

# Méthode primale-duale pour l'apprentissage des SVM

**COSI'2014**

Unité de recherche LaMOS,  
Département de Recherche Opérationnelle,  
Université de Bejaia

8-10 Juin 2014

- 1 Introduction
- 2 Classification binaire par SVM
  - Les SVM à marge dure
  - Les SVM à marge souple
  - Les SVM non linéaires
- 3 Méthodes classiques de résolution des problèmes SVM
- 4 Méthode primale-duale pour la résolution des problèmes SVM
  - Définitions
  - Algorithme de résolution par la méthode primale-duale
- 5 Expérimentations numériques
- 6 Conclusion et perspectives



## Introduction

Les Séparateurs à Vaste Marge ou Support Vector Machines(SVM), c'est une méthode basée sur le principe de minimisation du risque décrit dans la Théorie de l'Apprentissage Statistique de Vapnik en 1995.

Les SVM ont montré d'excellentes performances pour une large classe d'applications, telles que :

- La reconnaissance de caractères et d'objets ;
- La détection de face sur des images ;
- Le diagnostic médical ;
- Détection d'intrusion(sécurité informatique) ;
- La biologie ;
- L'analyse des bases de données, etc.



# Classification binaire par SVM

## L'idée des SVMs

Trouver, à base de notions de vaste marge et de fonctions noyau, un hyperplan qui sépare mieux les individus en deux classes.

- ★ L'algorithme SVM permet de prédire la classe d'appartenance de nouvelles entrées :

$$f_{\alpha} \in F : \mathbb{R}^p \rightarrow \{-1, +1\} \quad (1)$$

$$f_{\alpha}(x) = y$$



# Classification binaire par SVM

Etant donné un échantillon d'apprentissage

$S = \{(x_i, y_i), i \in I = \{1, \dots, n\}\}$  où :

- $x_i \in R^p$  est le  $i^{\text{ème}}$  exemple de l'échantillon ;
- $y_i \in \{-1, +1\}$  la classe du  $i^{\text{ème}}$  exemple ;
- $n$  est la taille de l'échantillon ;
- $p$  est le nombre d'attributs.



## Les SVM à marge dure

Les exemples de l'ensemble d'apprentissage sont linéairement séparables en deux classes par un hyperplan  $H$  :

$$H(x) = w^T x + b \quad (2)$$

$H$  satisfait les conditions

$$w^T x_i + b \geq +1, \forall x_i \in \text{Classe } +1$$

$$w^T x_i + b \leq -1, \forall x_i \in \text{Classe } -1$$

équivalent à

$$y_i(w^T x_i + b) \geq 1, \quad \forall i = \overline{1, n} \quad (3)$$



## Les SVM à marge dure

Distance d'un point à l'hyperplan :

$$D(x) = \frac{|w^T x + b|}{\|w\|} \quad (4)$$

La **marge maximale** entre les frontières de (+1 et -1) :

$$m = \frac{2}{\|w\|} \quad (5)$$

Maximiser  $m$  revient à minimiser  $\|w\|$  tout en préservant le pouvoir de classification :

$$\left\{ \begin{array}{l} \text{Min} \quad \frac{1}{2} \|w\|^2 \quad ; \\ \text{s.c} \quad y_i (w^T x_i + b) \geq 1 \quad i = \overline{1, n}. \end{array} \right. \quad (6)$$



## Les SVM à marge dure

La formulation duale :

$$\left\{ \begin{array}{l} \text{Max} \quad L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad ; \\ \text{s.c} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad ; \\ \quad \quad \alpha_i \geq 0 \quad . \end{array} \right. \quad (7)$$

où  $\alpha \in \mathbb{R}^n$  est le vecteur des multiplicateurs de Lagrange associés aux contraintes du problème primal.



## Les SVM à marge souple

- ★ Les données d'apprentissage sont linéaires et peuvent être bruitées et donc non séparables.
- ★ En 1995, C.Cortes et V.Vapnik ont introduit des **variables d'écart** pour relâcher sensiblement les contraintes sur la marge.

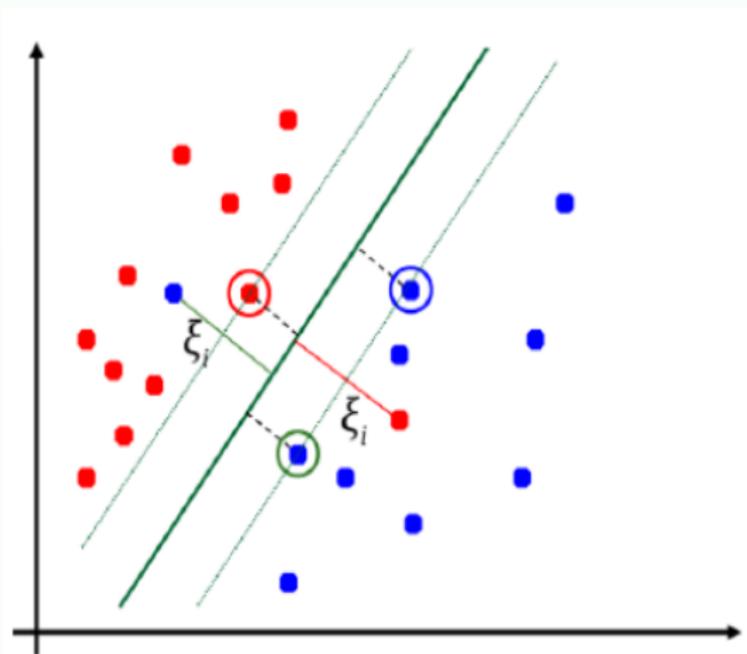
$$y_i f(x_i) \geq 1 - \xi_i \quad (8)$$

Le problème d'optimisation devient alors :

$$\left\{ \begin{array}{l} \text{Min} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad ; \\ \text{s.c} \quad y_i (w^T x_i + b) \geq 1 - \xi_i \quad ; \\ \quad \quad \xi_i \geq 0 \quad , \quad i = \overline{1, n}. \end{array} \right. \quad (9)$$



# Les SVM à marge souple





## Les SVM à marge souple

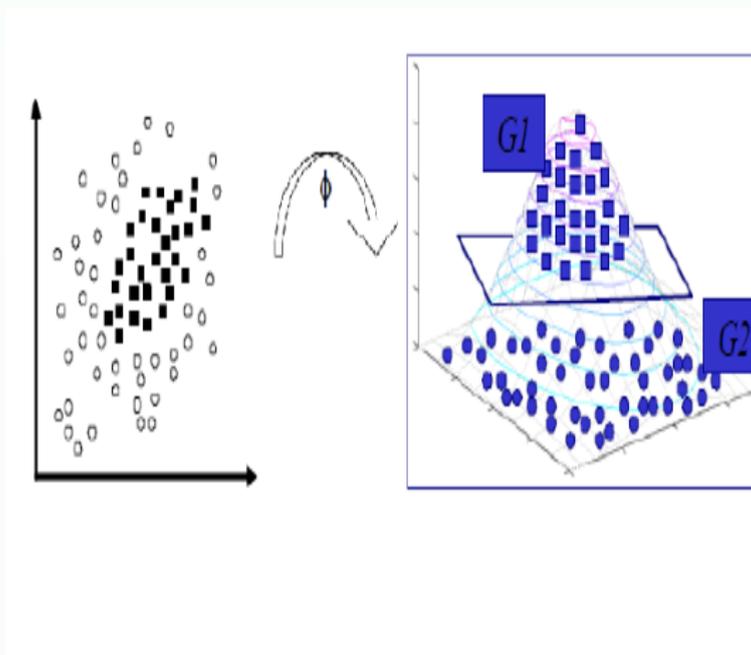
Le problème dual sera :

$$\left\{ \begin{array}{l} \text{Max} \quad L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad ; \\ \text{s.c} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad ; \\ \quad \quad 0 \leq \alpha_i \leq C \quad , \quad i = \overline{1, n}. \end{array} \right. \quad (10)$$

- ◇  $\alpha_i = 0$  : l'exemple d'apprentissage est bien classé mais n'est pas sur un hyperplan canonique ;
- ◇  $0 < \alpha_i < C$  : l'exemple est bien classé ; il s'agit donc d'un **vecteur support** qui servira à construire le classifieur final ;
- ◇  $\alpha_i = C$  : l'exemple est mal classé, mais étant donné qu'il n'est pas nul, il sera tout de même considéré comme **vecteur support** ;



# Les SVM non linéaires





## Les SVM non linéaires

Soit  $\phi$  une transformation non-linéaire qui transforme l'espace d'entrée  $S$  en un espace de redescription  $\phi(S)$

$$S = \{x_1, \dots, x_n\} \rightarrow \phi(S) = \{\phi(x_1), \dots, \phi(x_n)\}$$

Les *fonctions noyau* notées  $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$  sont des fonctions bilinéaires symétriques positives.

Le problème d'optimisation :

$$\left\{ \begin{array}{l} \text{Max } L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j); \\ \text{s.c } \quad \sum_{i=1}^n \alpha_i y_i = 0; \\ \quad \quad 0 \leq \alpha_i \leq C \quad , \quad i = \overline{1, n}. \end{array} \right. \quad (11)$$



## Les SVM non linéaires

Noyaux utilisés par les SVM :

- Noyau linéaire :  $K(x_i, x_j) = x_i^T x_j$
- Noyau polynomial :  $K(x_i, x_j) = (x_i^T x_j)^d$
- Noyau RBF :  $K(x_i, x_j) = e^{-\gamma(\|x_i - x_j\|)^2}$

### Remarque

Vue la possibilité d'une grande dimension du vecteur  $w \in F$ , on résout généralement le problème dual :

$$\left\{ \begin{array}{l} \max \quad L(\alpha) = -\frac{1}{2}\alpha^T D\alpha + e^T \alpha; \\ s.c \quad y^T \alpha = 0; \\ \quad \quad 0 \leq \alpha \leq Ce. \end{array} \right. \quad (12)$$





## Paramètres de l'hyperplan optimal ( $H$ )

- Soit  $\alpha^*$  la solution optimale du problème dual.
- Vecteurs supports :  $SV = \{i | \alpha_i^* > 0\}$
- Le vecteur poids :  $w^* = \sum_{i \in SV} \alpha_i^* y_i x_i$
- Le biais :  $b^* = \frac{1}{|I_{vs}|} \sum_{i \in I_{vs}} [y_i - w^{*'} \phi(x_i)]$ .
- La règle de classification d'une nouvelle observation  $x$  :

$$f(x) = \text{sign}\left(\sum_{i \in SV} \alpha_i^* y_i K(x_i, x) + b^*\right) \quad (13)$$

# Méthodes classiques de résolution des problèmes SVM

La classification SVM se réduit à un problème d'optimisation quadratique. Il existe diverses techniques permettant de résoudre ce type de problème, les mieux adaptées sont :

- ◇ Les méthodes de décomposition ;
- ◇ Les méthodes d'activation des contraintes ;

En pratique : la taille des échantillons  $n$  est très grand, alors  $D$  est de grande dimension, dense et mal-conditionnée ; Cela représente un challenge pour ces approches.



## Méthodes classiques de résolution des problèmes SVM

Le principe général de ces méthodes repose sur le calcul des multiplicateurs de Lagrange associés aux seuls vecteurs de support. La solution peut donc être obtenue très rapidement si l'on sait réduire la taille du problème à l'ensemble restreint des vecteurs de support :

- ◇  $[I_S]$  le groupe des points support (non-contraints), c-à-d des vecteurs supports candidats ( $0 < \alpha_i < C$ ) ;
- ◇  $[I_C]$  le groupe des points saturés (contraints à C), c-à-d des vecteurs de support bornés ( $\alpha_i = C$ ) ;
- ◇  $[I_0]$  le groupe des points inactifs (contraints à 0), c-à-d des points bien-classés ( $\alpha_i = 0$ ) .



# Méthodes classiques de résolution des problèmes SVM

Ces méthodes se distinguent par la façon de faire cette décomposition mais utilisent des algorithmes itératifs pour le calcul des multiplicateurs  $\alpha$ , et ces algorithmes ont une structure générale commune :

- 1 Initialisation
- 2 tant que la solution courante n'est pas optimale faire
  - mettre à jour la répartition des groupes
  - calculer les coefficients  $\alpha_j$  correspondant aux changements
- 3 fin tant que



## Méthode du Chunking

Décrite pour la première fois par Vapnik [1982], élimine les points  $\alpha_j = 0$  au fur et à mesure de l'avancement de l'algorithme.

## Méthode de décomposition successive

La technique de Osuna[1997] est similaire au chunking mais sur des sous-ensembles de taille fixe auxquels il est possible d'ajouter ou de retirer un exemple à chaque fois.

## Méthode SMO (Sequential Minimal Optimization)

Proposée par Platt [1999],résout à chaque itération un PQ à 2 variables dont la solution est calculée analytiquement.



# Méthode primale-duale pour la résolution des problèmes SVM

- La méthode proposée est une extension de la **méthode adaptée**, développée par R.Gabasov [1], et des travaux de B.Brahmi et M.O.Bibi [2, 3], pour la résolution des problèmes de classification binaire par SVM.
- Cette approche est composée de deux phases :
  - La phase 1 est basée sur une procédure de la méthode de points intérieurs pour construire un support initial non arbitraire.
  - La deuxième phase est itérative et calcule une solution optimale du problème.

## Formulation duale des SVM pour la classification

$$\left\{ \begin{array}{ll} \min & L(\alpha) = \frac{1}{2}\alpha^T D\alpha - e^T \alpha; \\ \text{s.c} & y^T \alpha = 0; \\ & 0 \leq \alpha \leq Ce. \end{array} \right. \quad (14)$$

$D$  : matrice d'ordre  $n$  symétrique semi-définie positive, telle que :

- $d_{ij} = y_i y_j k(x_i, x_j)$ ,  $\alpha \in R^n$ ,  $i \in I = \{1, \dots, n\}$
- $y = y(I) = (y_i, i \in I)$ ,  $e = (1, \dots, 1) \in R^n$
- $I_B \subset I$  tel que  $|I_B| = 1$  est appelé *support*,
- $I = I_B \cup I_N$ ,  $I_N = I_S \cup I_{NN}$ ,  $I_B \cap I_N = \emptyset$ ,  $I_S \cap I_{NN} = \emptyset$ .



## Définitions

- Un vecteur  $\alpha^0$  vérifiant les contraintes du problème (14) est appelé *solution réalisable* de ce problème.
- Une solution réalisable  $\hat{\alpha}$  est dite *optimale* si  $L(\hat{\alpha}) = \min L(\alpha)$ , où  $\alpha$  est pris parmi toutes les solutions réalisables du problème (14).
- Un vecteur  $\alpha^\epsilon$  est appelé  *$\epsilon$ -optimal* ou *suboptimal* si  $L(\alpha^\epsilon) - L(\hat{\alpha}) \leq \epsilon$ , où  $\hat{\alpha}$  est une solution optimale du problème (14) et  $\epsilon$  un nombre positif ou nul, donné à l'avance.
- L'ensemble  $I_P = \{I_B, I_S\}$  est appelé *support du problème* (14).
- Le couple  $\{\alpha, I_P\}$  est appelé *solution réalisable de support* (SRS) du problème (14). Elle est dite potentiellement *non dégénérée* si  $0 < \alpha_j < C, i \in I_B$ . Elle est dite *accordée* si  $E_S = 0$ .



## Définitions

- Le vecteur  $g(\alpha) = D\alpha - e$  est le gradient de la fonction objectif ;
- Le scalaire  $u = g_{BYB}$  est appelé potentiel ;
- Le vecteur  $E = g - uy$  est le vecteur des estimations ;

## Critère d'optimalité

Soit  $(\alpha, l_P)$  une **SRS accordée** du problème (14) . Alors les relations

$$\left\{ \begin{array}{l} E_i \geq 0, \quad \text{si } \alpha_i = 0 \quad ; \\ E_i \leq 0, \quad \text{si } \alpha_i = C \quad ; \\ E_i = 0, \quad \text{si } 0 < \alpha_i < C \quad , i \in I_{NN} \quad , \end{array} \right. \quad (15)$$

sont **suffisantes**, et dans le cas de la non dégénérescence de  $(\alpha, l_P)$ , aussi **nécessaires** pour l'optimalité de la solution réalisable  $\alpha$ .





## Définitions

L'estimation de suboptimalité est :

$$\beta(\alpha, l_P) = \sum_{i \in I_{NN}, E_i > 0} E_i \alpha_i + \sum_{i \in I_{NN}, E_i < 0} E_i (\alpha_i - C) \quad (16)$$

## Critère de suboptimalité

Soit  $(\alpha, l_P)$  une **SRS accordée** du problème (14) et  $\epsilon$  un nombre arbitraire positif ou nul. Si  $\beta(\alpha, l_P) \leq \epsilon$ , alors  $\alpha$  est une **solution suboptimale** de ce problème.



## Phase I

- Déterminer un point intérieur initial  $\alpha^0$  qui vérifie :

$$0 < \alpha^0 < C \quad \text{et} \quad y' \alpha^0 = 0.$$

Pour cela on pose

$$I^+ = \{i \in I : y_i = +1\}, I^- = \{i \in I : y_i = -1\},$$

$$|I^+| = p > 0, |I^-| = n - p > 0.$$

On prend

$$\alpha^0(I^+) = R, \quad \alpha^0(I^-) = S, \quad \text{avec} \quad 0 < R < C \quad \text{et} \quad 0 < S < C.$$

On doit donc avoir

$$Rp - S(n - p) = 0 \Rightarrow R = \frac{S(n - p)}{p} \quad \text{et} \quad S = \frac{Rp}{n - p}.$$



## Algorithme de résolution par la méthode primale-duale

Si  $p > (n - p)$ , on pose

$$\alpha_i^0 = C(n - p)/2p, i \in I^+ \quad \text{et} \quad \alpha_i^0 = C/2, i \in I^-.$$

Si  $p \leq (n - p)$ , on pose

$$\alpha_i^0 = C/2, i \in I^+ \quad \text{et} \quad \alpha_i^0 = Cp/2(n - p), i \in I^-.$$

- Calculer  $\mu = (y' T^2 y)^{-1} y' T^2 g(\alpha^0)$ , avec  $T = \text{diag}(\alpha^0)$ ;
- Calculer  $E_i(\alpha^0) = g_i(\alpha^0) - \mu y_i, \quad i \in I$ ;
- Déterminer  $I^0 = \{i_0 \in I : |E_{i_0}| = \min |E_i|\}$ ,  $i_b \in I^0, I_B^0 = \{i_b\}$ .  
Donc la solution réalisable initiale et le support initial sont respectivement  $\alpha^0$  et  $I_B^0$ .



## Phase II

- Démarrer avec la SRS  $(\alpha, l_P)$ , où  $\alpha = \alpha^0$ ,  $l_B = l_B^0$  et  $l_S = \emptyset$ ;

### (1) Vérification de l'optimalité de la SRS $(\alpha, l_P)$ :

Calculer l'estimation de suboptimalité.

- Si  $\beta(\alpha, l_P) \leq \epsilon$  ou bien les relations d'optimalité (15) sont vérifiées, alors  $\alpha$  est respectivement une solution suboptimale (ou optimale) du problème. Arrêter l'algorithme.
- Sinon Aller à l'étape (2).

### (2) Construction d'une nouvelle solution réalisable :

$$\bar{\alpha} = \alpha + \theta l, \quad \bar{E} = E + \theta t, \quad \bar{u} = u + \theta r \quad \text{avec} \quad t = Dl - ry.$$



## Algorithme de résolution par la méthode primale-duale

Pour cela

- Calculer les matrices  $Z$  et  $M$  telles que :

$$M = M(I_N, I_N) = Z'DZ \text{ et } Z = \begin{pmatrix} -yBY_N \\ J_N \end{pmatrix},$$

$J_N$  étant la matrice identité de dimension  $(n-1)$ .

- Calculer les directions d'amélioration  $l$ ,  $r$  et  $t$ , où

$$l_i = \begin{cases} -\alpha_i, & \text{si } E_i > 0 ; \\ C - \alpha_i, & \text{si } E_i < 0 ; \\ 0, & \text{si } E_i = 0, i \in I_{NN} ; \end{cases} \quad (17)$$

- $l_S$  sera déduit à partir de  $E_S(\bar{\alpha}) = 0$  :

$$l_S = -M_S^{-1}M(I_S, I_{NN})l_{NN} ;$$

- $l_B$  sera déduit à partir de la formule  $y'l = 0$  :

$$l_B = -y_B(y_S l_S + y_{NN} l_{NN}) ;$$



## Algorithme de résolution par la méthode primale-duale

$$\left\{ \begin{array}{l} r = y_B D(I_B, I) l \quad ; \\ t_B = 0 \quad , t_S = 0 \quad ; \\ t_{NN} = D(I_{NN}, I) l - r y_{NN} \quad ; \end{array} \right. \quad (18)$$

- Calculer le pas optimal  $\theta = \min(1, \theta_{i_b}, \theta_{i_s}, \sigma_f)$ , avec  
 $\theta_{i_b} = \theta_i, i \in I_B$  ,  $\theta_{i_s} = \min \theta_i, i \in I_S$  , et  
 $\sigma_f = \sigma_{i_0} = \min \sigma_i, i \in I_{NN}$ , où

$$\theta_i = \left\{ \begin{array}{l} \frac{-\alpha_i}{l_i}, \quad \text{si } l_i < 0 \quad ; \\ \frac{(C - \alpha_i)}{l_i}, \quad \text{si } l_i > 0 \quad ; \\ \infty, \quad \text{si } l_i = 0 \quad , i \in I_B \cup I_S; \end{array} \right. \quad (19)$$



## Algorithme de résolution par la méthode primale-duale

$$\sigma_i = \begin{cases} \frac{-E_i}{t_i}, & \text{si } E_i t_i < 0 ; \\ \infty, & \text{sinon } , i \in I_{NN} . \end{cases} \quad (20)$$

- Si  $\theta = 1$ , alors  $(\bar{\alpha}, l_P)$  est une SRS optimale et on arrête l'algorithme ;
- Sinon, calculer l'estimation de suboptimalité de la nouvelle SRS  $(\bar{\alpha}, l_P)$  :

$$\beta(\bar{\alpha}, l_P) = (1 - \theta)(\beta - \theta t'_{NN} l_{NN}).$$



## Algorithme de résolution par la méthode primale-duale

- Si  $\beta(\bar{\alpha}, l_P) \leq \epsilon$ , alors  $(\bar{\alpha}, l_P)$  est une SRS  $\epsilon$ -optimale et on arrête l'algorithme.
- Sinon, aller à l'étape (3).

### (3) Changement de support :

- Si  $\theta = \theta_{i_s}$ , poser

$$\bar{I}_B = I_B, \quad \bar{I}_S = I_S \setminus i_s, \quad \bar{I}_{NN} = I_{NN} \cup \{i_s\};$$

- Si  $\theta = \sigma_f = \sigma_{i_0}$ , poser

$$\bar{I}_B = I_B, \quad \bar{I}_S = I_S \cup \{i_0\}, \quad \bar{I}_{NN} = I_{NN} \setminus \{i_0\};$$

- Si  $\theta = \theta_{i_b}$  :



## Algorithme de résolution par la méthode primale-duale

✓ Si  $I_S = \emptyset$ , poser

$$\bar{I}_B = \{i_0\}, \quad \bar{I}_S = \emptyset, \quad \bar{I}_{NN} = I_{NN} \cup \{i_b\} \setminus \{i_0\};$$

✓ Si  $I_S \neq \emptyset$ , poser

$$\bar{I}_B = \{i_s\}, \quad \bar{I}_S = I_S \setminus i_s, \quad \bar{I}_{NN} = I_{NN} \cup \{i_b\}.$$

● Poser

$$\alpha = \bar{\alpha}, \quad I_B = \bar{I}_B, \quad I_S = \bar{I}_S, \quad I_{NN} = \bar{I}_{NN}, \quad I_P = \{I_B, I_S\}.$$

● Aller à l'étape (1).



## Expérimentations numériques

### Comparaisons entre SVM-A et SMO

- Implementation : Matlab 7.2, PC portable doté du système d'exploitation Windows 7, CPU=2.13 Ghz, RAM de 4GO ;
- SMO existe dans la toolbox SVM de Matlab ;
- Les Benchmarks utilisés sont pris de la base UCI ;
- $n$  : Taille de l'échantillon d'apprentissage ;
- $p$  : nombre d'attributs.

### Critères de comparaison

- $ns$  : Le nombre de vecteurs supports ;
- $CPU$  : Le temps d'apprentissage en secondes ;
- $nit$  : Le nombre d'itérations ;

○○○○  
○○○  
○○○

○○○○  
○○○○○○○○

La base ( $n \times p$ )	Noyau	C	SMO			SVM-A		
			ns	CPU	nit	ns	CPU	nit
Linsep( $7 \times 2$ )	Linéaire	100	3	1.82	25	4	<b>0.21</b>	5
	Polynomial(3)	100	3	1.80	89	4	<b>0.20</b>	4
	RBF(0.5)	100	7	1.84	27	7	<b>0.23</b>	6
Iris( $256 \times 4$ )	Linéaire	100	3	0.38	14	3	<b>0.22</b>	5
	RBF(1)	100	18	0.41	92	18	<b>0.25</b>	18
	Poly(3)	100	3	0.38	17	3	<b>0.21</b>	12
Ovarian-cancer( $216 \times 4000$ )	Linéaire	100	33	<b>0.71</b>	2831	33	0.73	468
	RBF(0.5)	100	216	<b>1.11</b>	974	216	1.73	1
	Poly(2)	100	34	1.23	3048	34	<b>0.72</b>	184
Heart( $270 \times 13$ )	Linéaire	10	-	-	-	99	<b>2.89</b>	588
	RBF(1)	10	270	1.27	1261	270	<b>0.36</b>	65
	Poly(2)	10	-	-	-	88	<b>1.45</b>	721
Diabetes( $768 \times 8$ )	Linéaire	1000	-	-	-	400	<b>41.52</b>	1520
	RBF(1)	1000	768	2.83	4121	768	<b>1.32</b>	49
German ( $1000 \times 24$ )	linéaire	1000	-	-	-	530	<b>66.96</b>	2627
	RBF(0.2)	1000	1000	<b>2.02</b>	3272	1000	3.08	93
Australian( $690 \times 14$ )	Linéaire	1000	-	-	-	207	<b>45.72</b>	1751
	RBF(0.4)	1000	690	3.10	2324	690	<b>1.52</b>	91
USPS( $1500 \times 256$ )	RBF(1)	100	1390	<b>2.82</b>	5084	1388	102.59	1168

**TAB. 1.** Résultats de comparaison numérique entre SMO et SVM-A sur la base UCI



## Commentaires

- Les deux approches donnent le même nombre de **vecteurs supports**, équivalentes en terme de qualité d'apprentissage ;
- L'efficacité de notre méthode en terme de **nombre d'itérations** ;
- Les deux méthodes sont équivalentes en termes de **temps d'exécution** pour les problèmes de taille moyenne. Cependant, SMO est rapide par rapport à SVM-A pour les problèmes de très grande taille, et cela dû à l'utilisation de la technique du caching dans les calculs par SMO.
- L'algorithme SMO ne converge pas pour des bases lorsque on utilise le noyau linéaire ou polynomial. Par contre, SVM-A converge toujours pour tous les noyaux.



## Conclusion

Dans ce travail, nous avons proposé une nouvelle méthode pour l'apprentissage des SVM composée de deux phases. La première est inspirée d'une procédure de la méthode de points intérieurs et la deuxième est une extension de la méthode adaptée de support. L'algorithme de la méthode est arrêté lorsque le critère d'optimalité ou de suboptimalité est rempli. Notre approche est comparée avec celle de SMO sur la base UCI.

## Perspectives

Comme perspective, nous envisageons de réduire la complexité de notre algorithme pour accélérer l'apprentissage des problèmes SVM de grande taille en appliquant les techniques de Caching et de Chunking utilisées dans SMO.



R. Gabasov, F. M. Kirillova, V. M. Raketky and O.I. Kostyukova.

*Constructive Methods of Optimization.*

Volume 4 : Convex Problems, University Press, Minsk, 1987.



B. Brahmi and M. O. Bibi.

*Dual support method for solving convex quadratic programs.*

Optimization, Vol. 59, No. 6, pp. 851-872, 2010.



B. Brahmi. and M.O. Bibi

*A dual support method for SVM training.*

26 th European conference on operational research, Book of abstracts, Rome, Italy, July 01-04, 2013.



V.N. Vapnik.

*The nature of Statistical Learning Theory.*

Springer-Verlag, 1995.



B. Schölkopf and A. I. Smola.

*Learning With Kernels : Support Vector Machines,  
Regularization, Optimization and Beyond.*

MIT Press, 2002.



I. C. Platt.

Fast training of support vector machines using sequential  
minimal optimization.

In C. I. C. Burges, B. Schölkopf and A. I. Smola, editors,  
*Advances in Kernel Methods : Support Vector Learning.* MIT  
Press, 1999.

**Merci pour votre attention !**