Support Vector Machines

Séparateurs à vaste marge

Arnaud Revel
revel.arnaud@gmail.com

Plan

Plan

Qu'est-ce que l'apprentissage ?

En psychologie

Toute acquisition d'un nouveau comportement à la suite d'un entraînement : habituation, conditionnement...

En neurobiologie

Modifications synaptiques dans des circuits neuronaux : règle de Hebb, règle de Rescorla et Wagner...

Apprentissage **automatique**

- construire un **modèle général**
- à partir de données **particulières**

But

prédire un comportement face à une nouvelle donnée
approximer une fonction ou une densité de probabilité

Formalisation

- Soit un ensemble d'apprentissage $S = \{(\mathbf{x_i}, y_i)\}_{1..n}$ dont les éléments obéissent à la loi jointe $P(x, y) = P(x)P(y|x)$
- On cherche à approcher une loi sous-jacente $f(x)$ telle que $y_i = f(\mathbf{x_i})$ par une hypothèse $h_\alpha(x)$ *aussi proche que possible*
- Les α sont les paramètres du système d'apprentissage.

- Si $f(.)$ est discrète on parle de **classification**
- Si $f(.)$ est une fonction continue on parle alors de **régression**

Mais que veut-on dire par

"aussi proche que possible" ?

Calcul du risque

Pour mesurer la qualité d'une hypothèse h_α on va considérer une fonction de coût $Q(z = (x, y), \alpha) \in [a, b]$ que l'on cherche à minimiser

Exemple de fonction de coût

Coût 0/1 : vaut 0 lorsque les étiquettes prévues et observées coïncident, 1 sinon : utilisé en classification

Erreur quadratique : $(f(x) - y)^2$: utilisé en régression

On cherche à minimiser : $R(\alpha) = \int Q(z, \alpha) dP(z)$

Comme on ne peut accéder directement à cette valeur, on construit donc le risque empirique qui mesure les erreurs réalisées par le modèle : $R_{emp}(\alpha) = \frac{1}{m} \sum_{i=1}^{n} Q(z_i, \alpha)$

Mais quel est le lien entre $R_{emp}(\alpha)$ et $R(\alpha)$?

Théorie de l'apprentissage de Vapnik (1995)

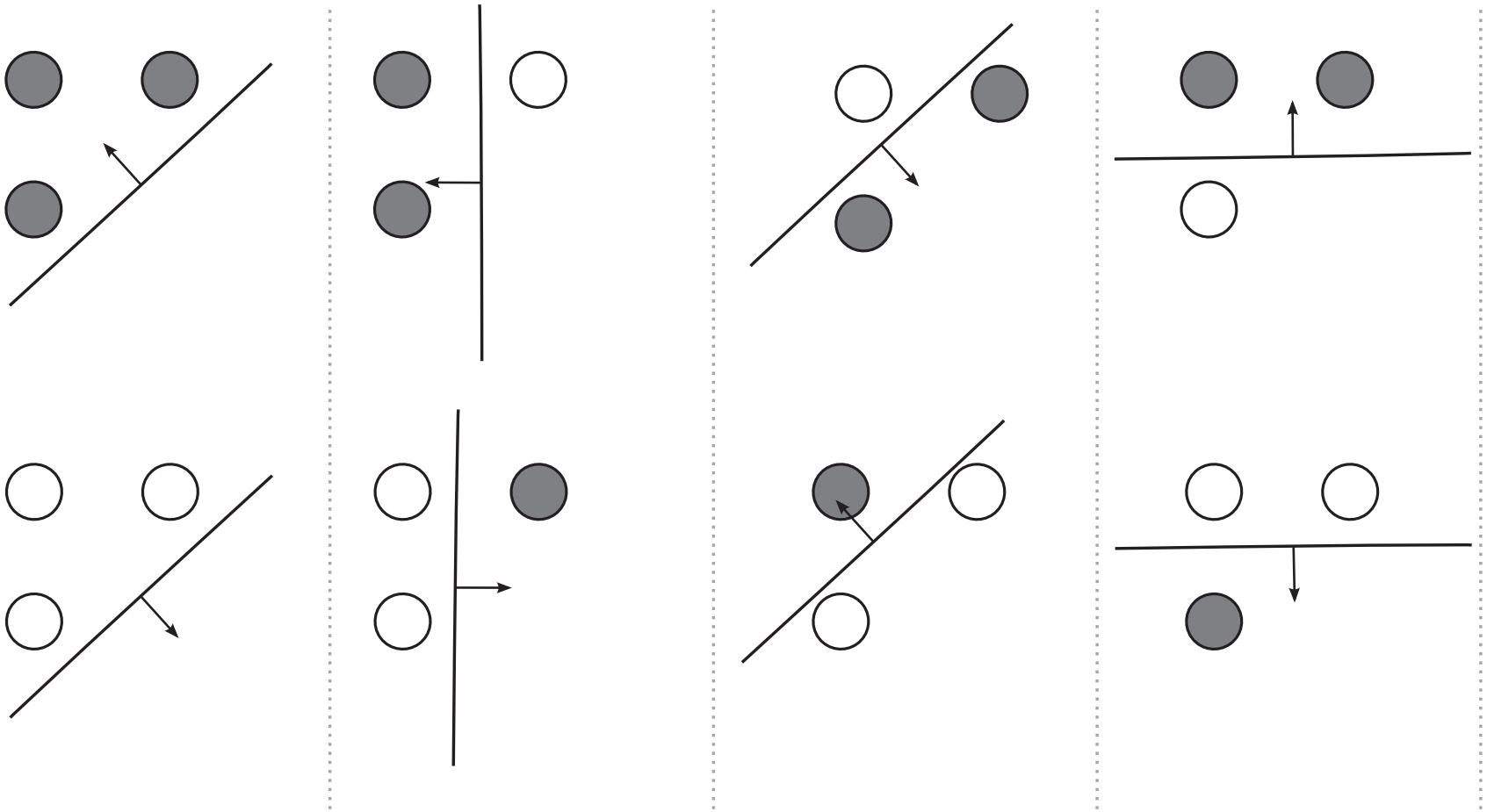
Vapnik a pu montrer l'expression suivante $\forall m$ avec une probabilité au moins égale à $1 - \eta$:

$$R(\alpha_m) \leq R_{emp}(\alpha_m) + (b - a)\sqrt{\frac{d_{VC}(\ln(2m/d_{VC}) + 1) - ln(\eta/4)}{m}} \quad (1)$$

La minimisation du risque dépend

- du **risque empirique**
- un **risque structurel** lié au terme d_{VC} qui dépend de la complexité du modèle h choisi (VC-dimension [a])

a. Dimension de Vapnik et Chervonenkis

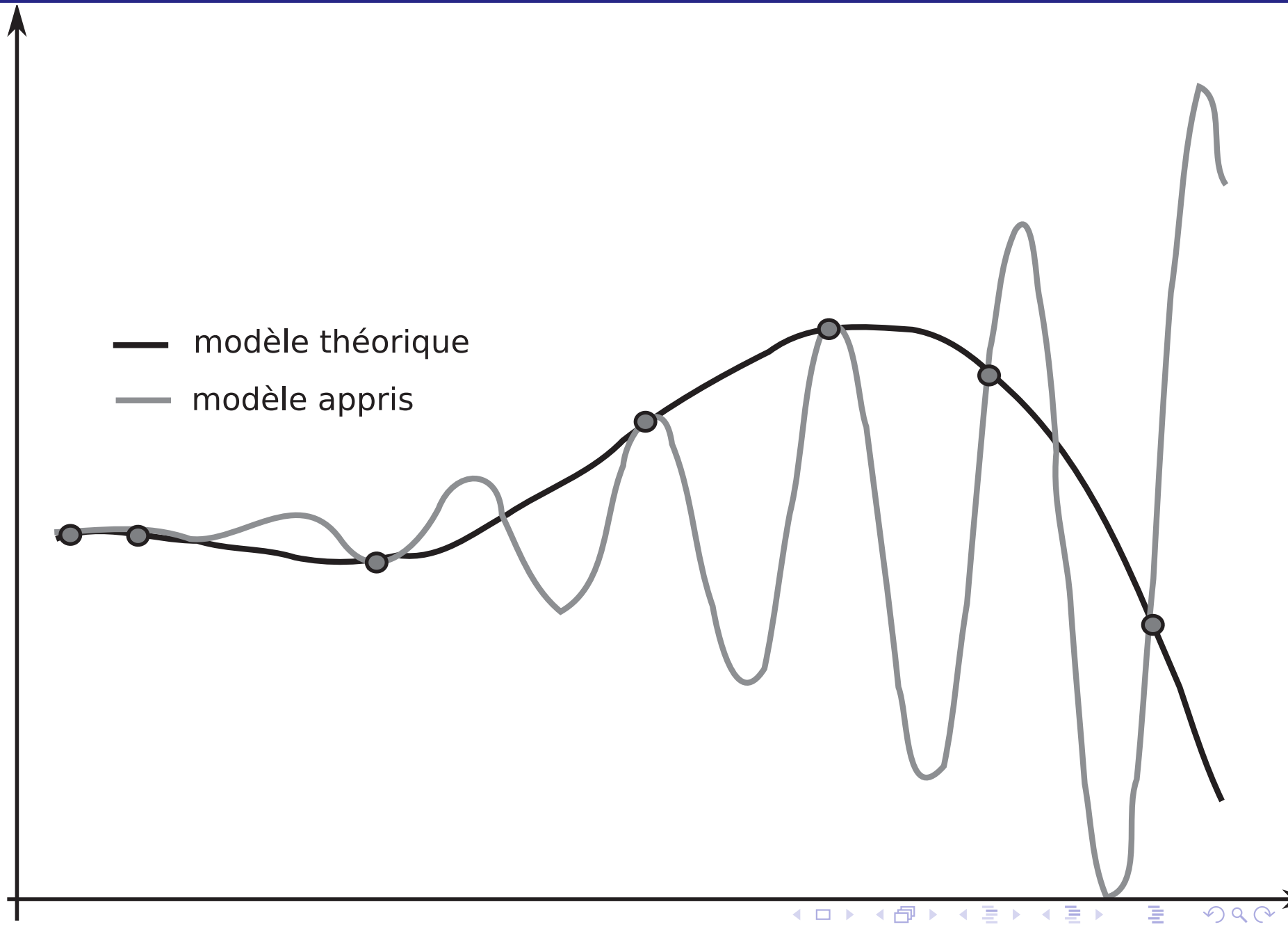


Ainsi, si pour construire un bon modèle d'apprentissage, il est nécessaire de :

minimiser les erreurs sur la base d'apprentissage

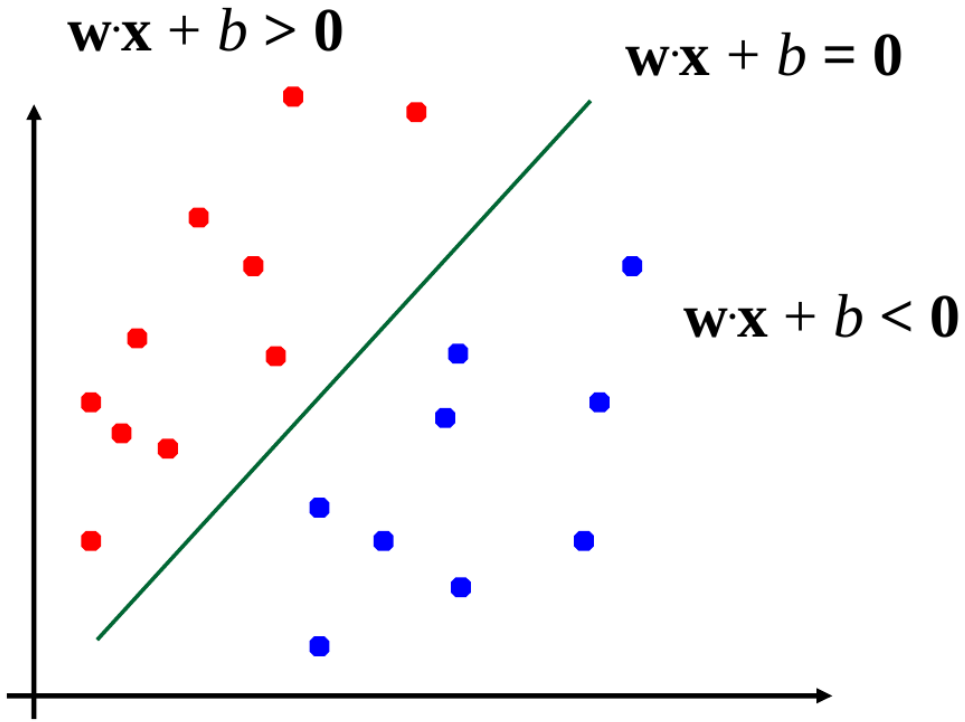
c'est le principe d'induction naïf utilisé dans les réseaux de neurones

de construire un système capable de **généraliser** correctement


modèle théorique
modèle appris

Revisitons le problème du perceptron :

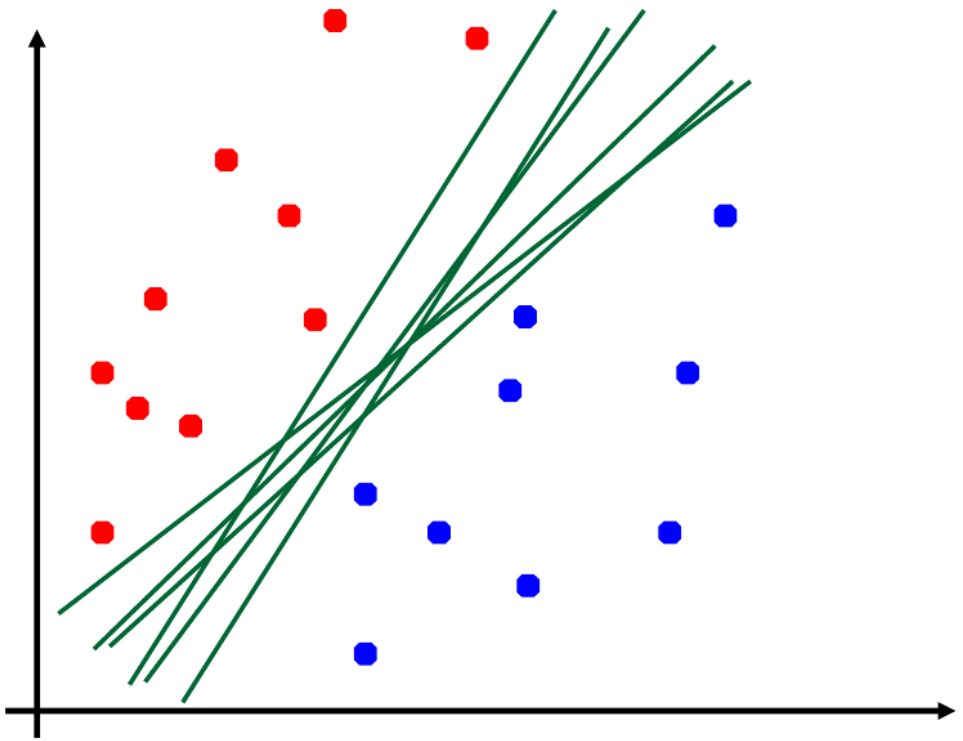
Classifieur linéaire : $y(x) = signe(\mathbf{w} \cdot \mathbf{x} + b)$



Quel est le meilleur classifieur ? :

Il existe de nombreux choix possibles pour **w** et b :

$$y(x) = signe(\mathbf{w} \cdot \mathbf{x} + b) = signe(k\mathbf{w} \cdot \mathbf{x} + k \cdot b) \qquad (2)$$



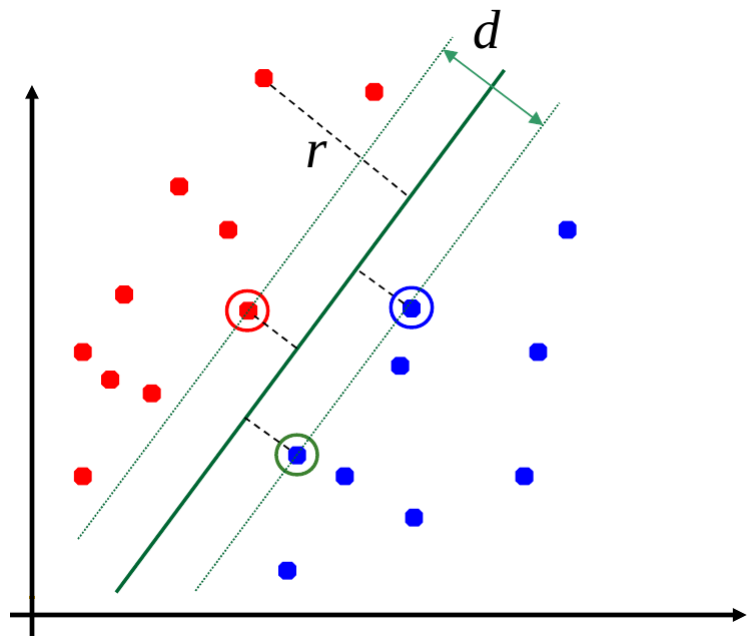
Plan

Notion de marge :

Dans le cas séparable, on va considérer les points les plus près de l'hyperplan séparateur : **vecteurs supports** (`support vectors`).

Pour tout point de l'espace des exemples, la distance à l'hyperplan séparateur est donnée par :

$$r = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{||\mathbf{w}||} \qquad (3)$$



On appelle marge d la distance entre les 2 classes

C'est cette distance **d** qu'on souhaiterait maximiser

Quantification de la marge :

Pour limiter l'espace des possibles on considère que les points les plus proches sont situés sur les hyperplans canoniques donnés par :

$$\mathbf{w} \cdot \mathbf{x} + b = \pm 1 \tag{4}$$

Dans ce cas, la marge est : $d = \frac{2}{||\mathbf{w}||}$

Les conditions d'une bonne classification sont :

$$\begin{cases} \mathbf{w} \cdot \mathbf{x} + b \geq 1, & \text{si } y_i = 1 \\ \mathbf{w} \cdot \mathbf{x} + b < 1, & \text{si } y_i = -1 \end{cases} \tag{5}$$

Maximisation de la marge :

Le problème revient alors à trouver $\mathbf{w}$ et b tels que $d = \frac{2}{||\mathbf{w}||}$ est maximale $\forall(x_i, y_i)$

Sous les contraintes :

$$\begin{cases} \mathbf{w} \cdot \mathbf{x} + b \geq 1, & \text{si } y_i = 1 \\ \mathbf{w} \cdot \mathbf{x} + b < 1, & \text{si } y_i = -1 \end{cases} \tag{6}$$

De manière équivalent, le problème peut s'écrire plus simplement comme la minimisation de :

$$\frac{1}{2}||\mathbf{w}||^2 \tag{7}$$

Sous les contraintes : $y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \forall i \in [1, N]$

Maximisation de la marge :

Cette minimisation est possible sous les conditions dites de "Karush-Kuhn-Tucker (KKT)"

Soit le Lagrangien $\mathcal{L}$:

$$\mathcal{L}(\mathbf{w}, b, \lambda) = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \lambda_i [y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1]$$

Les conditions de KKT sont alors :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0,\ \frac{\partial \mathcal{L}}{\partial b} = 0,\ \frac{\partial \mathcal{L}}{\partial \lambda_i} \geq 0,\ \lambda_j \geq 0$$
$$\lambda_i [y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1] = 0$$

Par ailleurs la dernière condition implique que pour tout point ne vérifiant pas $y_i(\mathbf{w} \cdot \mathbf{x_i} + b) = 1$ le λ_i est nul.

Les points qui vérifient $y_i(\mathbf{w} \cdot \mathbf{x_i} + b) = 1$, sont appelés "vecteurs supports". Ce sont les points les plus près de la marge. Ils sont sensés être peu nombreux par rapport à l'ensemble des exemples.

Le problème dual :

Le problème s'exprime sous forme duale comme la minimisation de :

$$W(\lambda) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) \qquad (8)$$

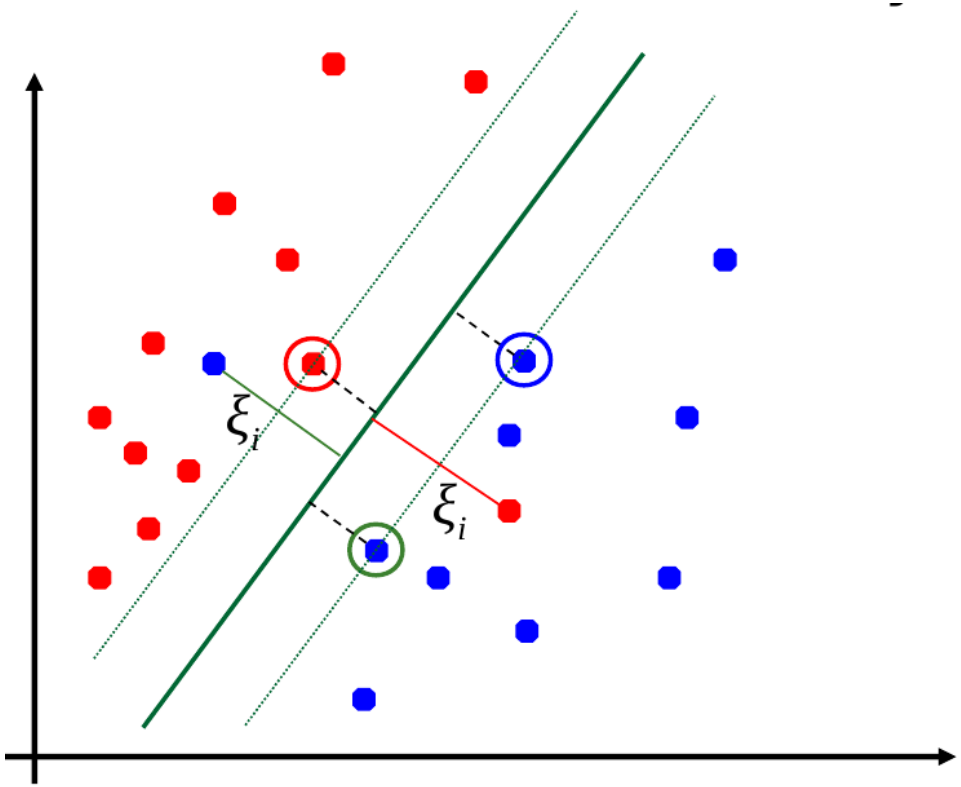
Fait partie des problèmes d'optimisation quadratique pour lesquels il existe de nombreux algorithmes de résolution

SMO	résolution analytique (par 2 points), gestion efficace de la mémoire, mais converge en un nombre d'étapes indéterminé
SimpleSVM	facilite de la reprise a chaud, converge en moins d'étapes mais limitation mémoire
LASVM	utilisation en ligne, résolution analytique mais solution sous optimale, plusieurs passes nécessaires pour les petites bases de données

Classification à marge souple :

Et si les données ne sont pas linéairement séparables ?

L'idée est d'ajouter des variables d'ajustement ξ_i dans la formulation pour prendre en compte les erreurs de classification ou le bruit



Classification à marge souple : formulation

Problème original

Minimiser $\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i$

Étant donné : $y_i(\mathbf{w}^* \cdot \mathbf{x_i} + b) \geq 1 - \xi_i$

pour $i = 1, ..., N$ et $\xi_i \geq 0$

C est une constante permettant de contrôler le compromis entre nombre d'erreurs de classement, et la largeur de la marge

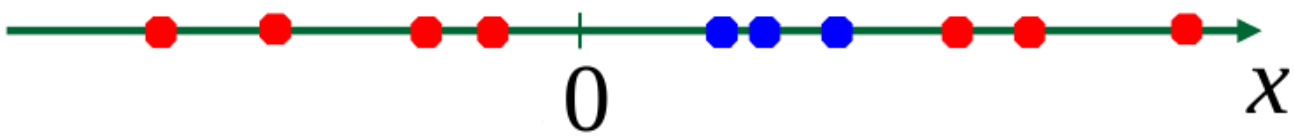
Problème dual

Minimiser $\mathcal{L}(\lambda) = \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}$

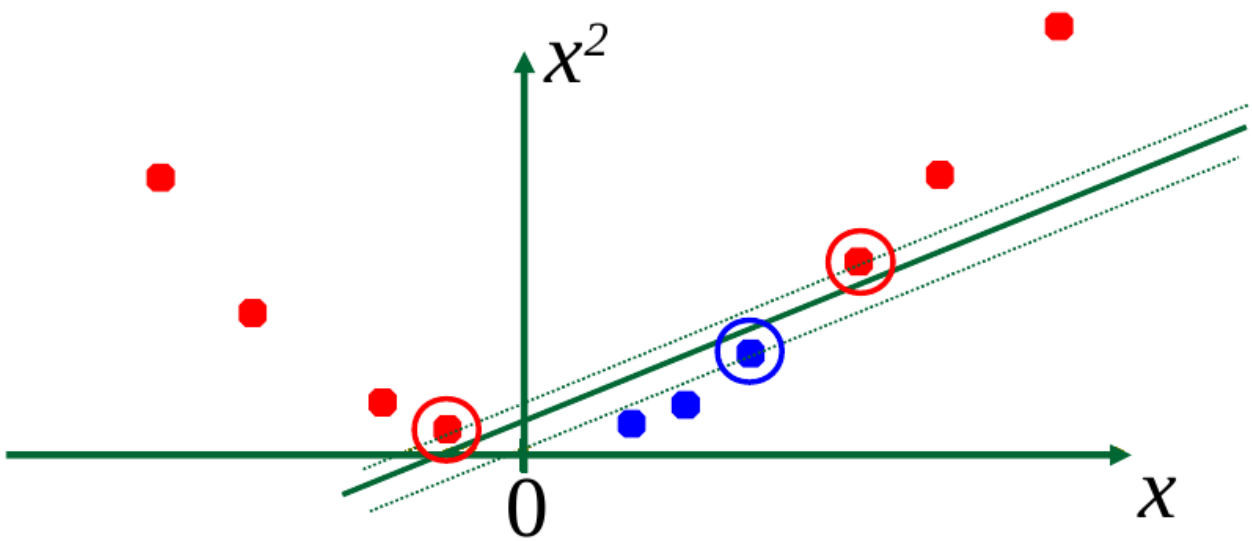
Avec les contraintes $0 \leq \lambda_i \leq C$

Au delà du séparateur linéaire

Que se passe-t-il si l'ensemble d'apprentissage est intrinsèquement non séparable ?



Pourquoi ne pas plonger le problème dans un espace de plus grande dimensionnalité ?



Plan

SVM non-linéaires : espace de caractéristiques

Idée générale

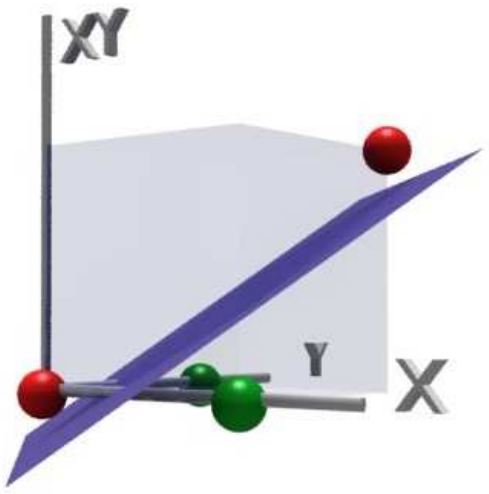
L'espace des données peut toujours être plongé dans un espace de plus grande dimension dans lequel les données peuvent être séparées linéairement

Exemple : XOR

On effectue la transformation :

$$(x, y) \rightarrow (x, y, x \cdot y) \qquad (9)$$

Dans ce cas le problème peut être séparé linéairement



Le "kernel trick"

La résolution des SVM ne s'appuient que sur le produit scalaire $< \mathbf{x_i}, \mathbf{x_j} >$ entre les vecteurs d'entrée

Si les données d'apprentissage sont plongées dans un espace de plus grande dimension via la transformation $\Phi : \mathbf{x} \rightarrow \phi(\mathbf{x})$, le produit scalaire devient :

$$K(\mathbf{x_i}, \mathbf{x_j}) =< \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) > \tag{10}$$

($K(\mathbf{x_i}, \mathbf{x_j})$ est appelée *fonction noyau*)

Pour faire apprendre le SVM seul le noyau est important, sans qu'il ne soit nécessaire d'effectuer la transformée $\phi(\mathbf{x})$

SVM non-linéaire : formulation

Problème original

- Minimiser $\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i$
- Étant donné : $y_i(\mathbf{w}^* \cdot \phi(\mathbf{x_i}) + b) \geq 1 - \xi_i$, pour $i = 1, ..., N$ et $\xi_i \geq 0$

Problème dual

- Minimiser $\mathcal{L}(\lambda) = \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_\mathbf{i}\lambda_\mathbf{j}\mathbf{y_i}\mathbf{y_j}\mathbf{K}(\mathbf{x_i}, \mathbf{x_j})$
- Sous les contraintes $0 \leq \lambda_\mathbf{i} \leq C$
- Les techniques d'optimisation restent les mêmes
- La solution est de la forme :

$$\mathbf{w}^* = \sum_{i \in SV}\lambda_i y_i \phi(\mathbf{x_i})$$
$$f(x) = \mathbf{w}^* \cdot \phi(\mathbf{x}) + b^* = \sum_{i \in SV}\lambda_i y_i K(\mathbf{x_i}, \mathbf{x}) + b^*$$

Exemples de noyaux

Noyau polynôme de degré 2 à 2 variables

Transformée non-linéaire :

$$\mathbf{x} = (x_1, x_2)$$
$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

Le noyau est alors :

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$
$$\phi(\mathbf{y}) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1y_2)$$
$$K(x, y) = \phi(x) \cdot \phi(y) = (1 + x \cdot y)^2$$

Comment savoir si K est un noyau ?

Noyau de Mercer

- On appel noyau de Mercer une fonction continue, symétrique, semi-définie positive $K(x, y)$
- $\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j), \forall \alpha \in \mathbb{R}$

Matrice de Gram

Matrice des termes $< x_i, x_j >$. Elle est symétrique et semi-définie positive pour une noyau de Mercer

Théorème de Moore-Aronszajn (1950)

- Toute fonction semi-définie positive $k(x, y)$ est un noyau, et réciproquement. Elle peut s'exprimer comme un produit scalaire dans un espace de grande dimension.
- $k(x_i, x_j) =< \phi(x_i), \phi(x_j) >$

Quelques noyaux utilisables dans Weka

Noyau linéaire $K(x_i, x_j) = x_i \cdot x_j$

Noyau polynomial de degré p $K(x_i, x_j) = (1 + x_i \cdot x_j)^p$

Noyau Gaussien $K(x_i, x_j) = \exp \frac{-||x_i - x_j||^2}{2\sigma^2}$

- Cette formulation est équivalente aux réseaux de neurones à bases radiales avec l'avantage supplémentaire que les centres des fonctions à base radiale (qui sont les vecteurs supports) sont optimisés

Perceptron à 2 couches $K(x_i, x_j) = \tanh(\alpha x_i \cdot x_j + \beta)$

Construire d'autres noyaux

- $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y}) + k_2(\mathbf{x}, \mathbf{y})$
- $k(\mathbf{x}, \mathbf{y}) = \alpha \cdot k_1(\mathbf{x}, \mathbf{y})$
- $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y}) \cdot k_2(\mathbf{x}, \mathbf{y})$
- $k(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) \cdot f(\mathbf{y})$ avec $f()$ une fonction de l'espace des attributs dans $\mathbb{R}$
- $k(\mathbf{x}, \mathbf{y}) = k_3(\phi(\mathbf{x}), \phi(\mathbf{y}))$
- $k(\mathbf{x}, \mathbf{y}) = \mathbf{x} B \mathbf{y}^T$ avec B une matrice $N \times N$ symétrique, semi-définie positive.

Plan

Cas multi-classes

Les Séparateurs à vaste marge ont été développés pour traiter des problèmes binaires mais ils peuvent être adaptés pour traiter les problèmes multi-classes.

Stratégie un contre tous

- L'idée consiste simplement à transformer le problème à k classes en k classifieurs binaires.
- Le classement est donné par le classifieur qui répond le mieux.

Pb : beaucoup d'exemples négatifs !

Cas multi-classes

Stratégie un contre un

- Cette fois le problème est transformé en $\frac{k \cdot (k-1)}{2}$ classifieurs binaires : chaque classe i étant en effet comparée à chaque classe j.
- Le classement est donné par le vote majoritaire ou un graphe acyclique de décision.

Plan

Applications des SVM

- Les avantages théoriques (minimisation de l'erreur empirique et structurelle) et pratiques (algorithmes optimisés) des SVM en ont fait un outil très prisé dans de nombreux problèmes pratiques de classification.
- Dans bien des cas, il s'agit de construire un noyau (donc une mesure de similarité) adapté aux données à traiter.

Exemple d'applications

- Classification de données biologiques/physiques
- Classification de documents numériques
- Classification d'expressions faciales
- Classification de textures
- E-learning
- Détection d'intrusion
- Reconnaissance de la parole
- CBIR : Content Based Image Retrieval

Plan

Bibliographie

CANU, S. (2007).
Machines à noyaux pour l'apprentissage statistique.
Techniques de l'ingénieur - Dossier : TE5255.

Cortes, C. and Vapnik, V. (1995).
Support-vector networks.
Machine Learning, 20(3) :273–297.

Guermeur, Y. and Paugam-Moisy, H. (1999).
Apprentissage Automatique, chapter Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines, pages 109–138.
Hermés.
http ://www.loria.fr/ guermeur/SVM-final.ps.