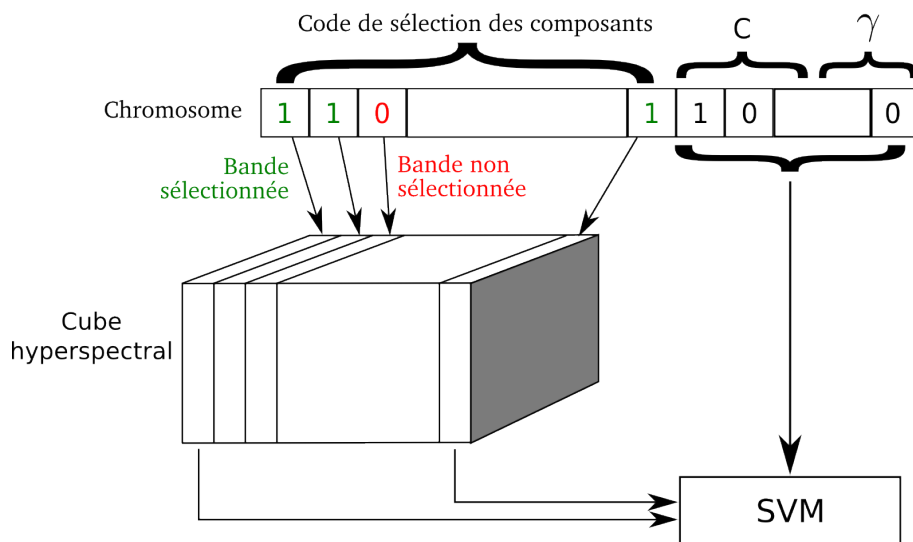


FIG. 3.3 – Codage de l'information d'un chromosome pour une optimisation d'un séparateur à vaste marge.

solution. De plus les mécanismes d'évolution 3.2 font que les individus de la population courante interagissent entre eux au lieu d'évoluer chacun indépendamment.

### 3.4.2 Optimisation des machines à vecteurs support

La construction de la population d'individu ou de chromosome est basé sur le codage des composantes pertinentes de chaque exemple et des paramètres du modèle. Les exemples sont des spectres sous forme de vecteur  $x_i \in \mathfrak{R}^d$  à  $d$  composantes correspondant au nombre de spectels des images hyperspectrales à traiter ( $d = N_\lambda$ ). Si nous restons dans le cas de l'utilisation d'un noyau gaussien nous avons deux paramètres  $(C, \gamma)$  pour notre modèle SVM. Nous allons donc considérer une population de  $p$  chromosomes  $C_m$  tel que  $C_m \in \mathfrak{R}^{d+2}$ . Le chromosome  $C_m$  est un vecteur de bits qui permet le codage de l'utilisation ou non d'une composante des  $x_i$  et le codage du couple de paramètres  $(C, \gamma)$  du noyau (voir la figure 3.3).

Une fois le codage effectué, l'élément important est la fonction d'évaluation de chaque chromosome. Les auteurs de [Bazi et Melgani, 2006] proposent deux fonctions d'évaluation : R2W2 (« radius margin bound », marge radial de liaison) et SV (« support vector count », nombre de vecteur support). Un résultat très intéressant de cet article est la comparaison des méthodes SVM avec l'utilisation de R2W2 ou de SV. Cette dernière a de nombreux points positifs : elle est moins complexe à implémenter et a une précision plus importante >80% (R2W2 <80%). SV a aussi une probabilité de fausse alarme inférieure à R2W2 (0,1 au lieu de 0,4). On peut donc en conclure que

la fonction SV est une bonne candidate pour l'étape d'évaluation. Il est à noter que la fonction SV est aussi utilisée par les auteurs de [Ghoggali *et al.*, 2009] comme fonction d'évaluation dans un contexte identique.

L'algorithme proposé par [Bazi et Melgani, 2006] reprend les grandes étapes présentées précédemment en ajoutant une classification finale avec les caractéristiques du chromosome ayant la meilleure fonction d'évaluation. Cet algorithme effectue une classification binaire ce qui n'est généralement pas suffisant. Les auteurs proposent donc une amélioration multi-classes que nous ne détaillerons pas ici.

### 3.4.3 Bilan

L'avantage de cette méthode est qu'elle permet une réduction de dimensionnalité sans connaissance à priori sur le nombre de composantes à sélectionner. Elle permet en parallèle de sélectionner les paramètres optimaux du noyau. Les résultats des comparatifs avec des méthodes SVM classiques sont meilleurs. On obtient entre 1 et 10% de précision supplémentaire. Cette précision se fait au détriment du temps d'apprentissage qui est en moyenne dix fois supérieur.



## Mise en œuvre

Après avoir présenté le principe des machines à vecteurs support nous allons voir sa mise en œuvre dans le cas de l'imagerie hyperspectrale planétologique. Son usage est étudié dans le cadre du projet de recherche Vahiné<sup>1</sup> [INRIA *et al.*, 2008]. Nous avons vu dans les chapitres précédents qu'un programme d'apprentissage par SVM se ramène à la résolution d'un problème d'optimisation quadratique. De nombreux outils sont disponibles pour l'implémentation ou l'utilisation d'un algorithme SVM, on trouvera notamment des références sur le site : [kernel-machines.org](http://www.kernel-machines.org) <http://www.kernel-machines.org>. Après une introduction au projet Vahiné nous détaillerons succinctement une implémentation de référence au paragraphe 4.2.

### 4.1 Projet Vahiné

Le projet Vahiné cofinancé par l'ANR<sup>2</sup> et le CNES a pour but de développer des modèles mathématiques, physiques, des algorithmes et des logiciels capables de travailler avec des données hyperspectrales. Ces données sont aussi bien d'origine spatiale<sup>3</sup>, expérimentale ou simulée<sup>4</sup>. Le projet vise à terme à publier un logiciel libre d'analyse et de traitement d'image hyperspectrale.

Le projet Vahiné travaille sur des données planétaires issues des spectro-imageurs OMEGA (Observatoire pour la Minéralogie, l'Eau, les Glaces et l'Activité) ou CRISM (« Compact Reconnaissance Imaging Spectrometer for Mars », spectro-imageur de reconnaissance compacte pour Mars.). En complément, de nombreuses données synthétiques sont aussi créées à partir de code de simulation physique.

### 4.2 Bibliothèque LIBSVM

C'est une des bibliothèques les plus usitées elle a été développée par Chang et Lin [Chang et Lin, 2010]. Elle contient de nombreux modules de classification et supporte la classification multi-classes ainsi que la validation croisée. Elle est développée en C++ et Java. Des interfaces sont disponibles pour Python, MATLAB, Perl et Ruby. Elle permet l'usage de différents noyaux : linéaire, polynomial ...

C'est cette bibliothèque que l'on retrouve dans des composants de plus haut niveau en traitement d'image tel que la boîte à outil OTB<sup>5</sup> [CNES, 2010] du CNES.

<sup>1</sup>« Visualization and analysis of multi-dimensional hyperspectral images in Astrophysics », visualisation et analyse d'image hyperspectrale en Astrophysique.

<sup>2</sup>Agence Nationale de la Recherche.

<sup>3</sup>Les données sont acquises par des satellites ou des sondes spatiales.

<sup>4</sup>Les simulations numériques fournissent aussi de nombreuses données hyperspectrales.

<sup>5</sup>« Orfeo Tool Box », boîte à outils Orfeo.

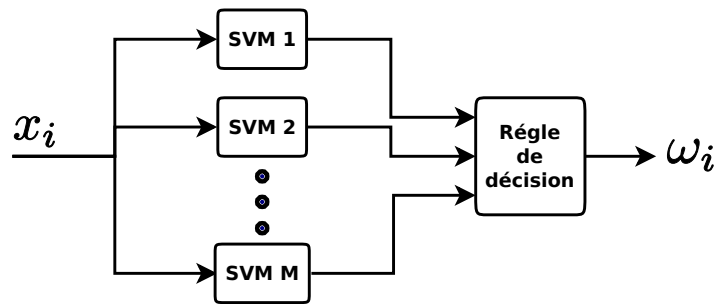


FIG. 4.1 – Architecture parallèle de classifieur SVM binaire.

Pour la résolution du problème dual, cette librairie utilise une méthode de décomposition [Bottou et Chih-Jen, 2007]. Celle-ci ne cherche pas à optimiser l'ensemble des coefficients  $\alpha$  mais à optimiser un sous ensemble  $\alpha_i, i \in \mathcal{B}$  de coefficients à chaque itération en laissant les autres coefficients  $\alpha_j, j \notin \mathcal{B}$  inchangés. LIBSVM utilise une optimisation appelée « optimisation séquentielle minimale » ou SMO. On retrouvera le détail de l'algorithme dans l'article de L. Bottou et C-J. Lin [Bottou et Chih-Jen, 2007].

### 4.3 Volume et taille des données

Les données issues des spectro-imageurs et des simulations numériques représentent de grands volumes de données (plusieurs To). Chaque image hyperspectrale fait plusieurs dizaines de mégaoctets. Le nombre de bandes spectrales étant d'environ 200, la taille d'une matrice servant au calcul d'une fonction noyau est de l'ordre de 3,2 Go pour une centaine d'échantillon. Pour gérer ce problème l'implémentation de LIBSVM utilise un système de cache pour une partie de la matrice noyau  $K$ . Celle-ci est définie par :

$$K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j), \forall i, j \in [1, p].$$

A tout moment seule une partie des coefficients  $K_{i,j}$  est mémorisée. Les coefficients manquants sont calculés à la demande et mémorisés. Les coefficients en mémoire qui n'ont pas été utilisés récemment sont désalloués et en outre, un codage dit « sparse » des données est utilisé. Celui-ci ignore les colonnes comportant la valeur 0 et permet de réduire l'empreinte mémoire.

### 4.4 Classification multi-classes

Les machines à vecteurs support vu précédemment effectuent une classification binaire. Dans la plupart des contextes dont le projet Vahiné nous cherchons à résoudre des problèmes multi-classes. Plusieurs méthodes ont été développées, elles se regroupent en deux approches : les approches parallèles et hiérarchiques. Ces dernières étant moins performantes, nous aborderons seulement les approches parallèles [Melgani et Bruzzone, 2004].

#### 4.4.1 Un contre tous

Cette approche utilise une architecture parallèle de  $k$  machines à vecteurs support, une pour chaque classe. Chaque machine à vecteurs support résout un problème à deux classes : une classe  $\omega_i$  ( $\omega_i \in \Omega$ ) et toutes les autres  $\Omega - \omega_i$ . Sur la figure de principe 4.1 nous avons  $M$  classifieurs binaires avec dans cette première approche  $M = k$ . La règle de décision finale est l'application du principe « winner take all ». Pour chaque classifieur un score va être établi et l'étiquette attribuée à

l'entrée  $x_i$  est celle associée au classifieur qui obtient le meilleur score. Son principal inconvénient est d'effectuer un apprentissage qui peut être grandement déséquilibré. Ainsi pour un classifieur  $\omega_i$  nous pouvons avoir un très petit nombre d'exemples de la classe  $i$  et un grand nombre de contre exemples.

#### 4.4.2 Un contre un

Cette approche utilise  $k(k-1)/2$  classifieurs où chaque SVM est entraîné pour départager deux classes  $\omega_i$  et  $\omega_j$ . On construit ainsi autant de classifieurs que de couples de classe  $(\omega_i, \omega_j)$ ,  $i \neq j$  possibles. Nous avons donc de nouveau l'architecture présentée à la figure 4.1 mais avec  $M = k(k-1)/2$ . La règle de décision s'obtient en utilisant la méthode du vote majoritaire. Chaque classifieur va incrémenter le score  $S_i$  (respectivement  $S_j$ ) associé à la classe  $\omega_i$  (respectivement  $\omega_j$ ). Ensuite le score  $S_i(x)$  le plus élevé va permettre d'attribuer l'étiquette  $i$  à l'entrée  $x$ . Certaines implémentations ajoutent une pondération au vote de chaque classifieur.

Dans [Hsu et Lin, 2002] les auteurs montrent que la méthode « un contre un » a une meilleure précision que la méthode « un contre tous » dans 60% des cas, mais dans toutes les comparaisons les taux de précision reste proche à 2%. Même si la différence de précision est faible un argument plus important en faveur de la stratégie « un contre un » est le temps nécessaire à l'apprentissage. Cette dernière est de 2 à 6 fois plus rapide que la méthode « un contre tous ». Chaque classifieur de la méthode « un contre un » est entraîné avec beaucoup moins de données que la méthode « un contre tous ». La stratégie « un contre tous » a certes beaucoup moins de classifieurs à entraîner, mais dû à la complexité des SVM elle s'avère nettement moins rapide.

#### 4.4.3 Parallélisation

Plusieurs approches de parallélisation dans le cas multi-classes ont été testées. L'approche naïve de distribuer les  $M$  classifieurs sur  $M$  processeur est inefficace. Les temps de calcul entre chaque classifieur ne seront pas identiques car ils ne disposeront pas, sauf hasard, du même nombre d'exemples dans leur jeu d'apprentissage. Finalement les  $M - 1$  processeurs attendront le dernier. Une approche assez simple pour contourner cela est de déployer une architecture « maître esclave » telle que décrite dans l'article [Plaza *et al.*, 2009]. Cette approche consiste à désigner un processeur ou nœud de calcul qui exécutera une tâche principale dite maître. Celle-ci va distribuer chaque tâche d'une liste constituée par l'ensemble des  $k(k-1)/2$  paires de classe  $(\omega_i, \omega_j)$ . Le processus maître va assigner les  $k - 1$  premières tâches à un processeur (ou nœud de calcul). Quand un processeur esclave aura fini son calcul le maître lui enverra la prochaine tâche de la liste jusqu'à épuisement de celle-ci. Cette approche permet d'avoir un taux d'occupation des processeurs esclaves optimal. D'autres approches plus complexes sont également abordées dans [Plaza *et al.*, 2009].

### 4.5 Choix des hyperparamètres

Nous avons vu dans les chapitres précédents différentes méthodes pour le choix des hyperparamètres. Nous revenons ici sur la fonction d'évaluation utilisée pour les algorithmes génétiques et l'aspect multi-classes.

#### 4.5.1 Fonction d'évaluation SV

La fonction d'évaluation notée SV est très simple puisqu'il s'agit du nombre de vecteurs support associé au modèle courant. Ce nombre est normalisé par le nombre total d'exemples traités par l'algorithme. L'idée mise en œuvre par cette évaluation est qu'un petit nombre de vecteurs

support définit un système d'apprentissage ayant de très bonne capacité de généralisation. La réduction du nombre de vecteurs support correspond à une simplification du modèle par élimination des points non pertinents ou redondants.

### 4.5.2 Multi-classes

La problématique du choix des hyperparamètres dans le cas multi-classes peut être facilement implémentée avec la méthode des algorithmes génétiques vu au paragraphe 3.4.2. En effet sur un chromosome nous n'allons plus coder un seul couple  $(C, \gamma)$  mais  $M$  couples correspondants aux différents classifieurs de la méthode « un contre un ». Il faudra juste veiller à ce que l'opérateur de croisement coupe les chromosomes entre deux couples d'hyperparamètres.

# Conclusion

Cette étude fut très intéressante pour ma part, mais assez difficile du fait de l'aspect mathématique. Pour conclure je me focaliserai sur les éléments marquants de mes lectures.

Compte-tenu de la grande dimensionnalité en imagerie hyperspectrale, l'utilisation d'une méthode pour réduire le nombre de spectels utiles est pertinente. Un apparié physique permet de justifier cela. Les terrains étant inhomogènes du point de vue de leurs compositions, chaque pixel est un mélange de spectres. Pour distinguer les molécules de  $\text{CO}_2$  et de  $\text{H}_2\text{O}$  sur un spectre les physiciens utilisent leurs bandes d'absorption spécifique (triplet de l'eau<sup>†</sup> par exemple). Celles-ci se situent sur un nombre restreint de spectels. Intuitivement, si notre problème est de détecter les molécules de  $\text{CO}_2$  et de  $\text{H}_2\text{O}$ , on peut supposer que l'ensemble des spectels peut être réduit. Le problème est de fixer le nombre minimal de spectels nécessaire à la reconnaissance des corps chimiques recherchés. Soit on se base sur la connaissance physique spectroscopique, soit on détermine numériquement le nombre optimum en effectuant une comparaison sur une base de tests.

Le deuxième point concerne les algorithmes génétiques. Leur grand intérêt est leur parallélisme implicite [Goldberg, 1994]. Par exemple, l'algorithme vu précédemment optimise à la fois l'espace des composantes (réduction du nombre de spectels) et les hyperparamètres. Mais on pourrait envisager dans un premier temps d'optimiser uniquement les hyperparamètres des classifieurs multi-classes. Le point négatif dans l'état actuel de la théorie des algorithmes génétiques est qu'il n'existe aucune garantie de découvrir la solution optimale. Une conséquence indirecte est leur temps de calcul assez long pour atteindre un résultat acceptable.

Enfin si la mise en œuvre d'un algorithme de SVM est en général peu coûteuse en temps grâce aux outils disponibles, il faut cependant noter que la recherche des meilleurs paramètres peut requérir des phases de test assez longues. Afin de diminuer le temps de traitement, la parallélisation me semble un point clef. Elle peut intervenir à plusieurs endroits : le parcours d'une grille de recherche, dans le cadre d'un algorithme génétique et surtout au niveau de l'implémentation multi-classes. Cette mise en œuvre du parallélisme est aussi importante du fait de la disponibilité désormais courante de machine quadri-cœur voir octo-cœur.

De ces trois points ci-dessus, il serait intéressant dans le cadre du projet Vahiné de les exploiter pour implémenter un SVM multi-classes à l'aide de la bibliothèque OTB.

Je conclus par le fait que les SVM peuvent également s'utiliser pour d'autres tâches que la classification comme des tâches de régression, c'est-à-dire de prédiction d'une variable continue en fonction d'autres variables. Le champ d'usage des SVM est donc vaste. En résumé les machines à vecteur de support constituent une technique d'apprentissage se développant depuis plus de 10 ans mais bien loin d'avoir atteint leurs limites.



# Glossaire

**Classification** : voir classification non-supervisée ou classification supervisée.

**Classification non-supervisée** : ne nécessitant pas d'autres informations que les données brutes.

**Classification supervisée** : classification qui nécessite des connaissances préalables.

**Condition de Mercer** : permet d'obtenir sous certaines conditions que si  $K(x,y)$  est un noyau défini positif il existe un espace  $\mathcal{F}$  et une fonction  $\phi$  tel que  $K(x,y) = \phi(x) \cdot \phi(y)$ .

**Distance** : ressemblance entre deux objets au sens mathématique.

**Fonction de perte** : une fonction de perte mesure le coût associé à une erreur de prévision, c'est-à-dire la différence entre la valeur prédite et la vraie valeur d'une variable. Il existe plusieurs formes de fonction de perte pour prendre en compte différentes caractéristiques, telles l'importance donnée aux valeurs extrêmes.

**Gradient conjugué** : la méthode du gradient conjugué est une amélioration de la méthode d'optimisation sans contrainte du gradient simple.

**Pixel** : élément discret de la dimension spatiale.

**Similitude** : ressemblance entre deux classes. En classification on utilise des indices de similarité, tel l'indice « Jaccard/Tanimoto » qui utilise ou non les proportions de caractéristique commune entre chaque classe [Bisson, 2009].

**Spectel** : élément discret de la dimension spectrale.

**Spectre** : flux d'énergie électro-magnétique décomposé en fréquences.

**Spectre de référence** : spectre représentatif d'un type de terrain (caractérisé par la présence d'une espèce chimique ou d'un mélange particulier). Le spectre de référence est au centre (en terme de distance<sup>†</sup>) de sa classe.

**Triplet de l'eau** : bandes d'absorption situées à 1.5  $\mu\text{m}$ , 2  $\mu\text{m}$  et 3  $\mu\text{m}$  permettant d'identifier la présence de la molécule  $\text{H}_2\text{O}$ .

# Index

- évaluation (fonction d'), 16, 21
- apprentissage, 3, 4, 13
- biomimétique, 15
- chromosome, 16
- classe, 3, 4
- classification, 2, 3
- complexité, 2, 4
- contingence (table de), 13
- contrainte, 9
- croisement, 15
- dimensionnalité, 15, 17
- distance, 4, 8
- duale, 8–10, 12
- erreur, 13
- génétique (algorithme), 15, 23
- génétique (opérateur), 15
- grille, 14
- hyperplan, 7, 9, 10
- hyperspectrale, 1, 2, 19, 20, 23
- imagerie, 1
- Lagrange (multiplicateurs), 8, 10
- machine à vecteurs support, 1, 4, 7
- marge, 4, 7, 10
- multi-classe, 20, 22, 23
- multi-spectrale, 1
- mutation, 15
- noyau, 9, 20
- parallélisation, 21, 23
- paramètre, 13, 17
- planétologie, 1, 19
- précision, 13
- primale, 8, 12
- régularisation, 10
- ressort (variable), 10
- similitude, 4
- spectel, 1
- spectre, 4
- spectro-imageur, 1, 20
- supervisé, 2, 3
- SVM, 1, 4
- télé-détection, 1, 3
- validation (croisée), 14
- vecteurs support, 7, 9

# Bibliographie

- [Appéré, 2010] APPÉRÉ, T. (2010). Spring evolution of mars northern seasonal condensates from omega/mars express. *Icarus*.
- [Bazi et Melgani, 2006] BAZI, Y. et MELGANI, F. (2006). Toward an optimal svm classification system for hyperspectral remote sensing images. *IEEE Transactions on geoscience and remote sensing*, 44:3374–3385.
- [Bisson, 2009] BISSON, G. (2009). Intelligence artificielle.
- [Bottou et Chih-Jen, 2007] BOTTOU, L. et CHIH-JEN, L. (2007). *Support Vector Machine Solvers, in Large Scale Kernel Machines*. MIT Press.
- [Bousquet, 2001] BOUSQUET, O. (2001). Introduction aux support vector machines.
- [Chang et Lin, 2010] CHANG, C.-C. et LIN, C.-J. (2010). Library for support vector machines. National Taiwan University. URL <http://www.csie.ntu.edu.tw/~cjlin/libsvm/LIBSVM>.
- [Chapelle, 2007] CHAPELLE, O. (2007). Training a support vector machine in the primal. *Neural computation*, 19(5):1155–1178.
- [CNES, 2009] CNES (2009). Système dual d’observation optique de résolution métrique. URL <http://smsc.cnes.fr/PLEIADES/Fr/>.
- [CNES, 2010] CNES (2010). Orfeo toolbox (otb). URL [http://smsc.cnes.fr/PLEIADES/Fr/lien3\\_vm.htm](http://smsc.cnes.fr/PLEIADES/Fr/lien3_vm.htm).
- [Cornuéjols *et al.*, 2002] CORNUÉJOLS, A., MICLET, L. et KODRATOFF, Y. (2002). *Apprentissage artificiel*. Eyrolles.
- [Cortes et Vapnik, 1995] CORTES, C. et VAPNIK, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.
- [ESA, 2009] ESA (2009). Mars express orbiter instruments. URL [http://www.esa.int/SPECIALS/Mars\\_Express/SEMUC75V9ED\\_0.html](http://www.esa.int/SPECIALS/Mars_Express/SEMUC75V9ED_0.html).
- [Ghoggali *et al.*, 2009] GHOGGALI, N., MELGANI, F. et BAZI, Y. (2009). A multiobjective genetic svm approach for classification problem with limited training samples. *IEEE Transactions on geoscience and remote sensing*, 47:1707–1718.
- [Goldberg, 1994] GOLDBERG, D. (1994). *Algorithmes génétiques. Exploration, optimisation et apprentissage automatique*. Addison-Wesley.
- [Holland, 1975] HOLLAND, J. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press.
- [Hsu *et al.*, 2009] HSU, C.-W., CHANG, C.-C. et LIN, C.-J. (2009). A practical guide to support vector classification. Rapport technique, National Taiwan University.

- [Hsu et Lin, 2002] HSU, C.-W. et LIN, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on neural networks*, 13:415–425.
- [INRIA et al., 2008] INRIA, LPG et GIPSA-LAB (2008). Vahine : Visualization and analysis of multi-dimensional hyperspectral images in astrophysics. URL <http://mistis.inrialpes.fr/vahine>.
- [Melgani et Bruzzone, 2004] MELGANI, F. et BRUZZONE, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(8):1778–1790.
- [Plaza et al., 2009] PLAZA, A., BENEDIKTSSON, J. A., BOARDMAN, J. W., BRAZILE, J., BRUZZONE, L., CAMPS-VALLS, G., CHANUSSOT, J., FAUVEL, M., GAMBA, P., GUALTIERI, A., MARCONCINI, M., TILTON, J. C. et TRIANNI, G. (2009). Recent advances in techniques for hyperspectral image processing. *Remote sensing of environment*, 113.
- [Rennard, 2002] RENNARD, J.-P. (2002). *Vie artificielle*. Vuibert informatique.
- [Vapnik, 1995] VAPNIK, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.



Les machines à vecteurs de support pour  
la classification en imagerie hyperspectrale :  
implémentation et mise en œuvre.

Ludovic Mercier

Grenoble le 11 février 2010

---

RÉSUMÉ

Les machines à vecteurs support (SVM) développées dans les années 1990 ont été utilisées dans de très nombreux domaines. Avec le récent développement des capteurs spectrométriques de haute résolution spatiale (quelques dizaines de centimètres par pixel) et spectrale (plusieurs centaines de bandes), les SVM peuvent s'avérer de précieux outils pour le traitement des images hyperspectrales. Ce document a pour but de montrer le fonctionnement des SVM et leur usage comme classifieur dans le cadre de l'imagerie hyperspectrale. Plusieurs techniques de sélection des hyper-paramètres sont abordées. Nous détaillons une technique particulière qui utilise les algorithmes génétiques.

La mise en œuvre sur des données réelles passe par l'usage de bibliothèques type LIBSVM et par la possibilité de faire de la classification multiple. Un algorithme SVM ne pouvant séparer que deux classes, il est alors nécessaire de choisir une stratégie multi-classes adaptée. L'utilisation de SVM multi-classes associées à un algorithme génétique pour la classification d'image hyperspectrale pourrait être alors envisagée dans le cadre du projet de recherche VAHINE.

**MOTS CLEFS :** Algorithme génétique (AG), apprentissage, classification, classification supervisée, classification multi-classes, image hyperspectrale, machines à vecteurs support (SVM), télédétection.

---

ABSTRACT

Support vector machine (SVM) created in 90s is a popular tool for classification problems. Last hyperspectral remote sensors generate large dimensional data because of their very high spatial and spectral resolution. The support vector machine classifier can help us to process these hyperspectral images. Therefore the goal of this document is to explain the mechanism of SVM and its use as classifier. Several methods of model parameters selections are presented. We focus on one of them, the selection by genetic algorithm.

The implementation of SVM typically uses LIBSVM solver to process real data. Our algorithms have to be able to operate in multi-class mode. As the SVMs were originally designed for binary classification, we have to select an efficient multiclass strategy. For VAHINE research project, we assume the usage of a SVM classifier with integrated genetic algorithm for parameter model selection.

**KEY WORD :** Classification, genetic algorithm (GA), hyperspectral image, multiclass classification, remote sensing, supervised classification, support vector machines (SVM).