Séparateur à Vastes Marges (SVM) et Méthodes à Noyau pour la Classification de Textes

"Classification pour les textes de corpus"

Ana Karina Fermín

Université Paris Ouest Nanterre, La Défense

Paris, 07/11/2013

- En Data Mining on est de plus en plus confronté à des données structurées notamment textuelles
  - Pages webs, courriels, documets, ...
- Les techniques d'app. statistique (ex. classifieurs textuels) sont devenus des outils indispensables au traitement automatique de textes.

Problème : Ces techniques sont généralement conçues pour travailler sur des données vectorielles. Comment calculer une mesure de similarité entre deux structures ?

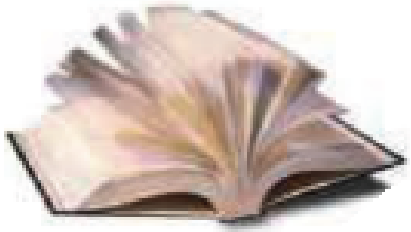
Astuce : Utiliser les fonctions à noyau

$$\longrightarrow \quad K = \begin{bmatrix} k(b_1,b_1) & k(b_1,b_2) & \cdots & k(b_1,b_{100}) \\ k(b_2,b_1) & k(b_2,b_2) & \cdots & k(b_2,b_{100}) \\ \vdots & \vdots & \ddots & \vdots \\ k(b_n,b_1) & k(b_n,b_2) & \cdots & k(b_{100},b_{100}) \end{bmatrix}$$

La classification de texte est une des tâches dans le domaine du Traitement Automatique des Langues (TAL).

Outils : Fonctions Noyau

- Catégorisation de textes
  - Noyau standard "sac de mots" : mots communs aux deux textes
  - Noyaux de sous-chaines : sous-chaines communes aux deux textes
- Catégorisation de relations
  - Noyaux d'arbres



$$\longrightarrow \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

But : utiliser les SVM (méthodes d'apprentissage basées sur des fonctions noyau) pour faire la classification de textes.

Menu du jour

1. Motivation
   - Exemple : Catégorisation de textes
2. SVM
   - Qu'est-ce que c'est ?
   - Comment cela marche-t-il ?
   - Pourquoi est-ce utile ?
3. Noyau pour les textes

Motivation I

La lute contre le Spam

Motivation II

$\mathcal{C}$ (corpus) : ensemble de n e-mails
$\mathcal{D}$ (dictionnaire de mots) : ensemble de p mots

$$\mathcal{D} = \{m_1, m_2, \ldots, m_p\}$$

Nature des données :

- Entrée : e-mail
- Sortie : spam/ non-spam
- Transformation : descripteur du message par "sac de mots "
  Fréquence/Occurrence de mots :
  Un texte t (message) est codé de la façon suivante :

$$x = [x_1 \quad x_2 \quad \ldots \quad x_p]$$

où x_i est la fréquence de m_i dans t.

Motivation III

Nature des données :

- X : Matrice des données (de taille $n \times p$)
  - Ligne i : $i$ème de l'email
  - Colonne j : fréquences du mot j
- Y : spam vs. non-spam

Motivation IV

Spam

WINNING NOTIFICATION
We are pleased to inform you of the result of the Lottery Winners International programs held on the 30th january 2005. [...] You have been approved for a lump sum pay out of 175,000.00 euros. CONGRATULATIONS ! ! !

No Spam

Dear George,
Could you please send me the report #1248 on the project advancement ? Thanks in advance.

Regards,
Cathia

Base de données de courriel identifiés ou non comme spam

Occurrences de mots "George", "send", "Lottery", "project", "pay", "euros", "NOTIFICATION", "CONGRATULATIONS", "!", report, ...

Motivation V

Tex Mining

- Initialement, chaque message du corpus $\mathcal{C}$ (ensemble de emails) subit un prétraitement qui permet d'éliminer les articles, prépositions.
- Le statisticien observe des mots nettoyés ("cleaned up" words) et ses fréquences.

No Spam

Dear George,
Could you please send me the report #1248 on the project advancement ? Thanks in advance.

Regards,
Cathia

project	report	advance	#
1	1	2	1

Motivation VI

Dictionnaire de mots : $\mathcal{D} = \{m_1, m_2, \ldots, m_p\}$

address	will	font	857	meeting
all	people	000	data	original
3d	report	money	415	project
our	addresses	hp	85	re
over	free	hpl	technology	edu
remove	business	george	1999	table
internet	email	650	parts	conference
order	you	lab	pm	!
mail	credit	labs	direct	(
receive	your	telnet	cs	#

Données Spam

Echantillon de taille $n = 4601$ (57 variables et une variable réponse)

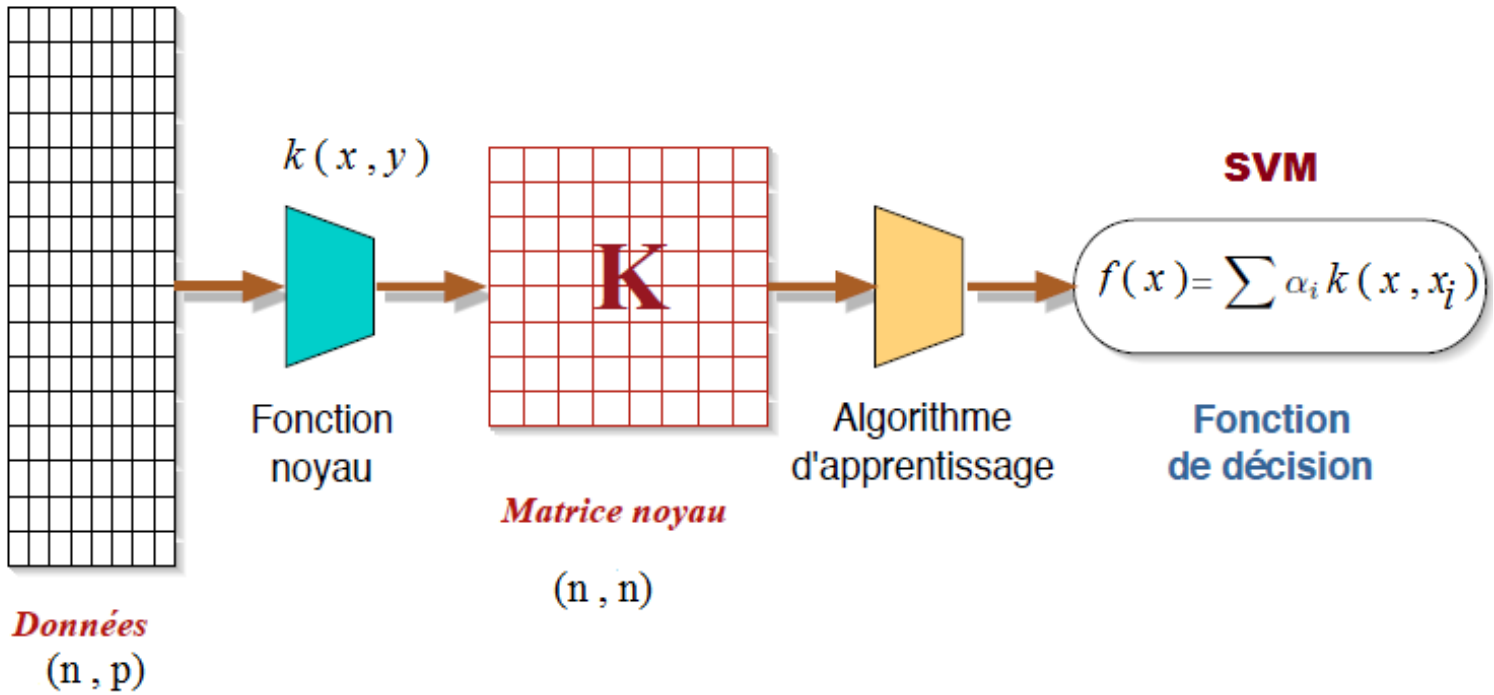
```
'data.frame':   4601 obs. of  58 variables:
 $ make            : num  0.00 0.21 0.06 0.00 0.00  ...
 $ address         : num  0.64 0.28 0.00 0.00 0.00  ...
 $ all             : num  0.64 0.50 0.71 0.00 0.00  ...
 $ num3d           : num  0.00 0.00 0.00 0.00 0.00  ...
 $ our             : num  0.32 0.14 1.23 0.63 0.63  ...
 $ over            : num  0.00 0.28 0.19 0.00 0.00  ...
 $ remove          : num  0.00 0.21 0.19 0.31 0.31  ...
 $ internet        : num  0.00 0.07 0.12 0.63 0.63  ...
 .
 .
 .
 $ capitalTotal    : num  278  1028 2259 191  191 ...
 $ type            : Fac  spam spam spam spam spam ...
```

Pour le 1er e-mail

```
> spam[1,1:8]
  make address  all  num3d  our   over   remove  internet
   0    0.64   0.64    0    0.32    0       0         0

> spam[1,20:28]
  credit  your  font  num000  money  hp  hpl  george
   0      0.96   0     0       0     0   0     0

> spam[1,52:57]
charExclamation charDollar charHash capitalAve capitalLong
    0.778           0          0       3.756       61

> spam[1,58]
    spam
```

Menu du jour :

- Une introduction aux SVM (classification)
  - Deux classes, linéairement séparables
  - Comment adapter les SVM aux cas non linéaires
    - Fonction noyau k
    - $k(x, y)$ représente intuitivement la similarité entre les x et y.
- Application : classification d'e-mails.



SVM (Formalisme) I

Rappel (pb classification à deux classes)

- $\mathcal{X}$ espace quelconque d'objets
- $\mathcal{Y} = \{-1, 1\}$ (classification).
- Données : On dispose d'un échantillon $S_n = (x_1, y_1), \ldots, (x_n, y_n)$
- But : Construire un classifieur $g : \mathcal{X} \to \{-1, 1\}$ à partir de S_n
- Plutôt que de construire directement g on construit $f : \mathcal{X} \to \mathbb{R}$
- Le classifieur est donné par le signe de f

$$g = \text{sgn}(f)$$

Cas linéaire : Supposons

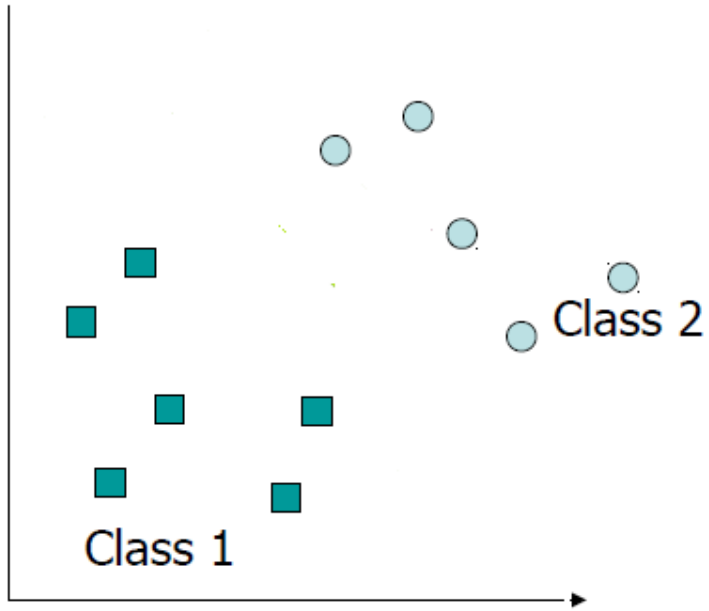
$$f(x) = < w, x > + b = w^T x + b$$

avec w et b inconnus.

SVM (Formalisme) II

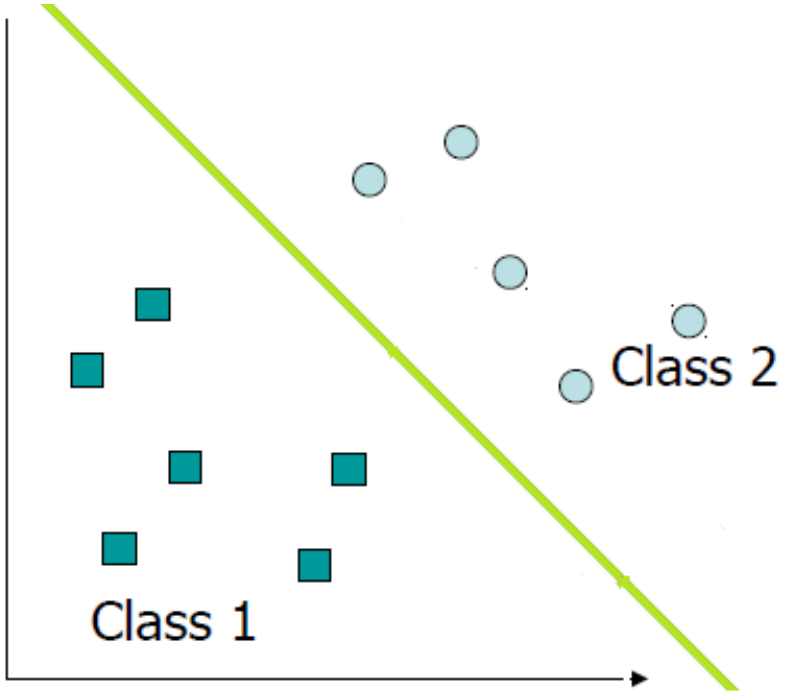
Cas simple : Le séparateur linéaire (dans le plan, ici $p = 2$).

- Affectons l'étiquette $y = +1$ à tous les points de la classe 1 et l'étiquette $y = -1$ à tous les points de la classe 2.



- But 1 : Trouver une frontière de décision (un hyperplan) qui sépare l'espace en deux régions

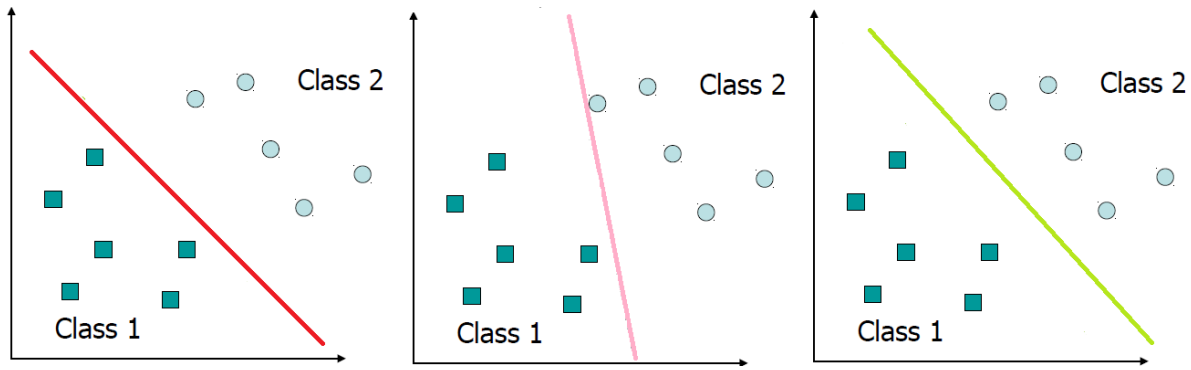
SVM (Formalisme) III



- But 2 : Nous cherchons un classifieur linéaire $g(x)$ qui permettra :
  - d'affecter à la classe 1 tous les points x pour lesquels $g(x) = +1$
  - d'affecter à la classe 2 tous les ponts x pour lesquels $g(x) = -1$

SVM (Formalisme) IV

Il existe en général une infinité d'hyperplans qui permettra de séparer les deux classes (laquelle choisir ?)



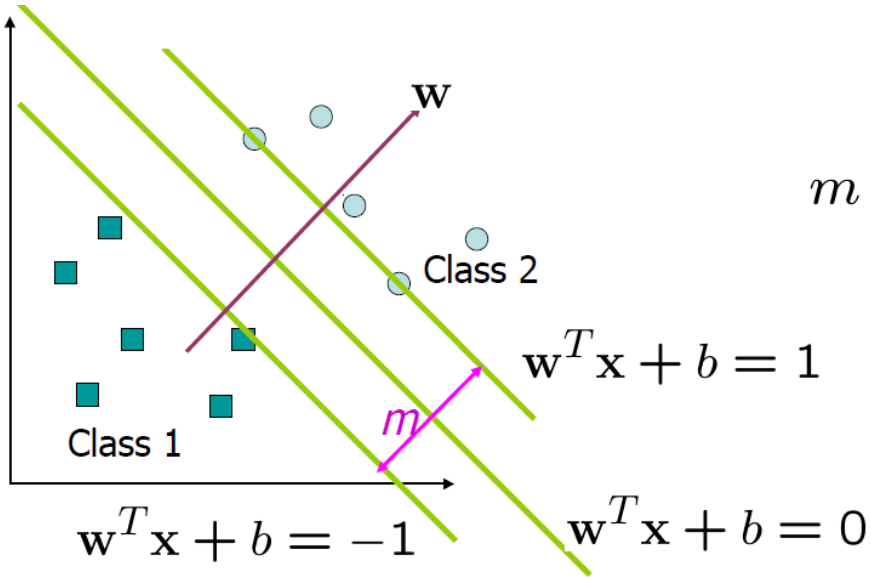
On veux sélectionner celui qui se trouve le plus loin possible de tous les points

Hyperplan Optimal

C'est l'hyperplan à **marge maximale**

SVM (Formalisme) V

- **marge** = distance du point le plus proche la l'hyperplan

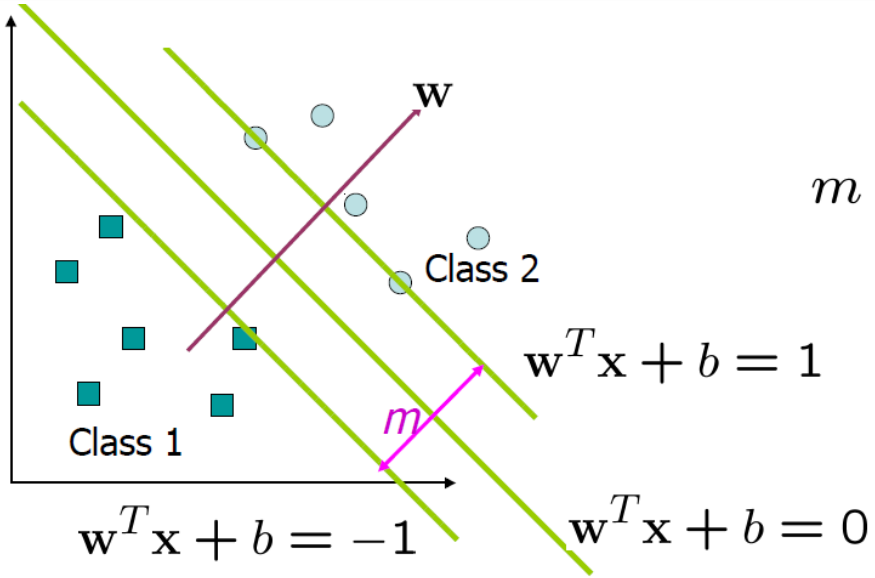


$$m = \frac{2}{||\mathbf{w}||}$$

- La marge normalisé étant égale à

$$\frac{2}{\|w\|^2}$$

SVM (Formalisme) VI



$$m = \frac{2}{||\mathbf{w}||}$$

vecteurs supports

- Seuls les points sur les hyperplans frontières (droite positive et droite négative) jouent un rôle important
- Ces points sont appelés vecteurs support

SVM (Formalisme) VII

A l'aide d'un algorithme d'optimisation quadratique sous contrainte linéaires, on estime w et b.

Forme primale (cas linéaire)

$$\begin{cases} \text{Minimiser} & \frac{1}{2}\|w\|^2 \\ \text{sous les contraintes} & y_i(w^T x_i + b) \geq 1 \quad i = 1, \ldots, n \end{cases}$$

- Ce problème d'optimisation possède une forme duale
- On passe par le Lagrangien

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum \alpha_i (y_i(w^T x_i + b) - 1)$$

SVM (Formalisme) VIII

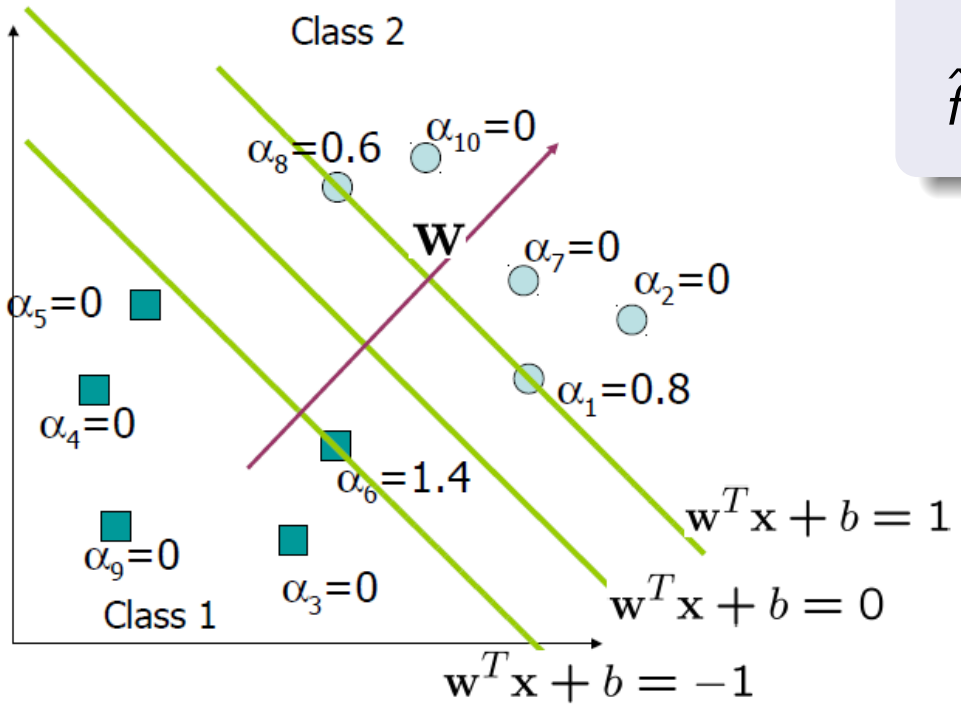
- La solution du Lagrangien

$$\sum \alpha_i y_i = 0 \qquad \hat{w} = \sum \alpha_i y_i x_i$$

- Beaucoup de α sont nuls (w est une combinaison de vs).

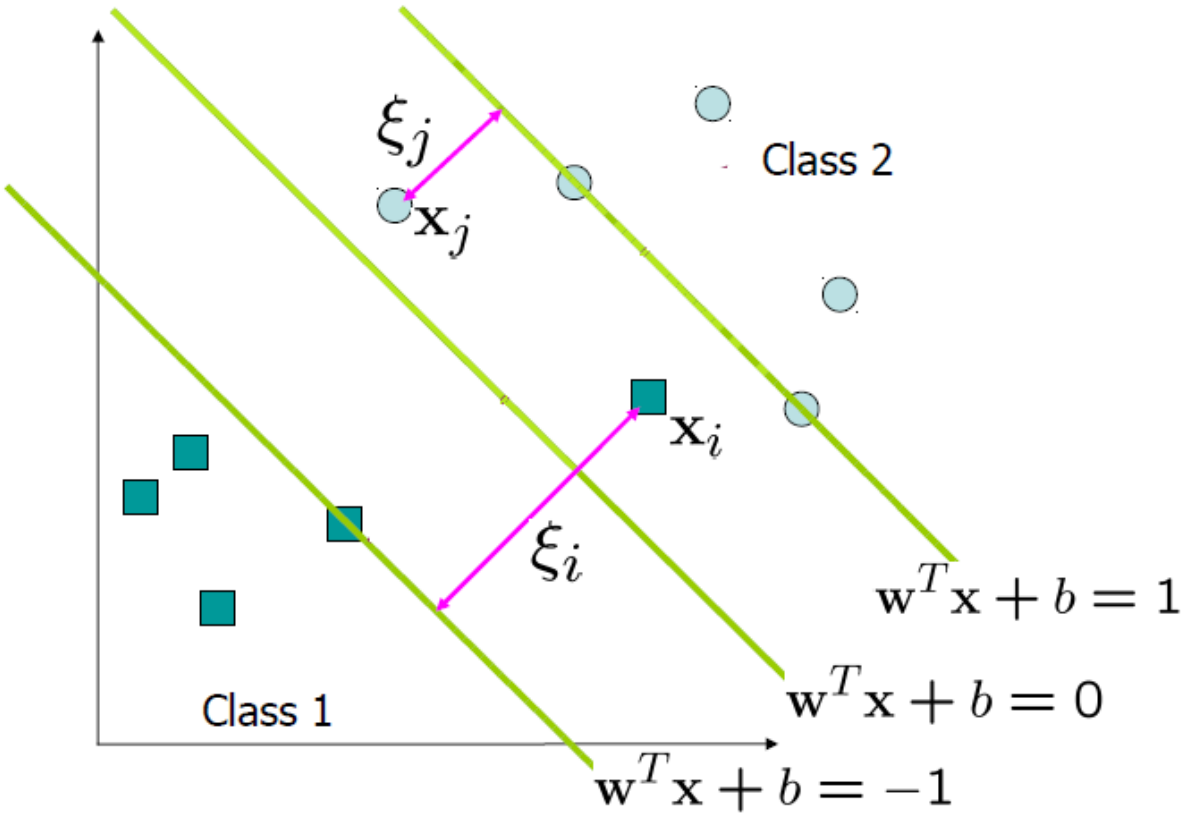
fonction de décision f

$$\hat{f}(x) = \sum \alpha_i y_i < x_i, x > + \hat{b}$$



SVM (Formalisme) IX

Marges poreuses (version relaxée)



SVM (Formalisme) X

Marges poreuses (version relaxée)

Forme primale (version relaxée)

$$\begin{cases} \text{Minimiser} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{sous les contraintes} & y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \ldots, n \end{cases}$$

1. Choisir de façon adéquate la constante C
2. w reste le même dans ce situation (rappel : w est une combinaison de vecteurs de support)

fonction de décision f

$$\hat{f}(x) = \sum \alpha_i y_i < x_i, x > + \hat{b}$$

Astuce de Fonctions Noyau I

Version non linéaire (cas plus général)

Astuce 1 : Au lieu de chercher un hyperplan dans $\mathcal{X}$, on passe d'abord dans $\mathcal{F}$ (feature space).

- Fonction non linéaire : $\phi : \mathcal{X} \mapsto \mathcal{F}$
- On suppose : $f(x) = < w, \phi(x) > + b$

fonction de décision f

$$\hat{f}(x) = \sum \alpha_i y_i < \phi(x_i), \phi(x) > + \hat{b}$$

Trouver $\phi()$ loin d'être évident !

Astuce 2 : Plutôt que de choisir $\phi : \mathcal{X} \mapsto \mathcal{F}$ on choisi une fonction $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ appelée fonction noyau

Astuce de Fonctions Noyau II

La fonction noyau représente un produit scalaire dans $\mathcal{F}$

$$k(x,z) =< \phi(x), \phi(z) >$$

Cette fonction utilise $\phi()$ indirectement, sans avoir le connaître

Exemple : k peut être calculer sans passer par ϕ
$k(x,y) = (1 + x_1y_1 + x_2y_2)^2$
$\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$
$\phi(y) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1y_2)$

- $k(x,y)$ représente intuitivement la similarité entre les x et y, obtenue de nos connaissances a priori
- $k(x,y)$ doit satisfaire certaines conditions (conditions de Mercer) pour que le $\phi()$ correspondant existe

Astuce de Fonctions Noyau III

fonction de décision f (formule generale)

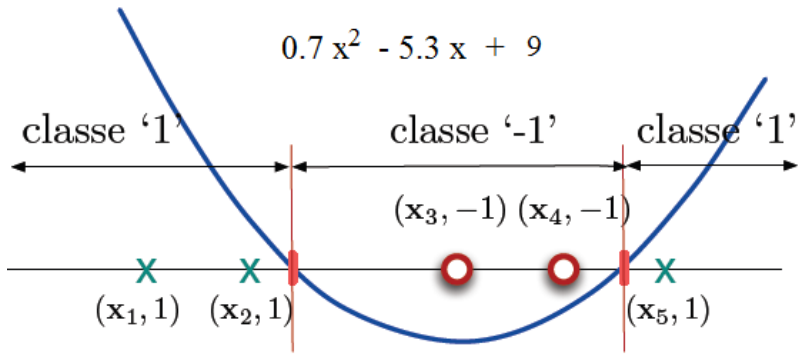
$$\hat{f}(x) = \sum \alpha_i y_i k(x_i, x) + \hat{b}$$

Si les données sont sous forme vectorielle on peut utiliser par ex. :

- Noyau linéaire : $k(x, y) =< x, y >$
- Noyau polynomial de degré d : $k(x, y) = (< x, y > +L)^d$
- Noyau gaussien : $k(x, y) = \exp(-\|x - y\|^2/2\sigma^2)$

Exemple

$S_n = (1, 1), (2, 1), (4, -1), (5, -1), (6, 1)$
Noyau : $k(x, y) = (1 + xy)^2$
Constante reg. : $C = 100$
Coef : $\alpha_1 = 0, \alpha_2 = 2.5, \alpha_3 = 0, \alpha_4 = 7.3, \alpha_5 = 4.8$



Astuce de Fonctions Noyau IV

La recherche d'autres fonctions noyau pour diverses applications est très active !

- Noyau sac de mots :

1. Représenter chaque document x sous la forme d'un vecteur

$$\phi(x) = (tf(m_1, x), tf(m_2, x), \ldots, tf(m_p, x))^T \in \mathbb{R}^p$$

où $tf(m_i, x)$ est la fréquence d'occurrence du mot m_i dans x.

2. Déterminer le produit scalaire k entre deux documents x_i et x_j

$$k(x_i, x_j) = < \phi(x_i), \phi(x_j) > = \sum_{k=1}^{p} tf(m_k, x_i) tf(m_k, x_j)$$

Astuce de Fonctions Noyau V

- Noyau Sémantiques (adaptation du noyau sac de mots) :
  1. Choisir une matrice de sémantique S de dim $p \times p$
  2. Déterminer

$$\tilde{\phi}(x) = S^T \phi(x) \qquad \tilde{k}(x_i, x_j) = \phi(x_i)^T S S^T \phi(x_j)$$

Exemple de S :

- S matrice Diagonale (**TF-IFD**) de terme

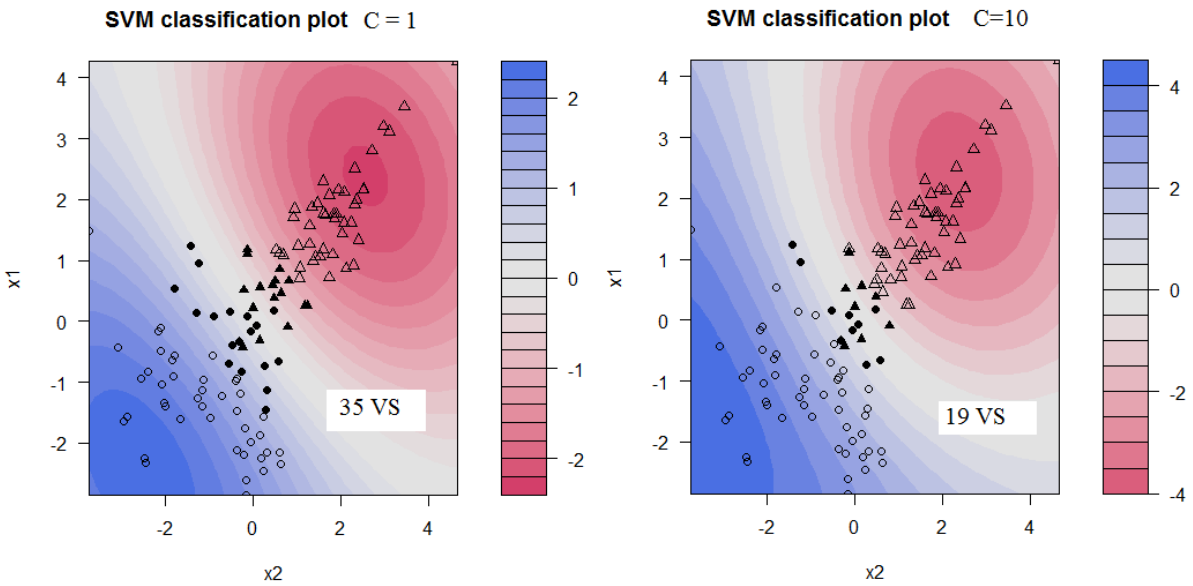
$$d_k = d(m_k) = tf(m_k, t) \ln(\frac{n}{n_k})$$

où f_k est la fréquence d'occurrence du mot m_k dans le texte t
n_k nombre de textes dans $\mathcal{C}$ contenant le mot m_k
n est le nombre de textes dans $\mathcal{C}$.

Aplications I

Exemple généré

```
x1 = rmvnorm(60, mean = c(1.5, 1.5),sigma = matrix(c(1, 0.8, 0.8, 1), 2))
x2 = rmvnorm(60, mean = c(-1 ,  -1),sigma = matrix(c(1, -0.3, -0.3,1),2))
X  = rbind(x1, x2)
ex1 = data.frame(x1 = X[,1], x2 = X[,2],class = factor(rep(1:2, c(60, 60))))
ksvm(class ~ x1 + x2, data = ex1, kernel = "rbfdot", C=1)
ksvm(class ~ x1 + x2, data = ex1, kernel = "rbfdot", C=10)
```



Aplications II

Données Spam

```
library(kernlab); data(spam)
tindex = sample(1:nrow(spam), 10) #10 points de test
f = ksvm(type ~ ., data = spam[-tindex, ],
kernel = "rbfdot", kpar = "automatic", C = 60, cross = 5)

Hyperparameter : sigma =  0.0294746438181063
Number of Support Vectors : 1133
Training error : 0.011327
Cross validation error : 0.076455

table(predict(f, spam[tindex, ]), spam[tindex, 58])
          nonspam spam
  nonspam       5    0
  spam          0    5
```

Bibliographie

T. Hastie, R. Tibshirani et J. Friedman (2009)
The Elements of Statistical Learning
Springer Series in Statistics.

C. Bishop (2009).
Pattern recognition and machine learning.
Springer.

A. Smola et al (New version 2013).
kernlab : A Kernel Methods Package

Schölkopf, A. Smola (2002)
Learning with kernels.

A. Conuéljos [AgroParisTech]
Notes de cours méthodes à Noyau
Apprentissage artificiel : Concepts et algorithmes (2ème éd.) (Chapitre 14)

S. Aseervatham, E. Viennet
Méthodes à noyaux appliquées aux textes structurés
Université de Paris-Nord

G. Dreyfus, J.-M. Martinez et al.
Apprentissage Statistique